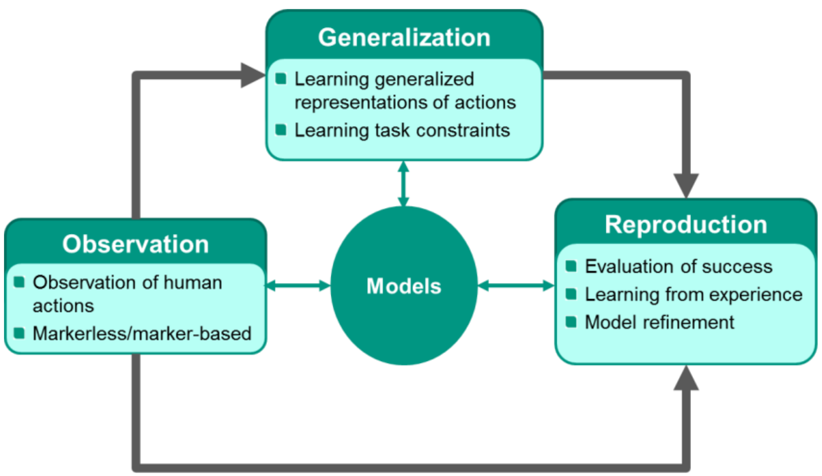
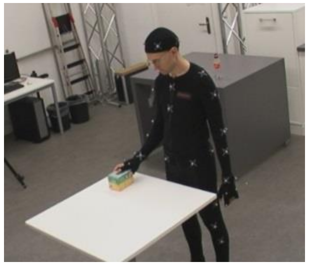
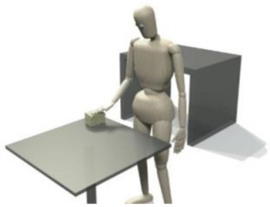
Imitation Learning

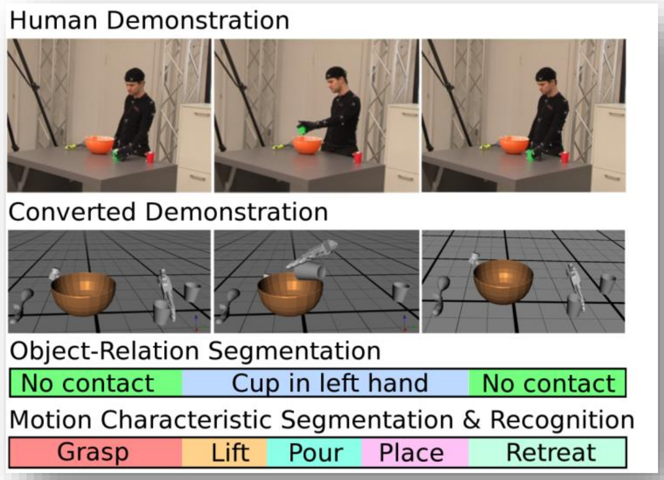
Tamim Asfour

KIT-Department of Informatics, Institute for Anthropomatics and Robotics, High Performance Humanoid Technologies (H^2T)









What is imitation?

- Imitation (from Latin imitatio, "a copying, imitation") is an advanced behavior whereby an individual observes and replicates another's behavior. Imitation is also a form of social learning that leads to the "development of traditions, and ultimately our culture. It allows for the transfer of information (behaviours, customs, etc.) between individuals and down generations without the need for genetic inheritance.“

- The word imitation can be applied in many contexts, ranging from animal training to politics

Source: Wikipedia

Imitation Learning

- Imitation learning from the viewpoint of
  - Computer science
  - Cognitive science
  - Neuroscience

Computer science view

- Three Phases Model
  - Decomposition into subproblems
  - Solution to different problems

Cognitive science view (1)

- Focus on „Software“ of brains
- Until 1970, movement imitation did not receive widespread attention anymore, partially due to the prejudice that “imitating” or “mimicking” is not an expression of higher intelligence
- True imitation, if
  - the imitated behavior is absolutely new for the imitator
  - the same strategy is employed as that of the demonstrator
  - The same task goal is achieved

Stefan Schaal (1999). Is Imitation Learning the Route to Humanoid Robots? *Trends in Cognitive Sciences* 3:233-242

Cognitive science view (2)

- Is Imitation an innate ability?
- Newborns can imitate? → yes!
- Meltzoff and Moore (Meltzoff 1997, Meltzoff 1983)
  - They reported on the ability of 12-21 day old and, later, even less than an hour old infants to imitate both facial and manual gestures.
  - Young infants of this age had neither seen their own faces nor been exposed to viewing faces of other humans for any significant amount of time. Thus, the ability to map a perceived facial gesture to their own gestures was concluded to be innate and contradicted Piaget’s ontogenetic account of imitation (developmental theory)

Cognitive science view (3)

- In the light of this new interest in imitation learning, it was discovered that many animals are unable to learn by imitation.

- These findings contributed to today's view of imitation as an important expression of higher intelligence.

Cognitive science view (4)

- Innate Releasing Mechanism (IRM), Meltzoff and Moore:
  Model to explain the ability of newborn to imitate

  - Every newborn has a certain repertoire of innate movement patterns
  - By observing others‘ movements, these pattern will be activated and a correspondent movement is released;
  - The model is very interesting because it supports the idea of implementing a set of pre-programmed behaviors in a technical system (robot) to bootstrap learning by imitation

Cognitive science view (5)

- IRM is unlikely because:
  - On the one hand, many movements can be imitated, which means that a large number of these movements should be pre-programmed
  - On the other hand, newborns are constantly trying to improve the imitated movements, which also speaks against a firm anchorage.

J. Demiris and Hayes, G. (1996). Imitative learning mechanisms in robots and humans. In Klingspor, V., editor, *Proceedings of the 5th European Workshop on Learning Robot*.

Cognitive science view (5)

- Active Intermodal Mapping (AIM) Modell:
  - The visual perception of the teacher's movement is converted into a higher level representation that can be matched against appropriately transformed proprioceptive information about one's own movement.
  - If this matching space is given, imitation can be seen as learning to achieve a target representation, a problem that can be tackled with techniques from supervised learning

Neuroscience and Cognitive Neuroscience (1)

- Focus on „Hardware" of brains
- Imitations-region in brains (F5)
  - Active during observation **and** movement
  - Active during **whole** observation
  - An essential prerequisite for imitation is a connection between the sensory systems and the motor systems such that percepts can be mapped onto appropriate actions. This mapping is a difficult computational process as visual perception takes place in a different coordinate frame to motor control.
- Connection to some neurons in F5, the mirror neurons, which are active *both* when the monkey *observed* a specific behavior and when it *executed* it itself

G. Rizzolatti, R. Camarda, L. Fogassi, M. Gentilucci, G. Luppino, M. Matelli (1988). Fuctional organization of inferior area 6 in the macaque monkey. II. Area F5 and the control of distal movements. Exp. Brain Res., 71, 491-507

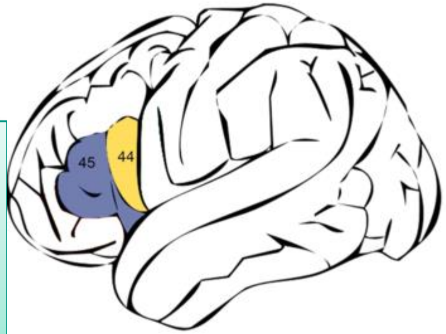
Neuroscience and Cognitive Neuroscience (2)

- Mirror neurons are active during the whole observation and action and not only during a certain subsequence

  → neurons codes complete movement and concrete types of movements, not only their subsequences.

- This result plays a very important role for finding an appropriate representation of observed movements.
- A special area in human‘s brain (Broca-Area) is found, which has almost the same function as mirror neurons.

- M. Jeannerod, M. A. Arbib, G. Rizzolatti, H. Sakata (1995) Grasping objects: the cortical mechanisms of visuomotor transformation. Trens Neurosci., 18, 314-320
- A. Murata, L. Fadiga, L. Fogassi, V. Gallese, V. Raos, G. Rizzolatti (1997). Object representation in the ventral premotor cortex (area F5) of the monkey. J. Neurophysiol., 78, 2226-30



Broca-Area

Key Issues of Imitation

- **WHAT** to imitate
  - … essential
  - … Start and end by sequencing
  - … goal
- **HOW** to imitate
  - … match the movement of the demonstrator to the own movement
- **WHOM** to imitate
- **WHEN** to imitate

Key Issues of Imitation

- How
  - After knowing what to imitate, the imitated movement must be realized → Mapping the movement of the demonstrator onto the own body.
  - Because two bodies might be different. They might have different sizes or different degrees of the freedom

    → **Correspondence-Problem**

  - On the other hand, a goal-oriented imitation must be found to reach the goal if the movement itself can not be imitated for this goal.
  - Solutions for correspondence problems
    - Exact matching (Learning from the identical body)
    - Achieving some of the objectives is sufficient for a valid solution.

Meltzoff's imitation learning model

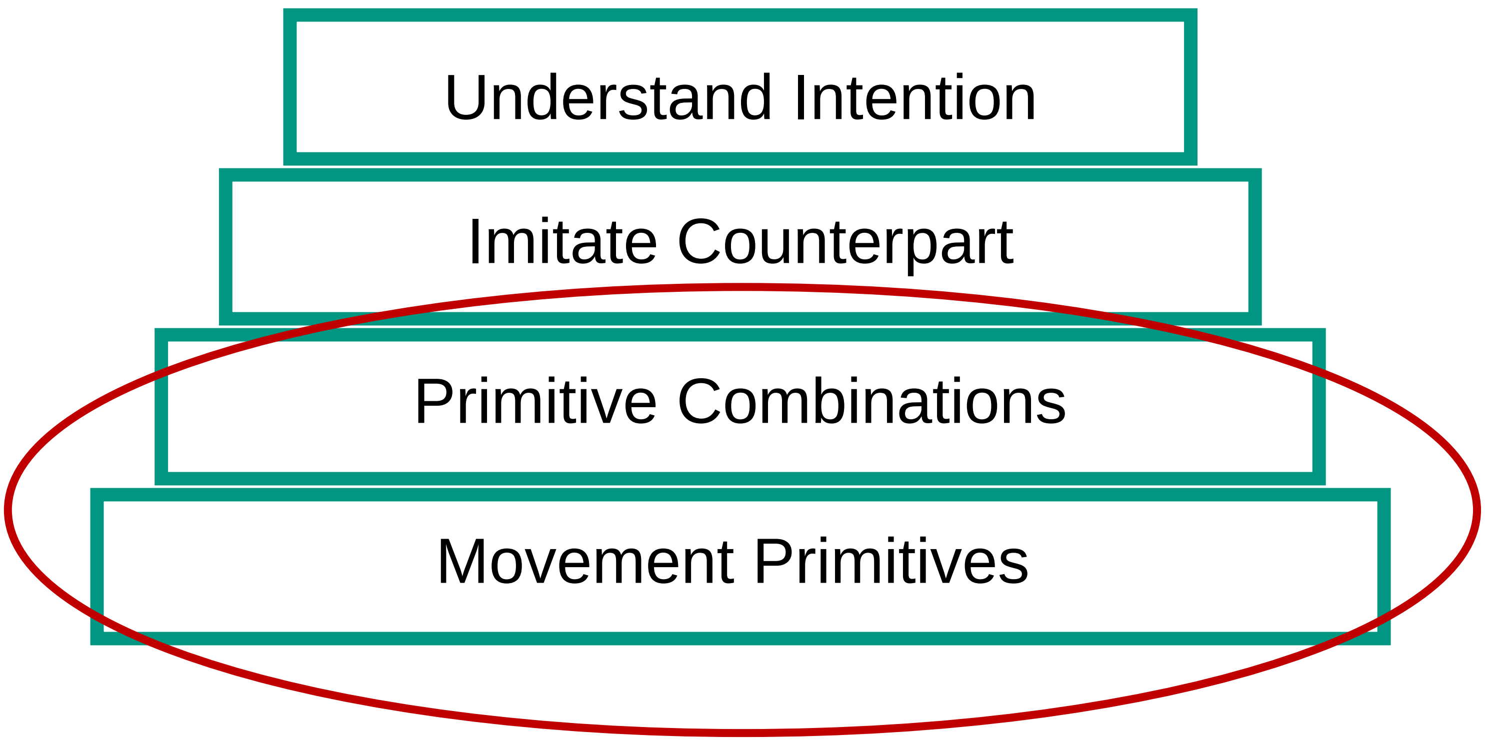
Understand Intention

Imitate Counterpart

Primitive Combinations

Movement Primitives

Meltzoff's imitation learning model

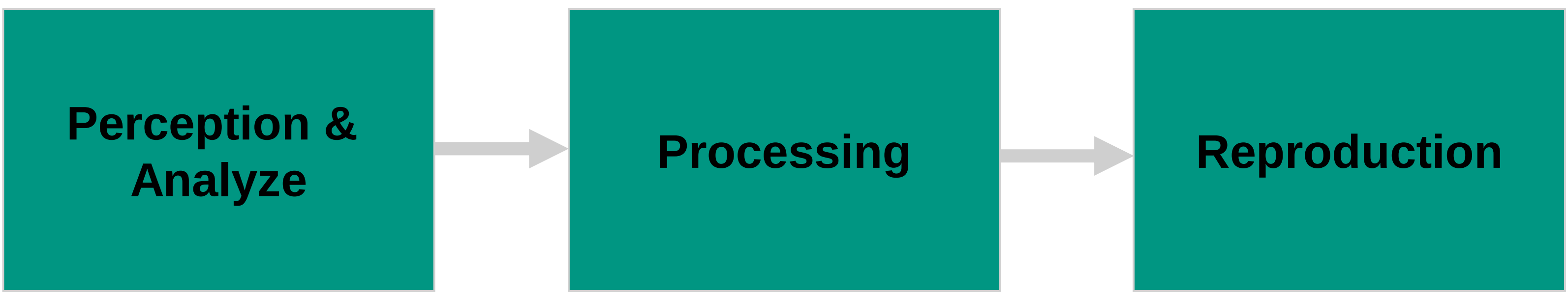


Passive und aktive Imitation

- Passive Imitation:
  The motor system of the imitator is only activated during the reproduction phase and not during the observation phase.

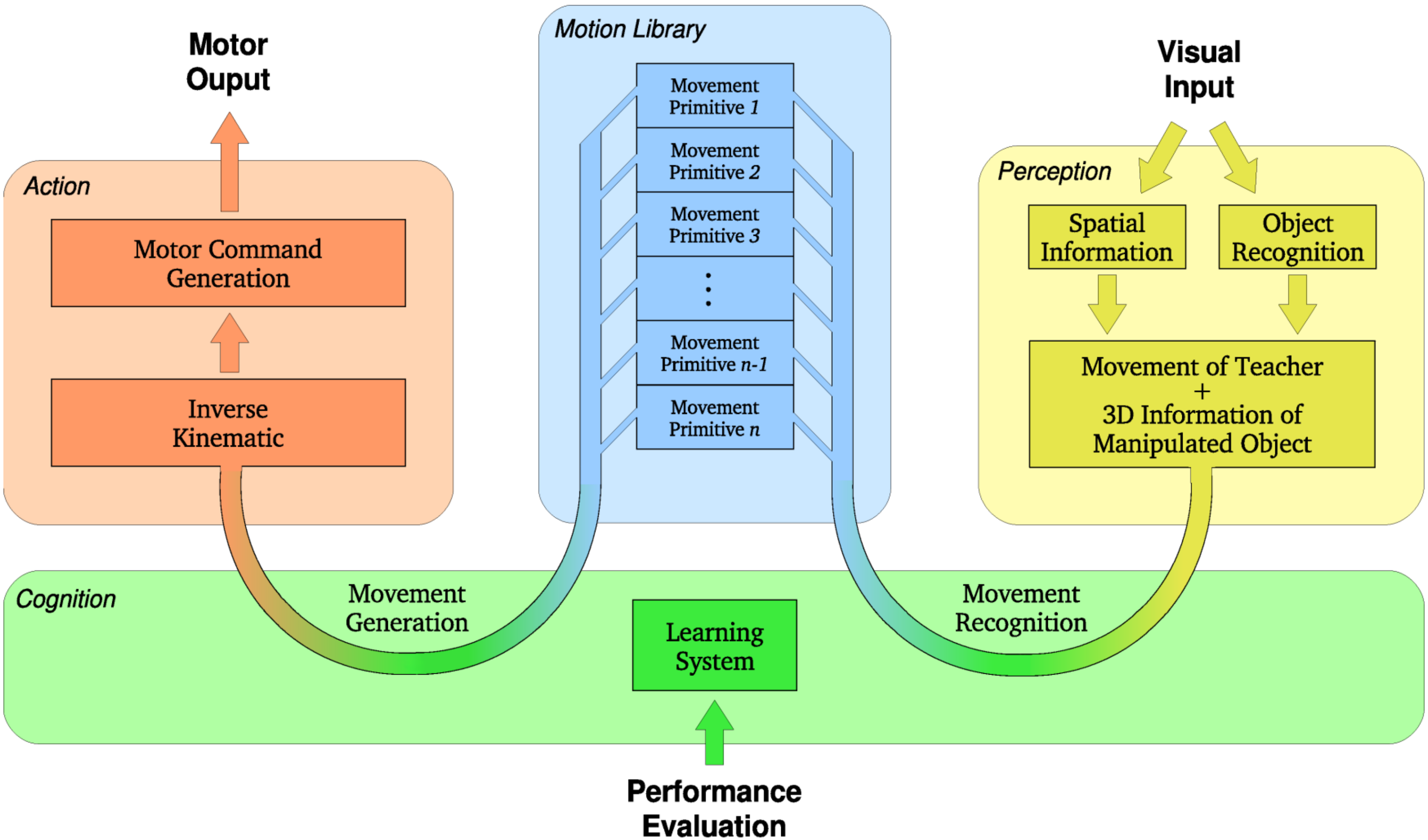
- Active Imitation:
  The motor system of the imitator is actived during both observation and reproduction phases.

Process (simplified)



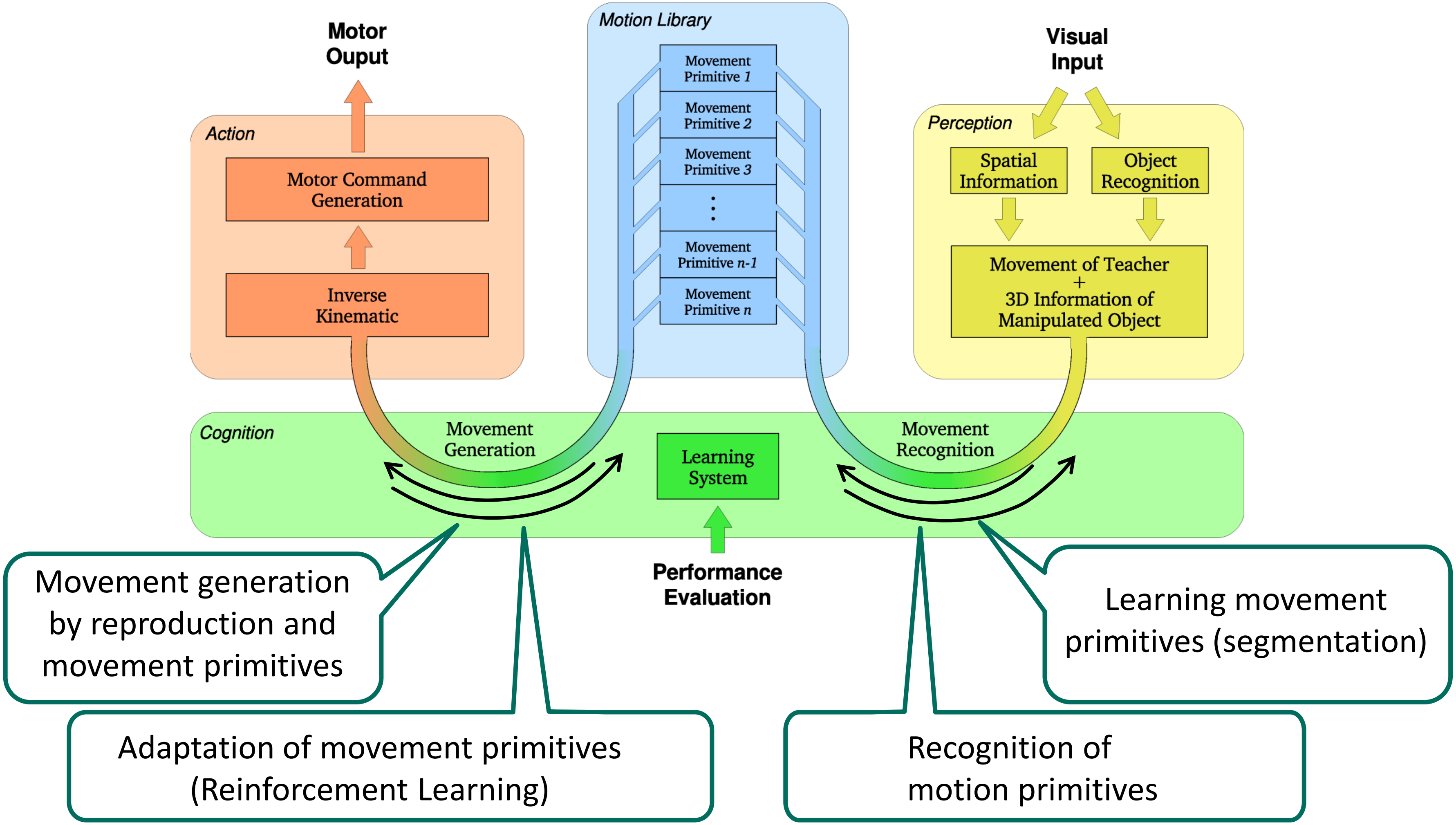
Imitation Learning

Idea: Reproduction of mirror neurons by movement primitives



Imitation Learning

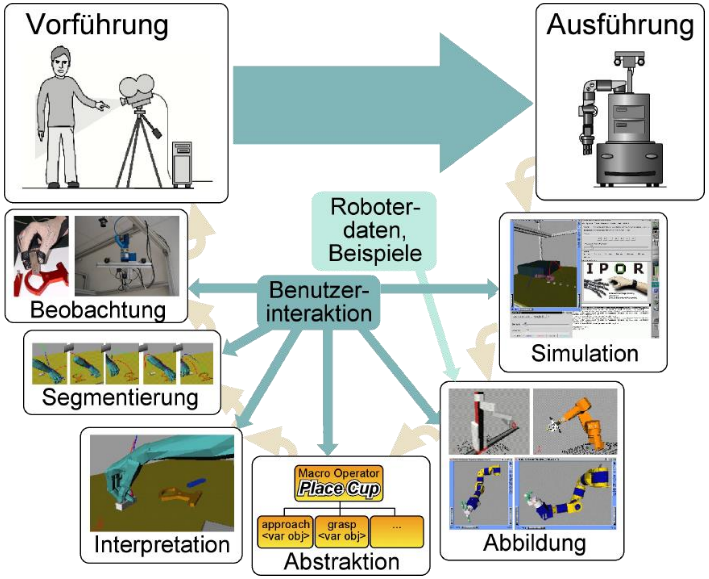
Idea: Reproduction of mirror neurons by movement primitives



Motivation: Learning from demonstration

Technical oriented

Programming by demonstration

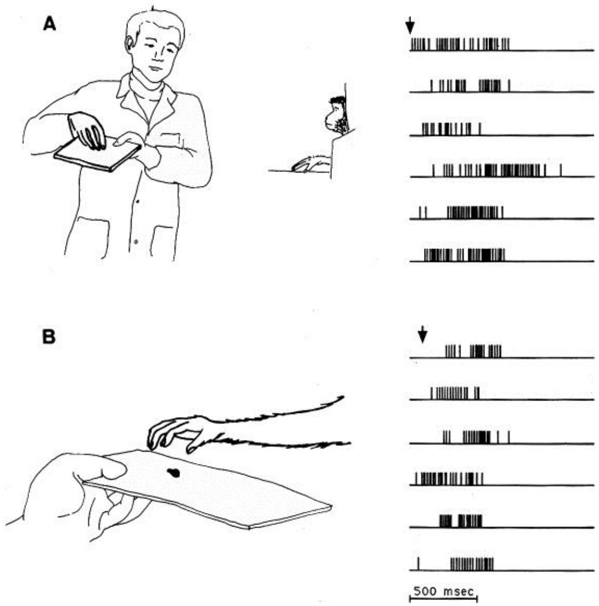


Dillmann et. al (2002)

⟷

Biologically motivated

Learning by Imitation

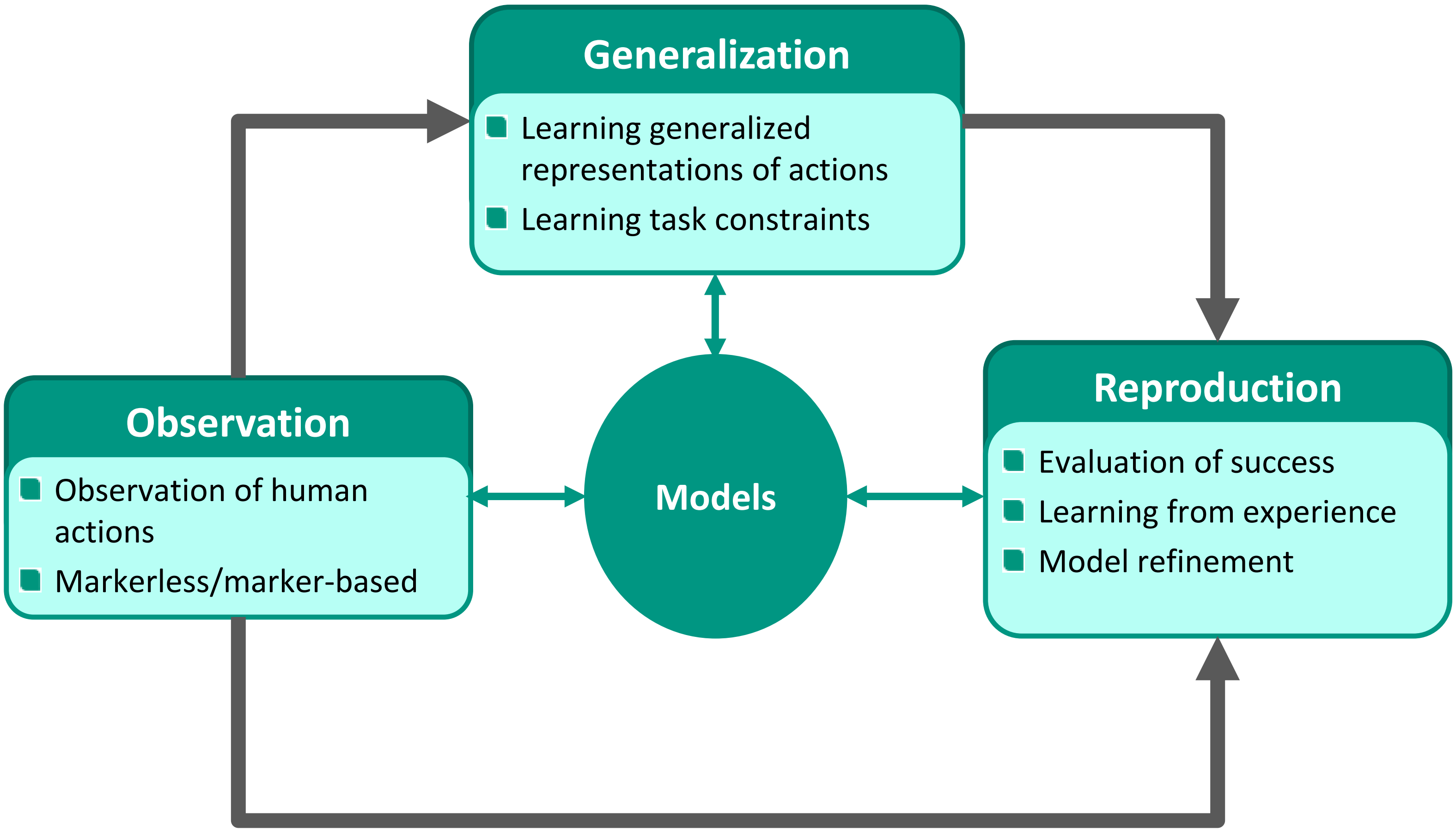


Gallese & Rizzolatti et. al (1992)

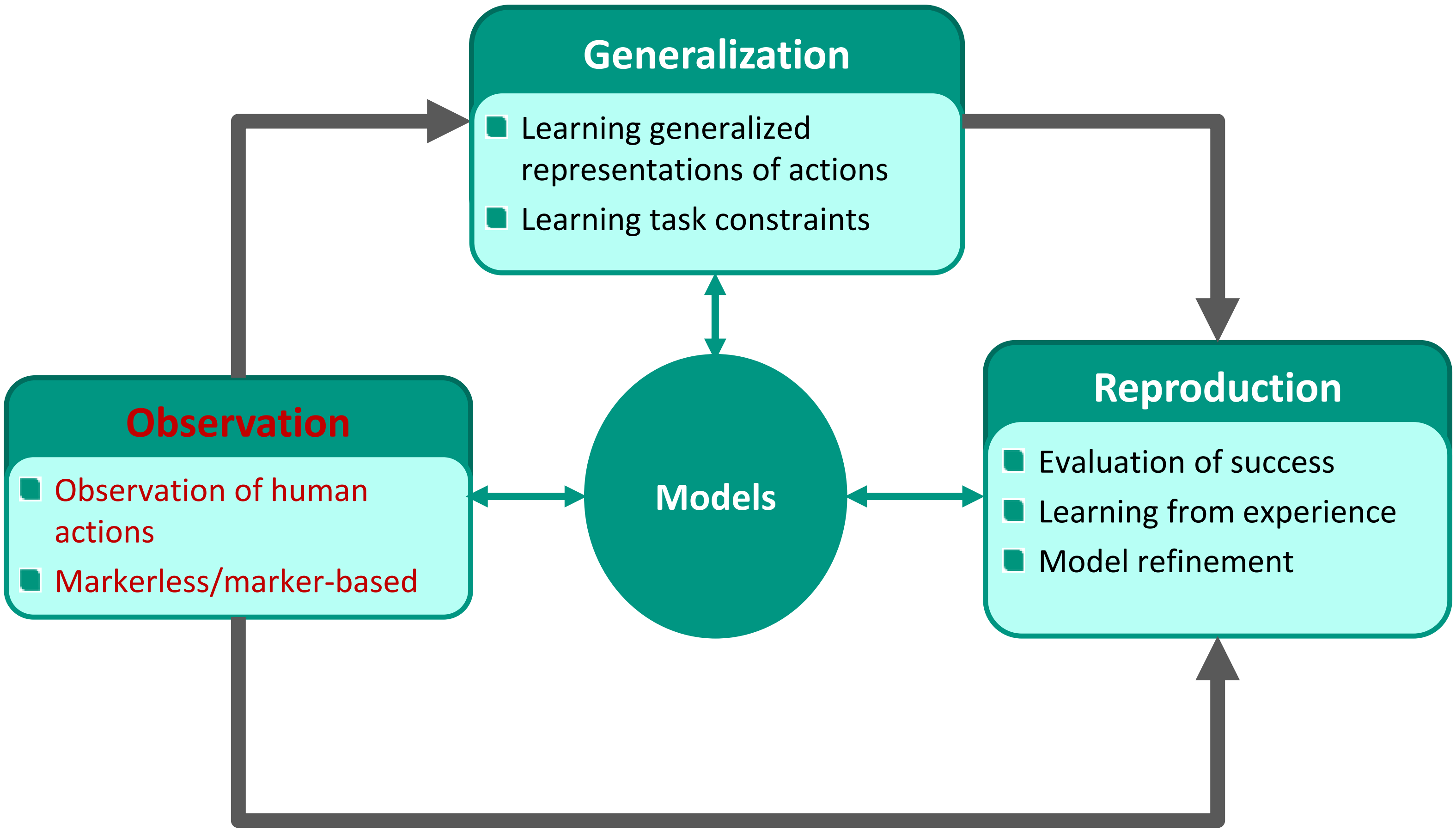
Proposal for a procedure

- Build a motion library by observing human‘s action
  - Elements of motion library: Motion Primitives
  - Focus on goal-oriented movements, consider other constraints, such as objects and forces ...
  - Methods of recording human movements (Motion Capture Techniques)
- Application of motion primitives to generate more complicated whole body movements
- **Key questions:** Representation and adaptation of motion primitives

Learning from human observation



Learning from human observation



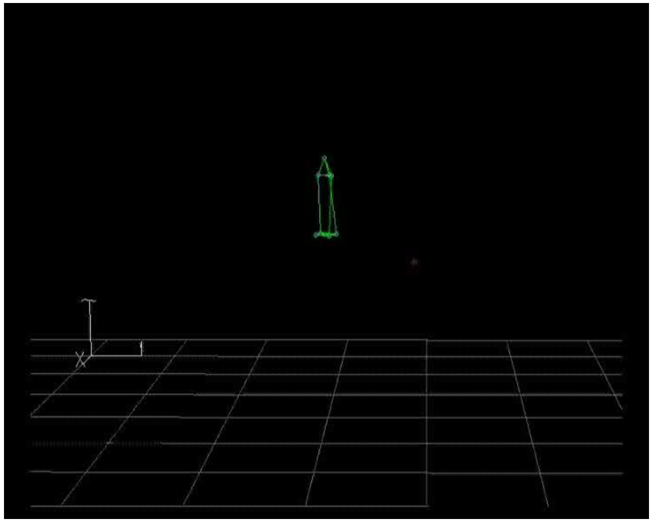
OBSERVATION

Observation

- Motion capture techniques
  - Marker-based
    - SFB 588
    - KIT Whole-body human motion database
    - Jessica Hodgins (CMU)
    - Ales Ude (JSI, ATR)
    - Ikeuchi (Tokyo University)
  - Markerless
    - Deutscher et al., Robotics Research Group, Oxford
    - Pedram Azad (KIT)
    - And many others

Observation

- Motion capture techniques
  - Marker-based
    - SFB 588
    - KIT Whole-body human motion database
    - Jessica Hodgins (CMU)
    - Ales Ude (JSI, ATR)
    - Ikeuchi (Tokyo University)
  - Markerless
    - Deutscher et al., Robotics Research Group, Oxford
    - Pedram Azad (KIT)
    - And many others



Observation

Karlsruhe Institute of Technology

- Motion capture techniques
  - Marker-based
    - SFB 588
    - KIT Whole-body human motion database
    - Jessica Hodgins (CMU)
    - Ales Ude (JSI, ATR)
    - Ikeuchi (Tokyo University)
  - Markerless
    - Deutscher et al., Robotics Research Group, Oxford
    - Pedram Azad (KIT)
    - And many others

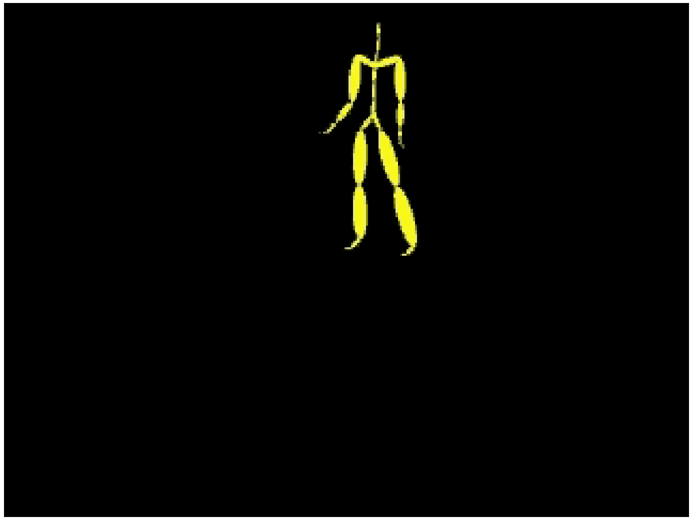
H2T

KIT Whole-body human motion database

https://motion-database.humanoids.kit.edu/

Observation

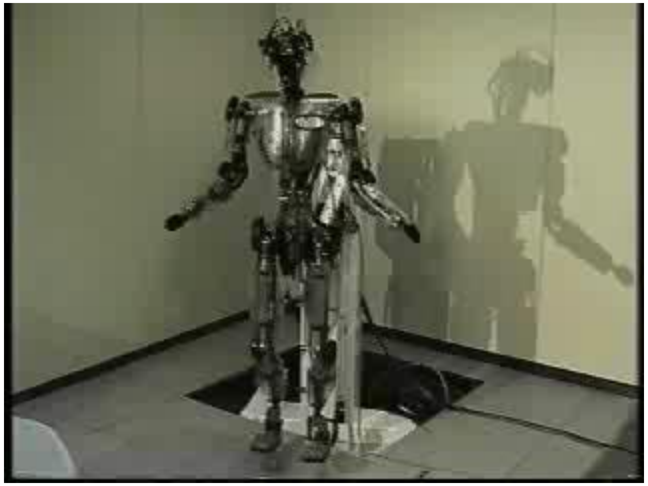
- Motion capture techniques
  - Marker-based
    - SFB 588
    - KIT Whole-body human motion database
    - Jessica Hodgins (CMU)
    - Ales Ude (JSI, ATR)
    - Ikeuchi (Tokyo University)
  - Markerless
    - Deutscher et al., Robotics Research Group, Oxford
    - Pedram Azad (KIT)
    - And many others



CMU Graphics Lab Motion Capture Database
http://mocap.cs.cmu.edu/

Observation

- Motion capture techniques
  - Marker-based
    - SFB 588
    - KIT Whole-body human motion database
    - Jessica Hodgins (CMU)
    - Ales Ude (JSI, ATR)
    - Ikeuchi (Tokyo University)
  - Markerless
    - Deutscher et al., Robotics Research Group, Oxford
    - Pedram Azad (KIT)
    - And many others



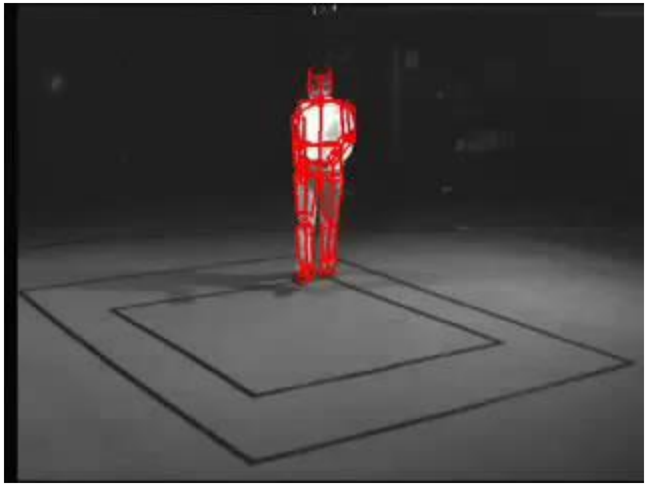
DB (Dynamic Brain), Kyoto, ATR, Japan

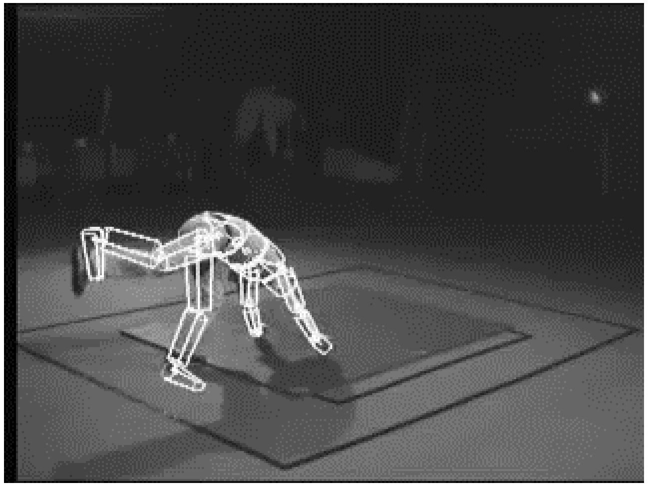
Observation

- Motion capture techniques
  - Marker-based
    - SFB 588
    - KIT Whole-body human motion database
    - Jessica Hodgins (CMU)
    - Ales Ude (JSI, ATR)
    - Ikeuchi (Tokyo University)
  - Markerless
    - Deutscher et al., Robotics Researc
      Group, Oxford
    - Pedram Azad (KIT)
    - And many others

Observation

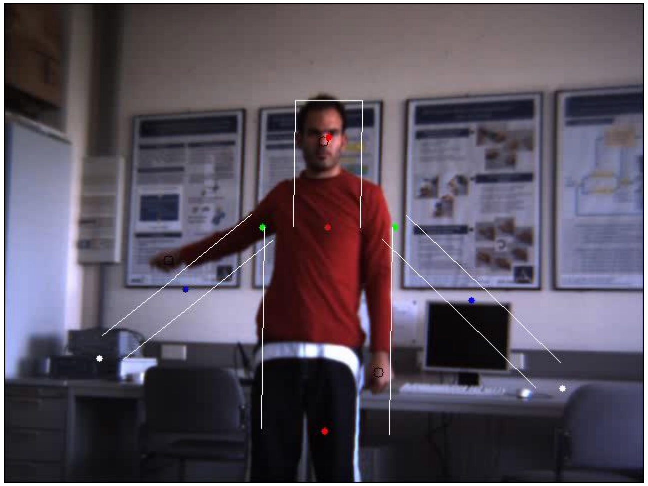
- Motion capture techniques
  - Marker-based
    - SFB 588
    - KIT Whole-body human motion database
    - Jessica Hodgins (CMU)
    - Ales Ude (JSI, ATR)
    - Ikeuchi (Tokyo University)
  - Markerless
    - Deutscher et al., Robotics Research Group, Oxford
    - Pedram Azad (KIT)
    - And many others

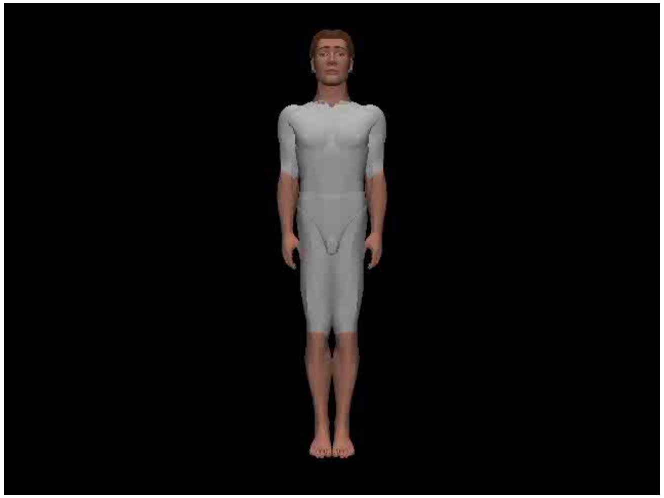




Observation

- Motion capture techniques
  - Marker-based
    - SFB 588
    - KIT Whole-body human motion database
    - Jessica Hodgins (CMU)
    - Ales Ude (JSI, ATR)
    - Ikeuchi (Tokyo University)
  - Markerless
    - Deutscher et al., Robotics Research Group, Oxford
    - Pedram Azad (KIT)
    - And many others

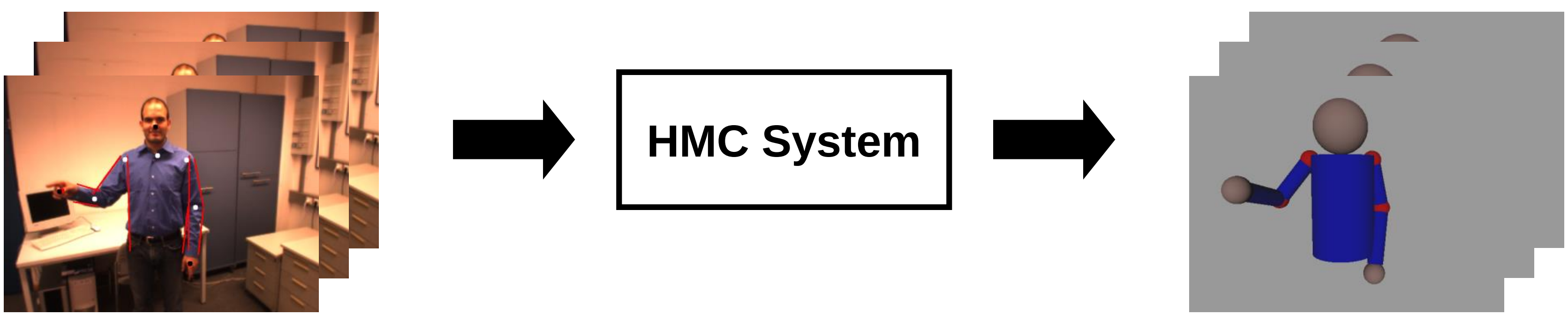




ICRA 2007

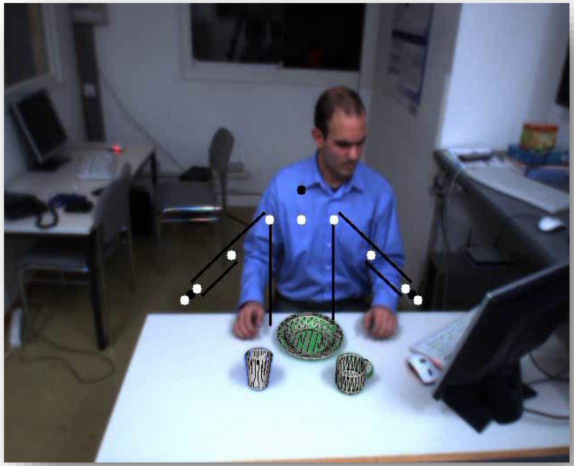
Markerless Human Motion Capture

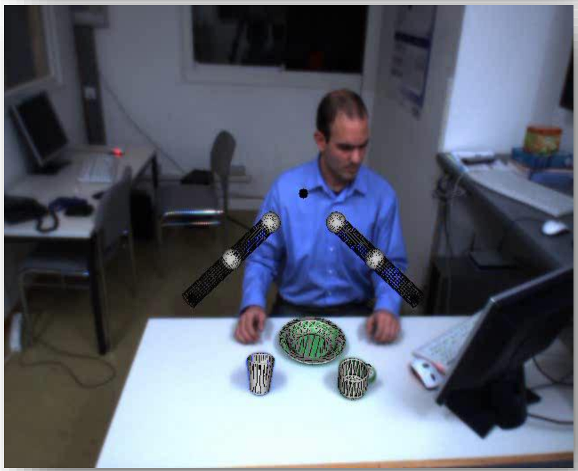
- Human Motion Capture (HMC):
  - The system operates on a simplified 3D human model
  - Output is a sequence of configuration vectors of this model, one for each frame
- Markerless:
  - The only input to the system is a sequence of stereo image pairs
  - No markers are used



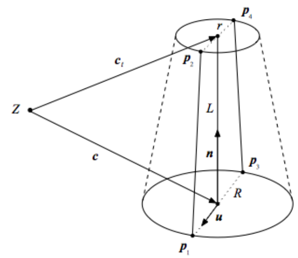
Stereo-based 3D Human Motion Capture (HMC)

- Capture 3D human motion based on the image input from the cameras of the robot‘s head only
- Approach
  - Hierarchical Particle Filter framework
  - Localization of hands and head using color segmentation and stereo triangulation
  - Fusion of 3d positions and edge information
  - Half of the particles are sampled using inverse kinematics
- Features
  - Automatic Initialization
  - 30 fps real-time tracking on a 3 GHz CPU, 640x440 images
  - Smooth tracking of real 3d motion





HMC Cues



- **Edge cue:**
  Operates on 2D image positions along the projected contour

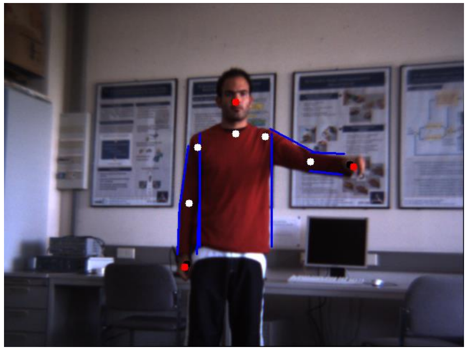
$$w_g(I_g, P) = 1 - \frac{1}{|P|} \sum_{i=1}^{|P|} I_g(\boldsymbol{p}_i)$$

- **Distance cue:**
  Operates on 3D positions of the hands and the head

$$w_d(I_d, P) = \sum_{i=1}^{|P|} |\boldsymbol{p}_i - \boldsymbol{p}'_i(I_d)|^2$$

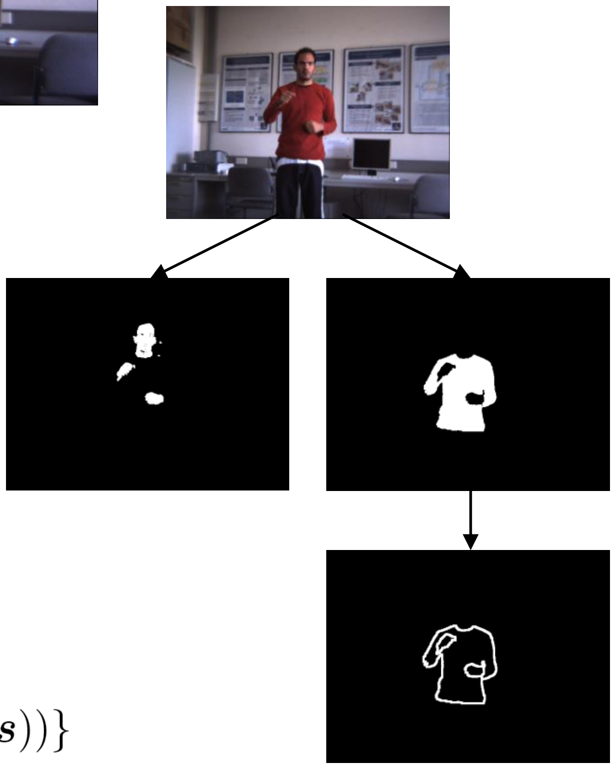
- **Basic fusion:**

$$p(I_g, I_d \,|\, \boldsymbol{s}) \propto \exp\{-s_g w_g(I_g, f_g(\boldsymbol{s}))\} \cdot \exp\{-s_d w_d(I_d, f_d(\boldsymbol{s}))\}$$



Model projection

Image processing pipeline



Related work

- M. Riley, A. Ude, and C. Atkeson, “Methods for motion generation and interaction with a humanoid robot: Case studies of dancing and catching”, 2000
  - Minimizing the difference between pre-captured markers and its position according to the configuration of the robot
  - No collision detection and avoidance

- N. Pollard, J. Hodgins, M. Riley, and C. Atkeson, “Adapting human motion for the control of a humanoid robot”, 2002
  - Using Vicon’s Bodybuilder software in order to map motions kinematically (upper-body gestures)
  - Various methods to constrain mapped motion
  - No collision detection and avoidance

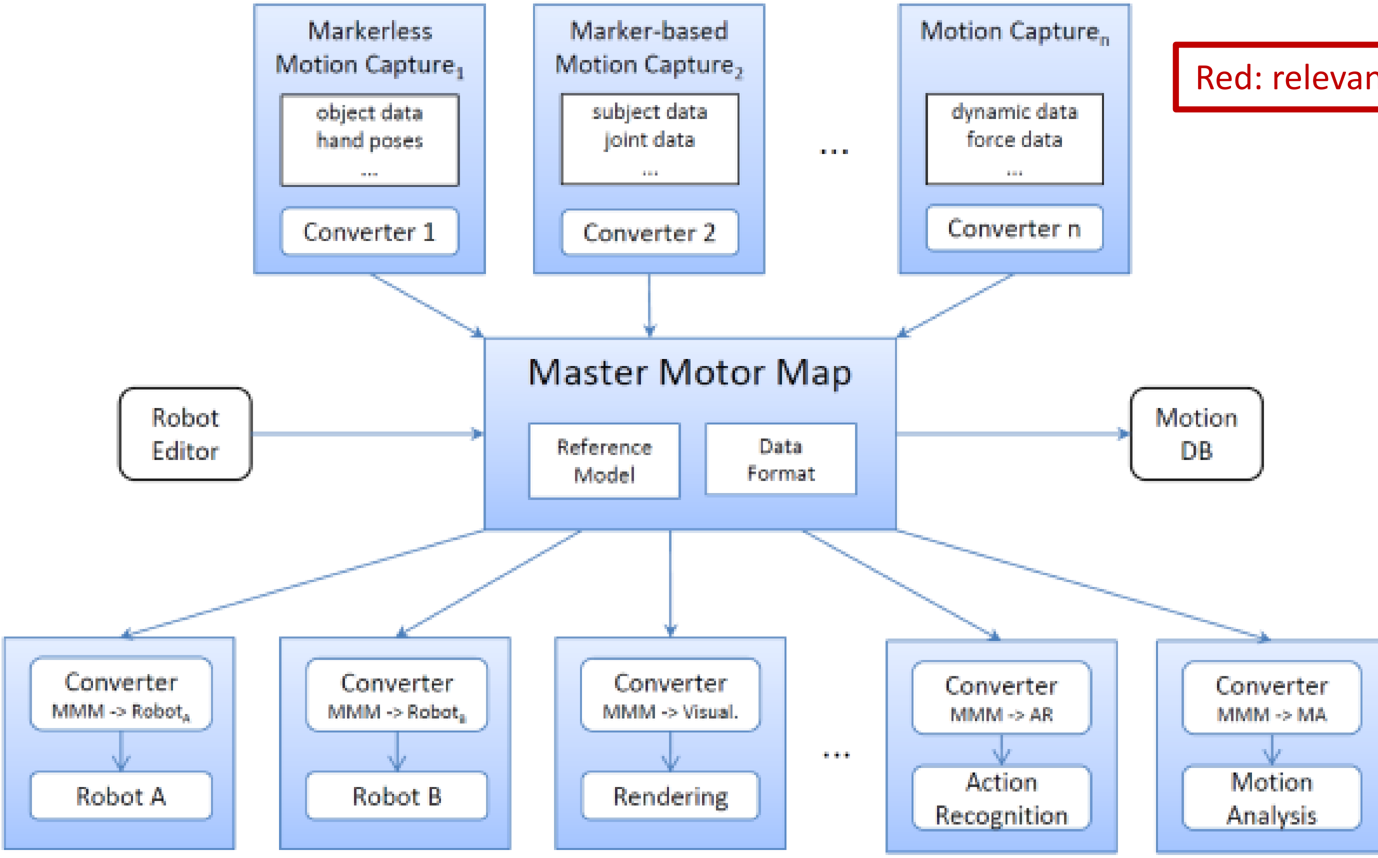
Related work

- A. Safonova, N. S. Pollard, and J. K. Hodgins, “Optimizing human motion for the control of a humanoid robot”, 2003
  - Based on Pollard et al., 2002
  - Objective function and optimization to preserve oscillations and the overall configuration
  - Collision avoidance is considered

- M. Do, P. Azad, T. Asfour, and R. Dillmann, “Imitation of human motion on a humanoid robot using nonlinear optimization”, 2008
  - System for the imitation of human motion
  - Using intermediate model to transfer motion to a robot
  - Similarity measure and optimization to generate trajectories
  - No explicit collision detection and avoidance

The Master Motor Map (MMM), see Chapter 2

- **Unifying framework** for capturing, representation, visualization and whole body human motion and mapping/converting to different embodiments

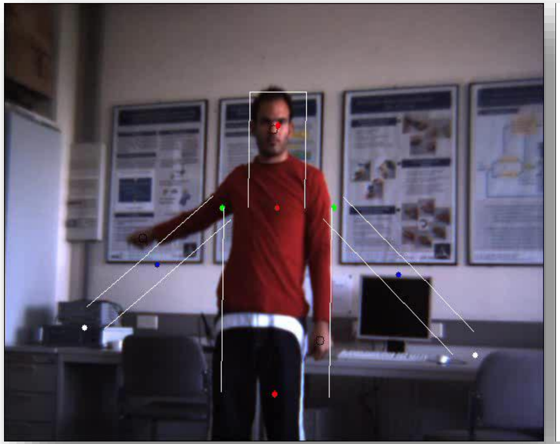


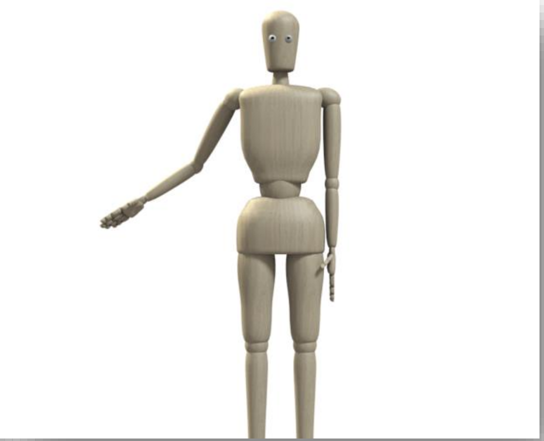
Red: relevant for the exam

T-RO 2016

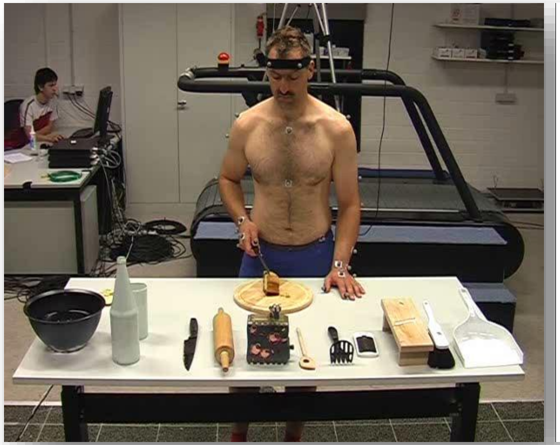
Motion reproduction using MMM

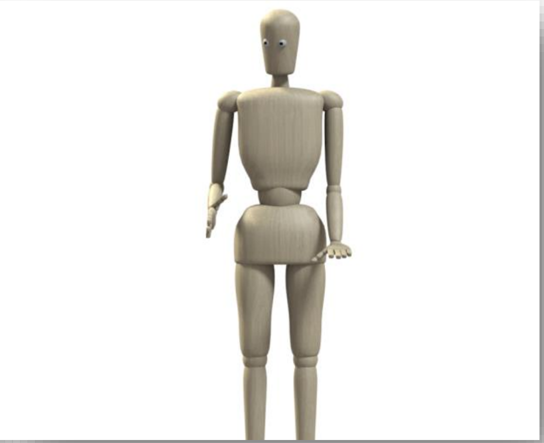
- Data from stereo-based markerless human motion capture system





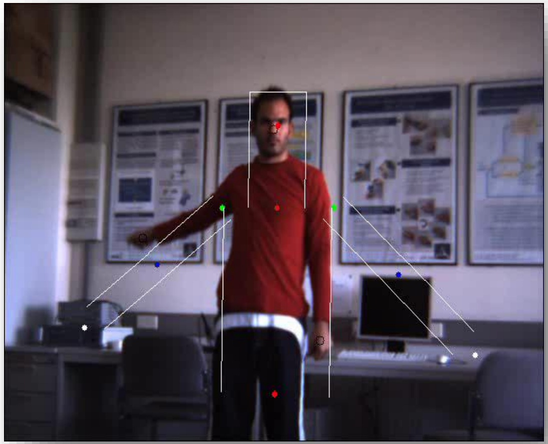
- Data from VICON system (SFB 588)

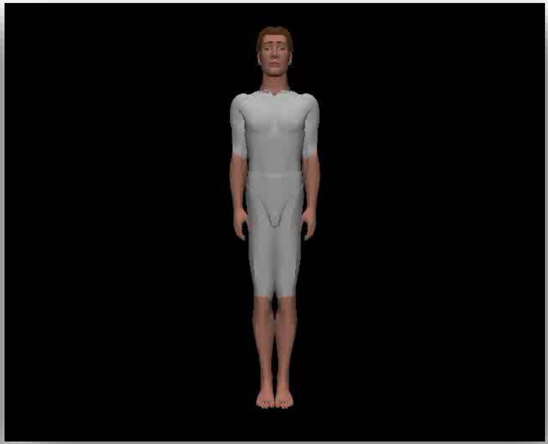




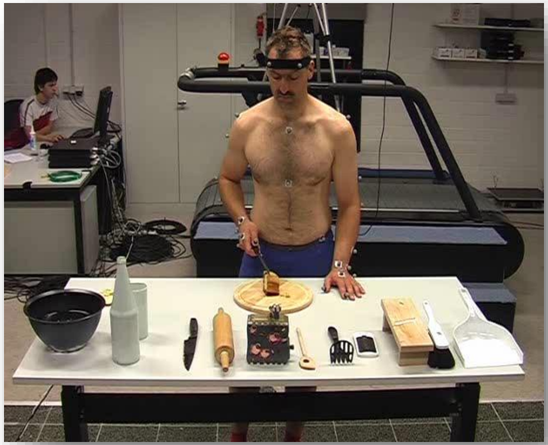
Reproduction using MMM

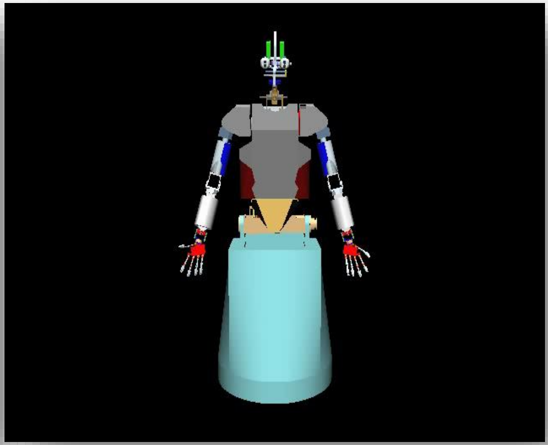
- Data from stereo-based markerless human motion capture system



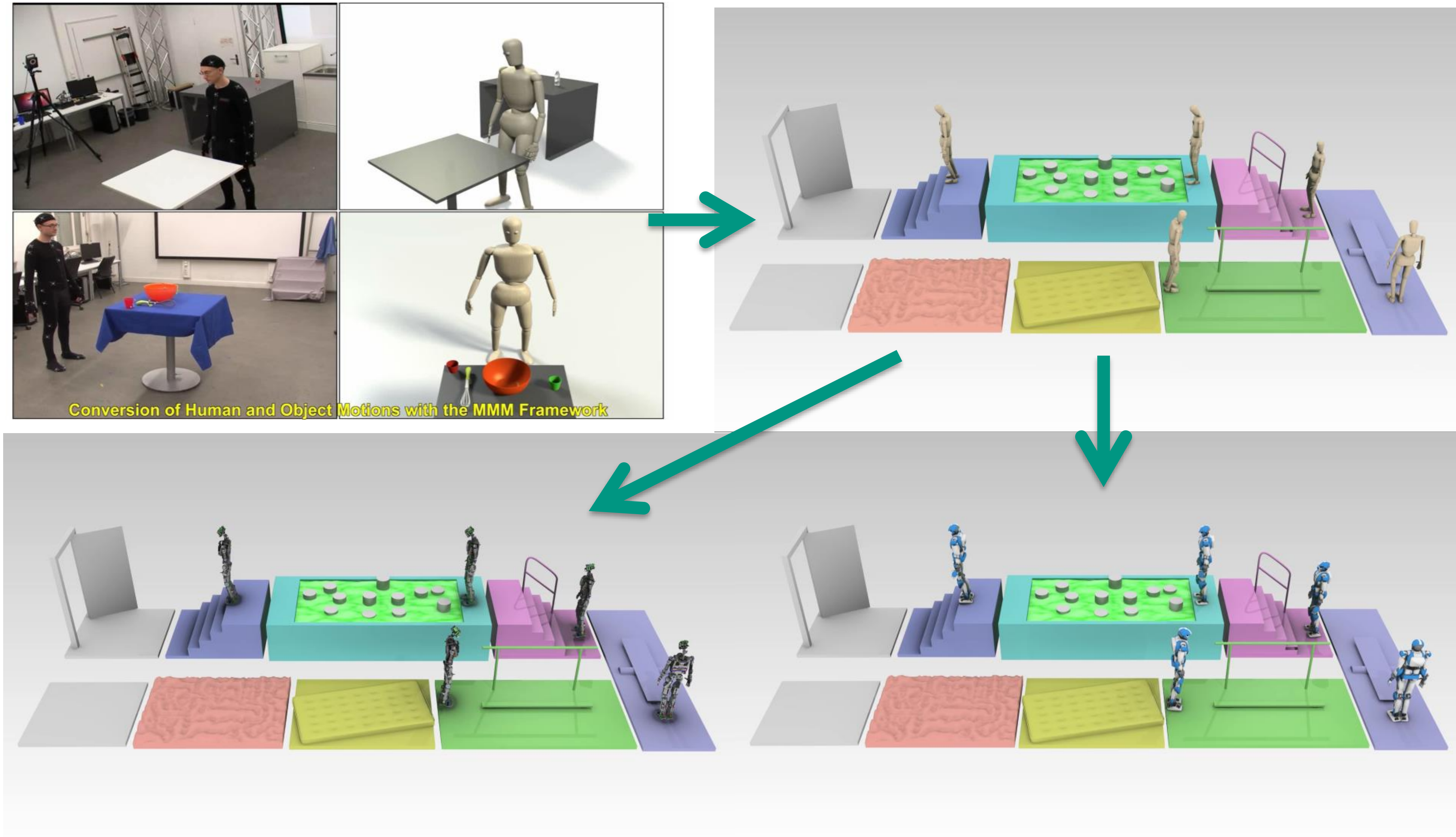


- Data from VICON system (SFB 588)

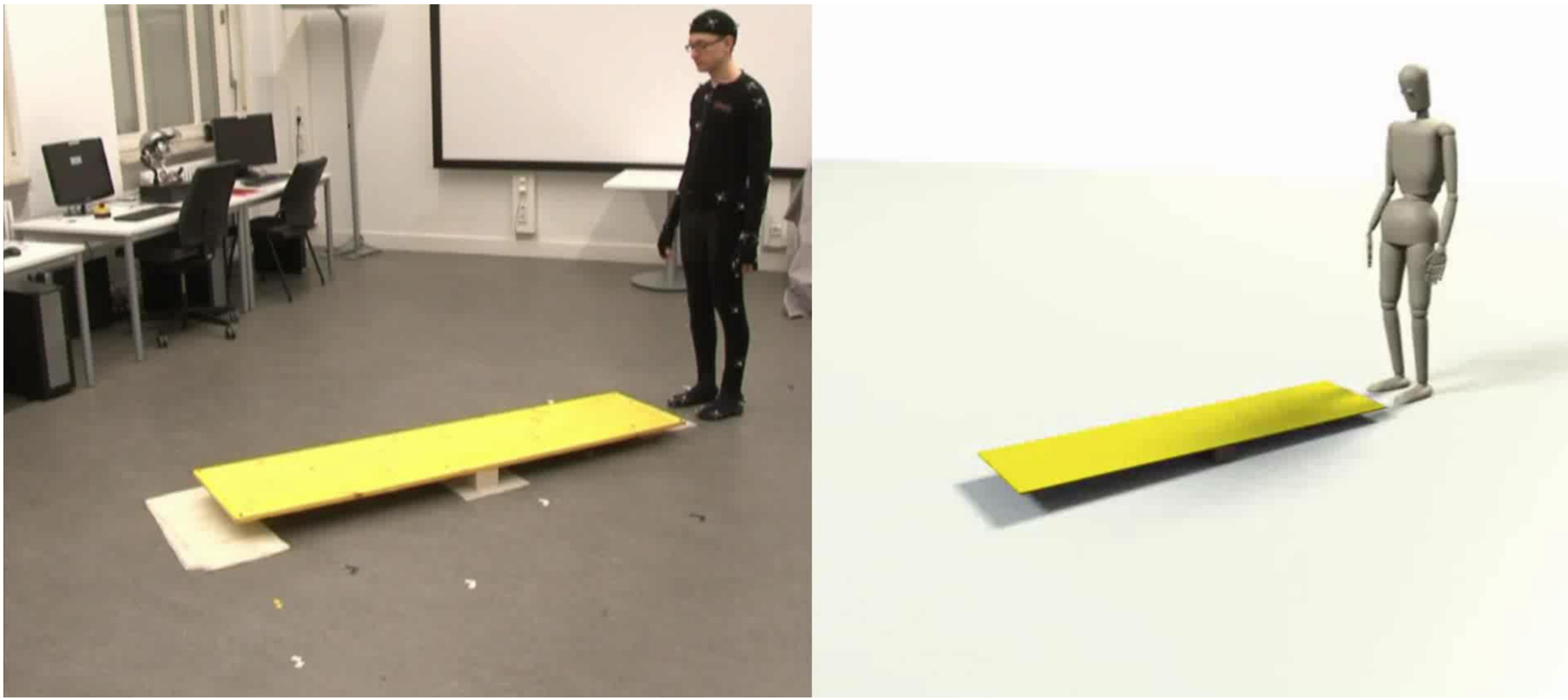




Reproduction using MMM



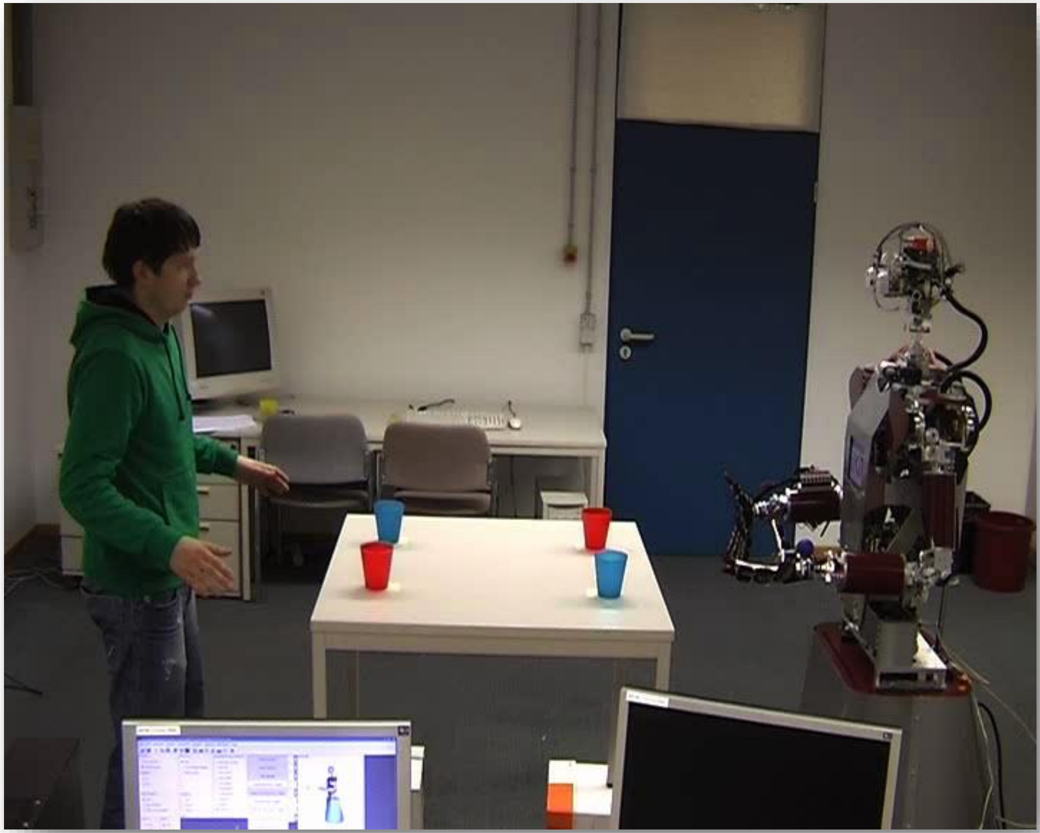
Motion Reproduction using MMM



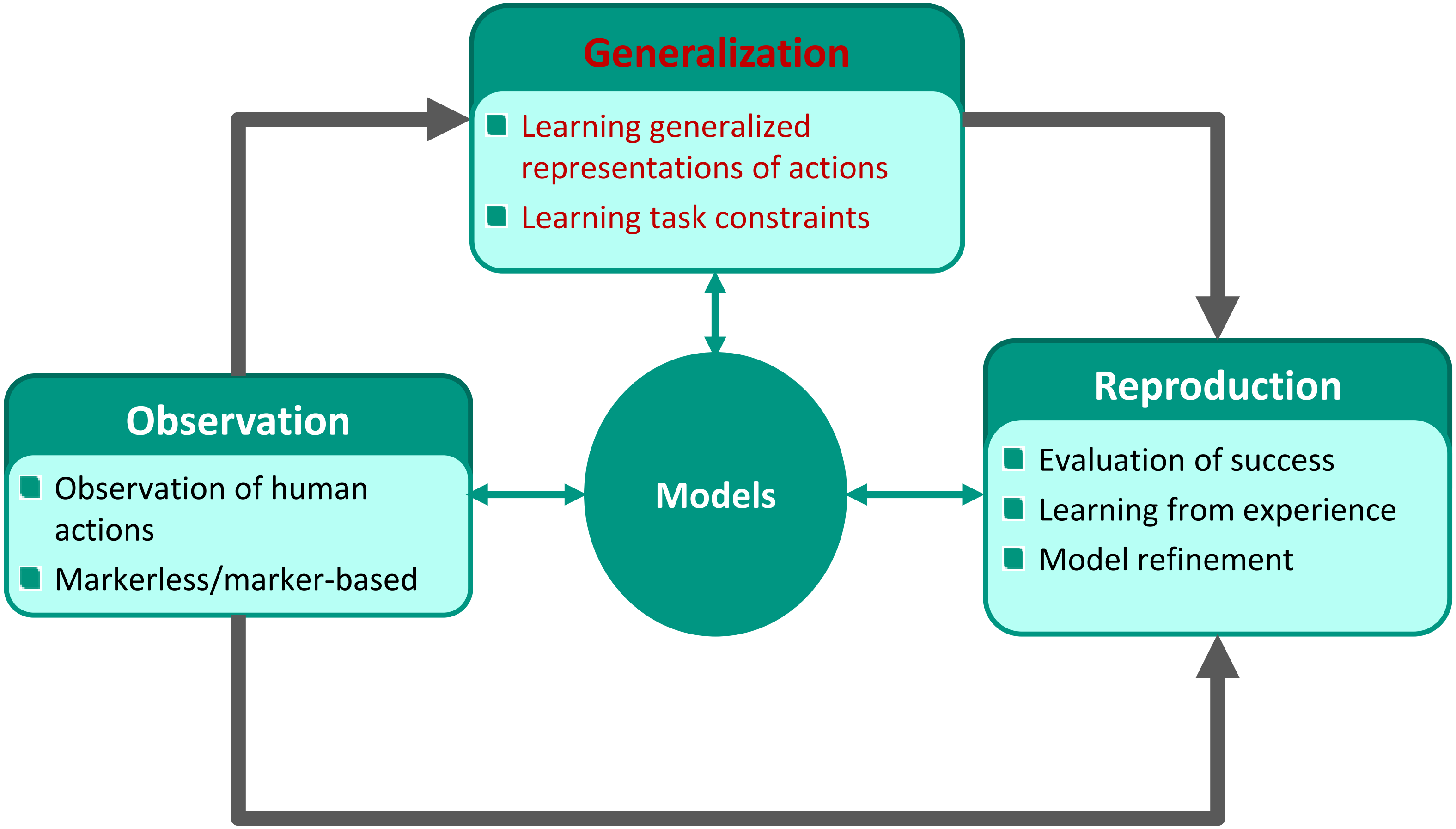
Reproduction on ARMAR

- Tracking of human and object motion
- Visual servoing for grasping

Generalization?



Learning from human observation



GENERALIZATION

Action representation

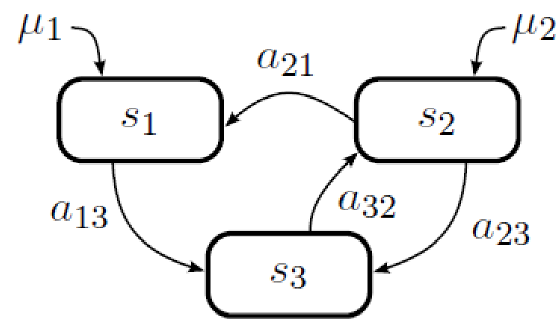
- Hidden Markov Models (HMM) Humanoids 2006, IJHR 2008
  - Extract key points (KP) in the demonstration
  - Determine key points that are common in multiple demonstrations (common key points: CKP)
  - Reproduction through interpolation between CKPs
- Dynamic movement primitives (DMP) ICRA 2009, T-RO 2010
  - Ijspeert, Nakanishi & Schaal, 2002
  - Trajectory formulation using canonical systems of differential equations
  - Parameters are estimated using locally weighted regression
- Spline-based representations Humanoids 2007
  - fifth order splines that correspond to minimum jerk trajectories to encode the trajectories
  - Time normalize the example trajectories
  - Determine common knot points so that all example trajectories are properly approximated. Similar to via-point, key-points calculation.

Action representation

- Hidden Markov Models (HMM) Humanoids 2006, IJHR 2008
  - Extract key points (KP) in the demonstration
  - Determine key points that are common in multiple demonstrations (common key points: CKP)
  - Reproduction through interpolation between CKPs
- Dynamic movement primitives (DMP) ICRA 2009, T-RO 2010
  - Ijspeert, Nakanishi & Schaal, 2002
  - Trajectory formulation using canonical systems of differential equations
  - Parameters are estimated using locally weighted regression
- Spline-based representations Humanoids 2007
  - fifth order splines that correspond to minimum jerk trajectories to encode the trajectories
  - Time normalize the example trajectories
  - Determine common knot points so that all example trajectories are properly approximated. Similar to via-point, key-points calculation.

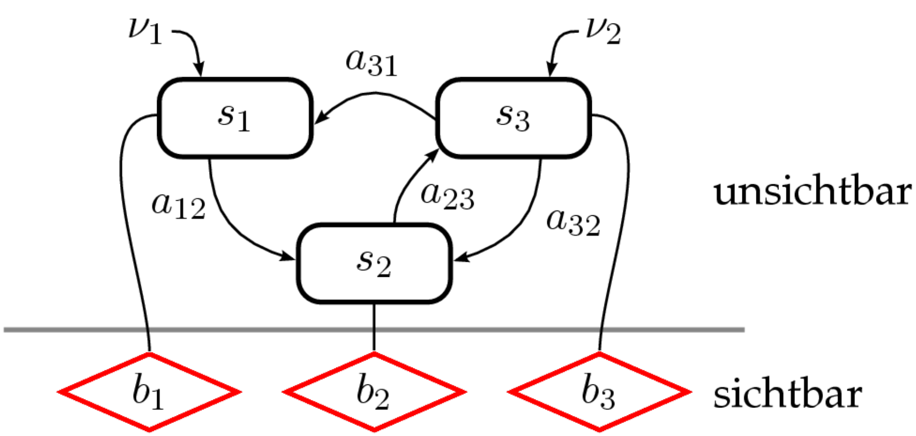
Hidden-Markov-Model

- HMMs are first order Markov chains, i.e. the subsequent state depends exclusively on the current state



- States are not observable, but we can observe the output they generate. Therefore "hidden“

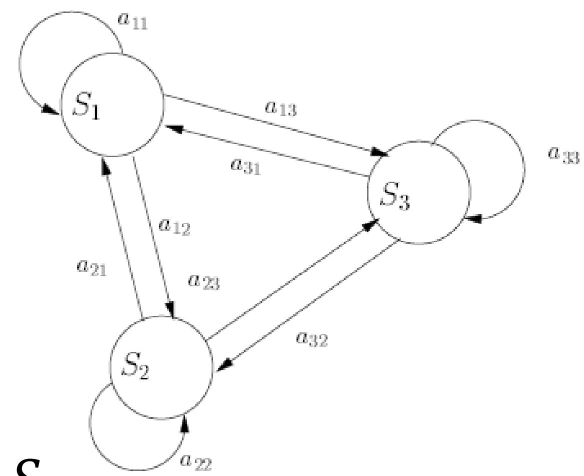
- In speech
  - sounds are known; syllables are hidden



Hidden-Markov-Model

- Suitable for the classification of time series data, such as speech or gestures signals

- One HMM contains:
  - States S_i
  - Transition probabilities a_{ij}
  - Start probabilities π_{ij}
  - Observation probabilities b_i for each state S_i
    - Discrete / continuous
    - In the continuous case,
      - Probability density with mean μ_i und covariance matrix U_i



Example: One day of a baby

- Markov-Chain: States set S = {wach, hungrig, schlafend}

A	$X_{i+1} = wach$	$X_{i+1} = hungrig$	$X_{i+1} = schlafend$
$X_i = wach$	0,5	0,4	0,1
$X_i = hungrig$	0,2	0,1	0,7
$X_i = schlafend$	0,3	0	0,7

μ	$X_0 = wach$	$X_0 = essend$	$X_0 = schlafend$
	0,7	0	0,3

- HMM: A neighbor does not know what the baby is doing

B	$ruhig$	$schreiend$
$X_i = wach$	0,5	0,5
$X_i = hungrig$	0,1	0,9
$X_i = schlafend$	1	0

Three Problems

Given a model λ and one observation sequence $O = O_1, \dots, O_n$

- How to efficiently calculate $P(O|\lambda)$, i.e. the probability that O is generated by λ?
- How to determine the most likely sequence of states which generated the observation sequence O ?
- How to find the the parameters λ that maximize $P(O|\lambda)$?

HMM Basic Algorithms

- **Forward-Algorithm:**
  Given an observation sequence $O = O_1, \dots, O_n$ and a model λ.
  How to efficiently compare $P(O|\quad)$, the probability of the observation sequence, given the model?

- **Viterbi-Algorithm:**
  Given an observation sequence $O = O_1, \dots, O_n$ and a model λ.
  How to find a corresponding state sequence $S = S_1, \dots, S_n$ which is optimal in some sense (e.g. best “explain” the observation sequence

- **Baum-Welch-Algorithm:**
  How to adjust the model parameters λ (training of the HMM) to maximize $P(O|\lambda)$.

Approach with Hidden Markov Models

Model-based Imitation of arm movements

- Approach with Hidden Markov Model (HMMs)
- Perception and analyse
- Generalization
- Reproduction

T. Asfour et al. (2006, 2008). Imitation Learning of Dual-Arm Manipulation Tasks in Humanoid Robots. International Journal on Humanoid Robots, 2008. (International Conference on Humanoid Robots, 2006)

Approach

- Imitation learning process based on multiple demonstrations
- HMM for recognition and reproduction of the motion
- Analysis of trajectories for characteristic points(„**key points**“)
- Reproduction on the kinematic model of Humans‘

Red: relevant for the exam

T. Asfour et al. (2006, 2008). Imitation Learning of Dual-Arm Manipulation Tasks in Humanoid Robots. International Journal on Humanoid Robots, 2008. (International Conference on Humanoid Robots, 2006)

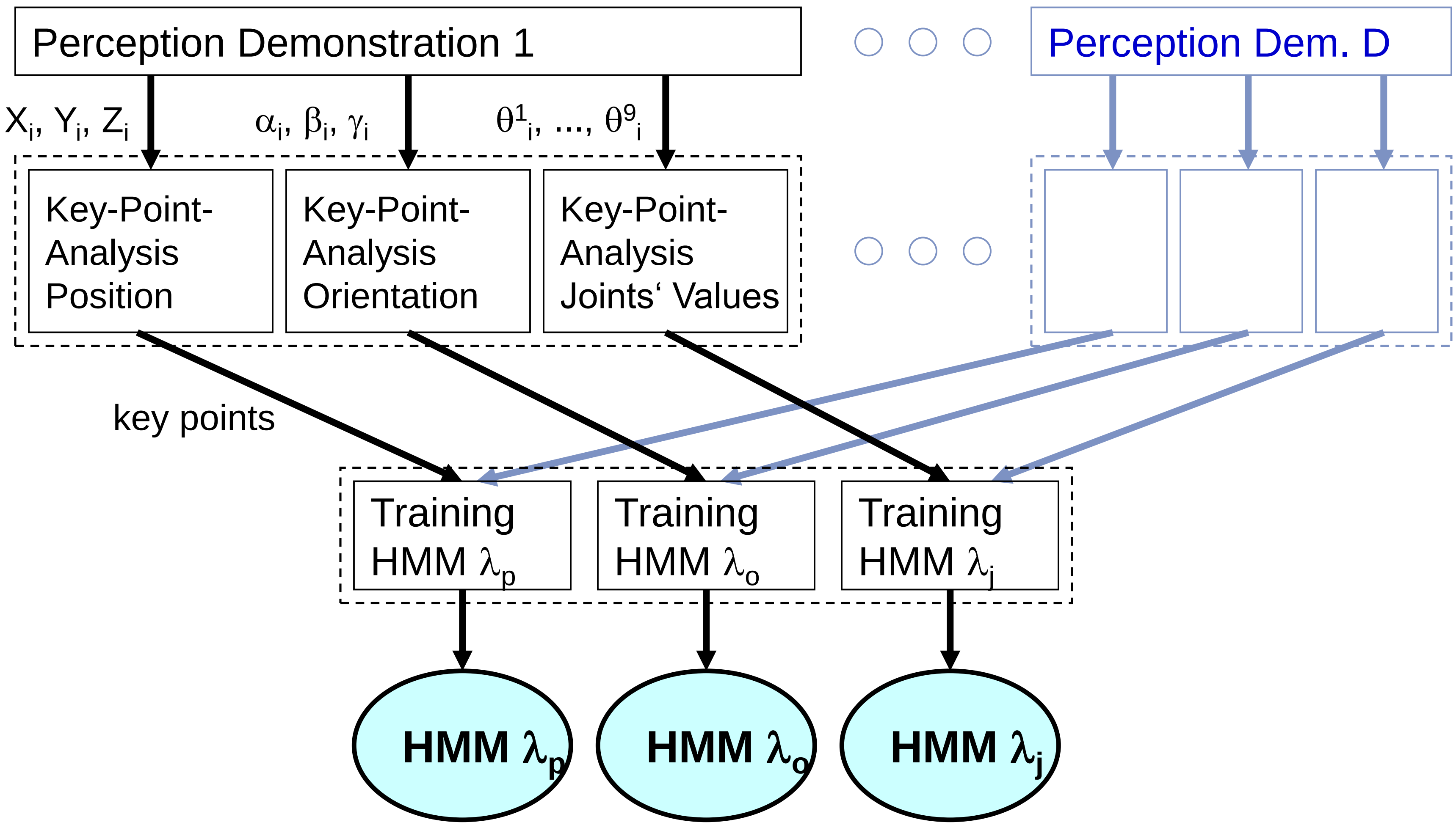
Approach (2)

- Both joint angles as well as positions and orientations of the hand are recorded and three different HMMs are trained.

- Model of a human arm with 9 DOF

- Depending on the priority - imitation of the exact TCP trajectory or, if possible, arm positions - the influence of the respective HMM on the reproduction can be controlled by a set of weighting factors

Analysis in Detail : Training HMMs

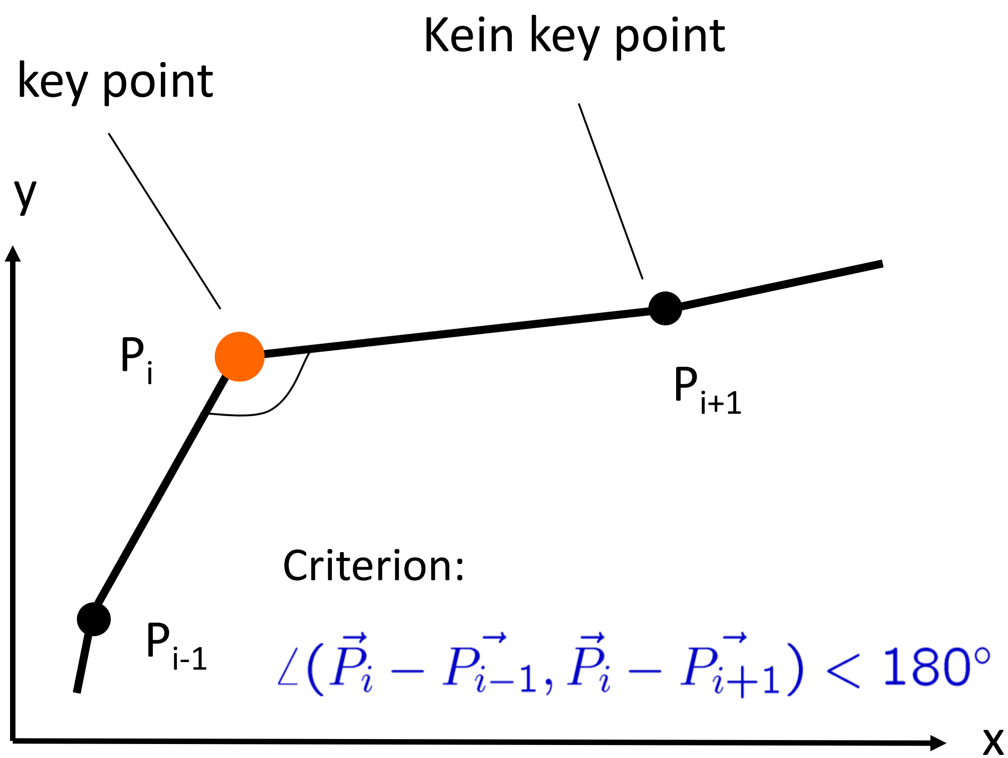


Key Points

- Key points are prominent/distinctive points of a movement

- Purpose to determine these points:
  - Data reduction (important for HMM)
  - Movement is represented exclusively by characteristic points
    - ➔ Better comparability of trajectories

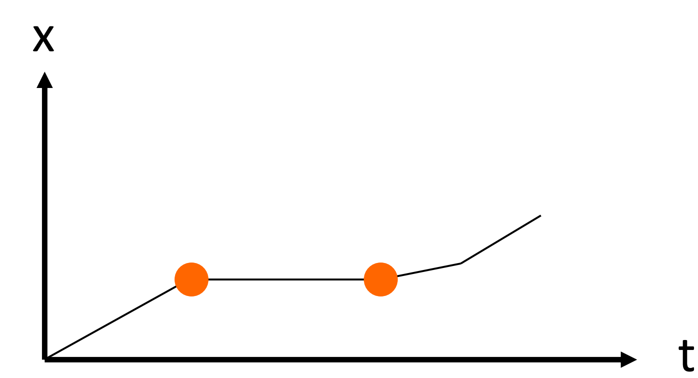
Key Points



Criterion:

$$\angle(\vec{P_i} - \vec{P_{i-1}}, \vec{P_i} - \vec{P_{i+1}}) < 180^\circ - \epsilon$$

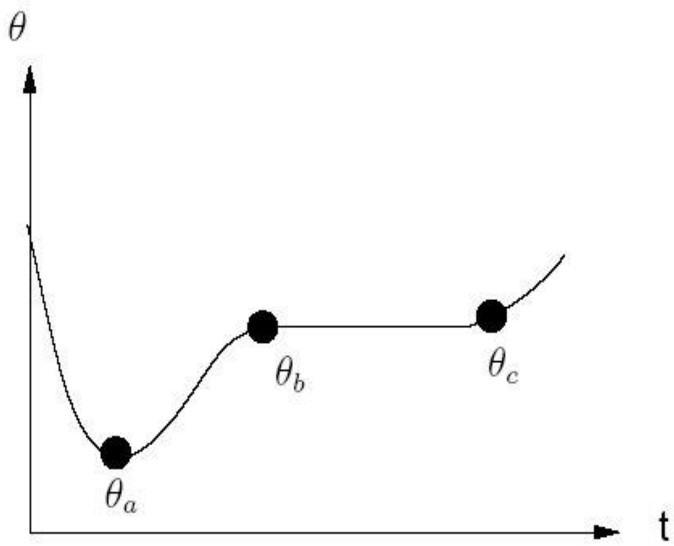
Hand-Trajectory (x/y-Coordinate)



Hand-Trajectory (x-Coord.)

Joint values: local minima and maxima and also pauses

Key Points



Joints - Trajectory

Local minimum and maximum and also pauses

Reproduction with HMM

- HMM is trained by multiple demonstrations;
- Continuous HMM: Output probabilities are calculated from probability densities of normal distribution
- After the training, mean values of these distributions can be calculated to form an "average" trajectory
- Which states should be considered for the reproduction?

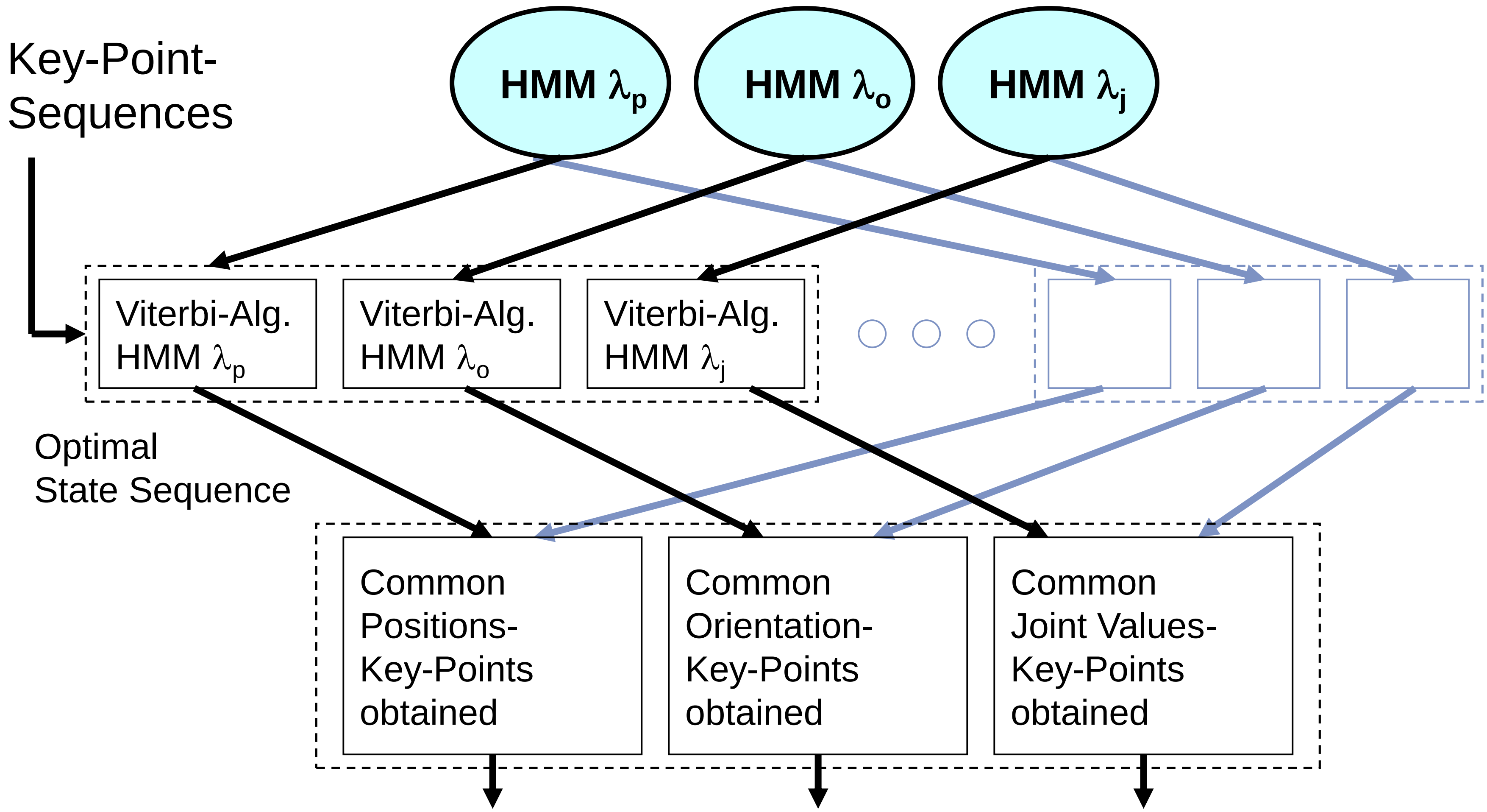
Common Key Points

- Relevant to a generalized movement: Key Points that occur in (almost) all demonstrations

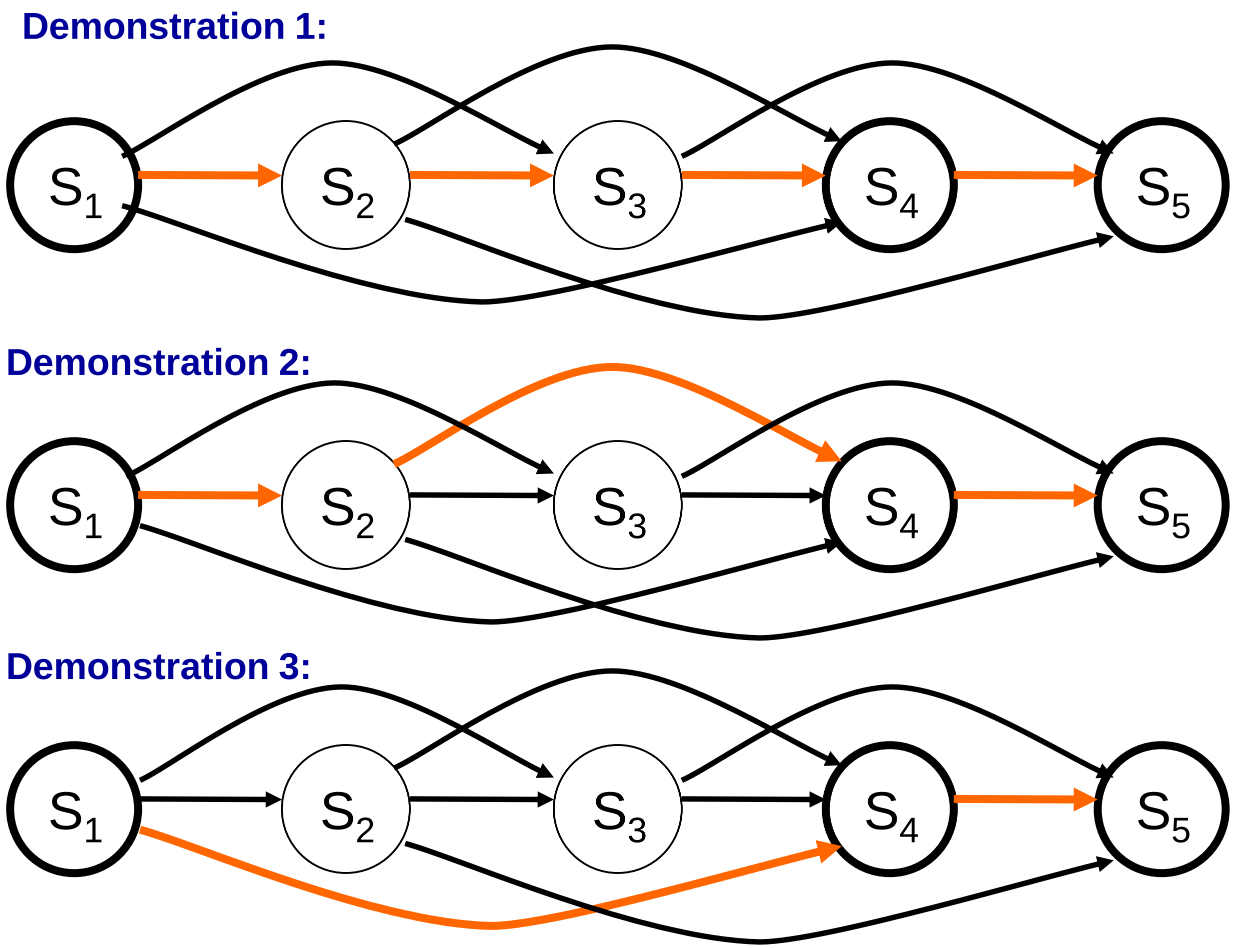
- Similar key points from the various demonstrations are **common key points** $C_i = (\kappa_1, \ldots, \kappa_D)$

- How to find common key points?
  - Usually used for such problems: Dynamic Programming Matching (Dynamic-Time-Warping, DTW)
  - Alternative: HMM
    - Viterbi algorithm provides optimal state sequences for the key-point sequences of the various demonstrations
    - Common key points are obtained from states that occur in all sequences

Common Key Points

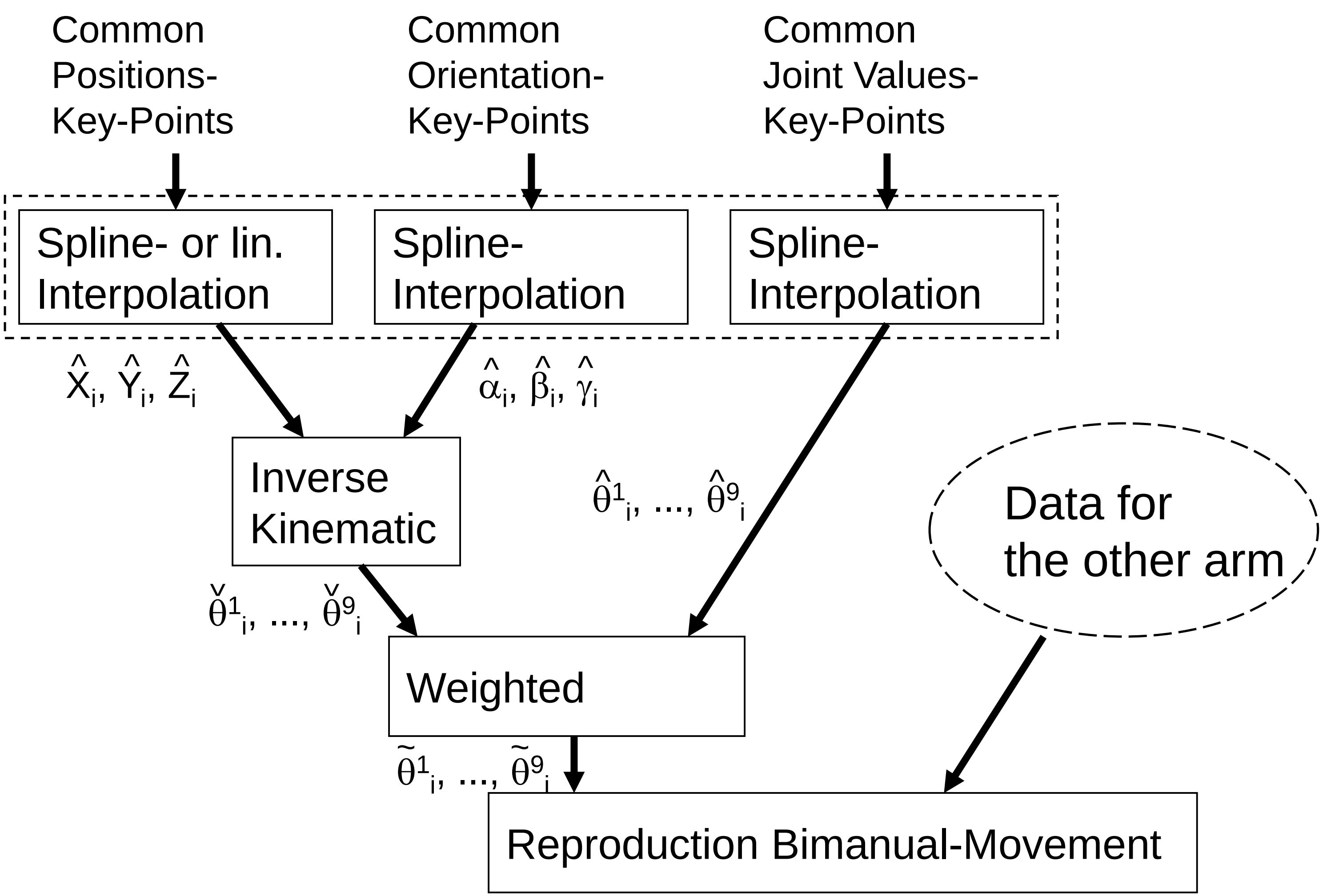


Common Key Points (Example)



Only the states S_1, S_4 and S_5 represent common key points

Rreproduction Phases in Detail (2)



Generalization/Reproduction

- Interpolation between common key points (separately for Positions-, Orientations- und Joint Values - CKP)
- Subsequently, inverse kinematics, from position and orientation to joint angle $\check{\theta}^j_i$
- Reproduction:

$$\tilde{\theta}_i^j = \omega \cdot \hat{\theta}_i^j + (1-\omega) \cdot \check{\theta}_i^j \qquad \omega \in [0,1]$$

Joints

IK

Temporal Coordination (1)

- Irrespective of the approach presented here, interesting question:

  What temporal relations must be taken into account when imitating a two-arm movement?

- Example: Pour (with two arms)

Temporal Coordination (2)

In the case of several demonstrations, the common key points just described can be used:

- For each common key point $C_i = (\kappa_1, ..., \kappa_D)$, a vector of time stamps of the individual key points can be specified: $(\tau_1, ..., \tau_D)$

- Common key point C_i of the right arm must be reached before common key point C_j of the left arm, if we have (k is the number of the demonstraions):

$$\forall k : \left(\tau_i^k < \tau_j^k \right)$$

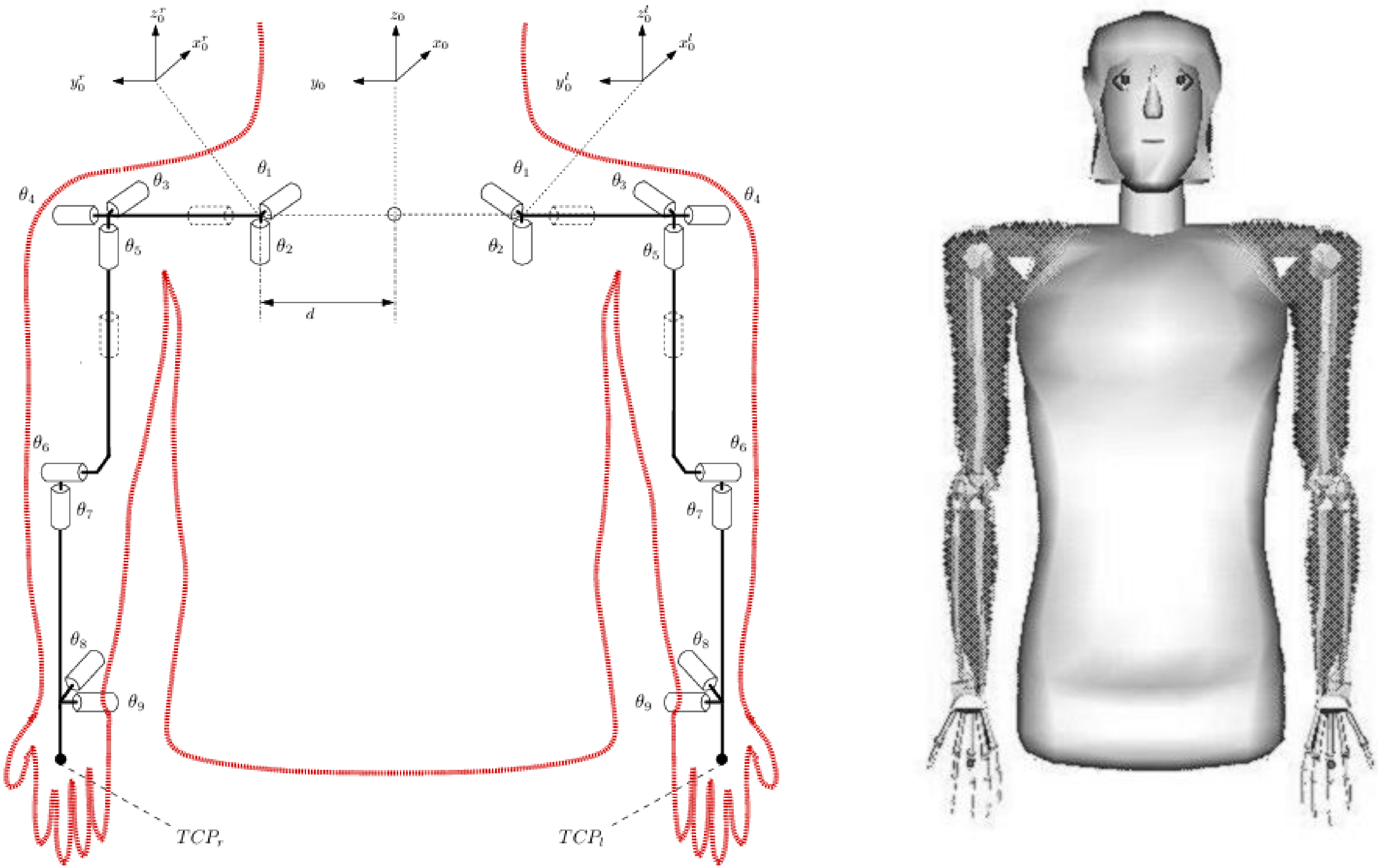
  (and the same for the other direction)

Reproduction: The Kinematic Model (1)

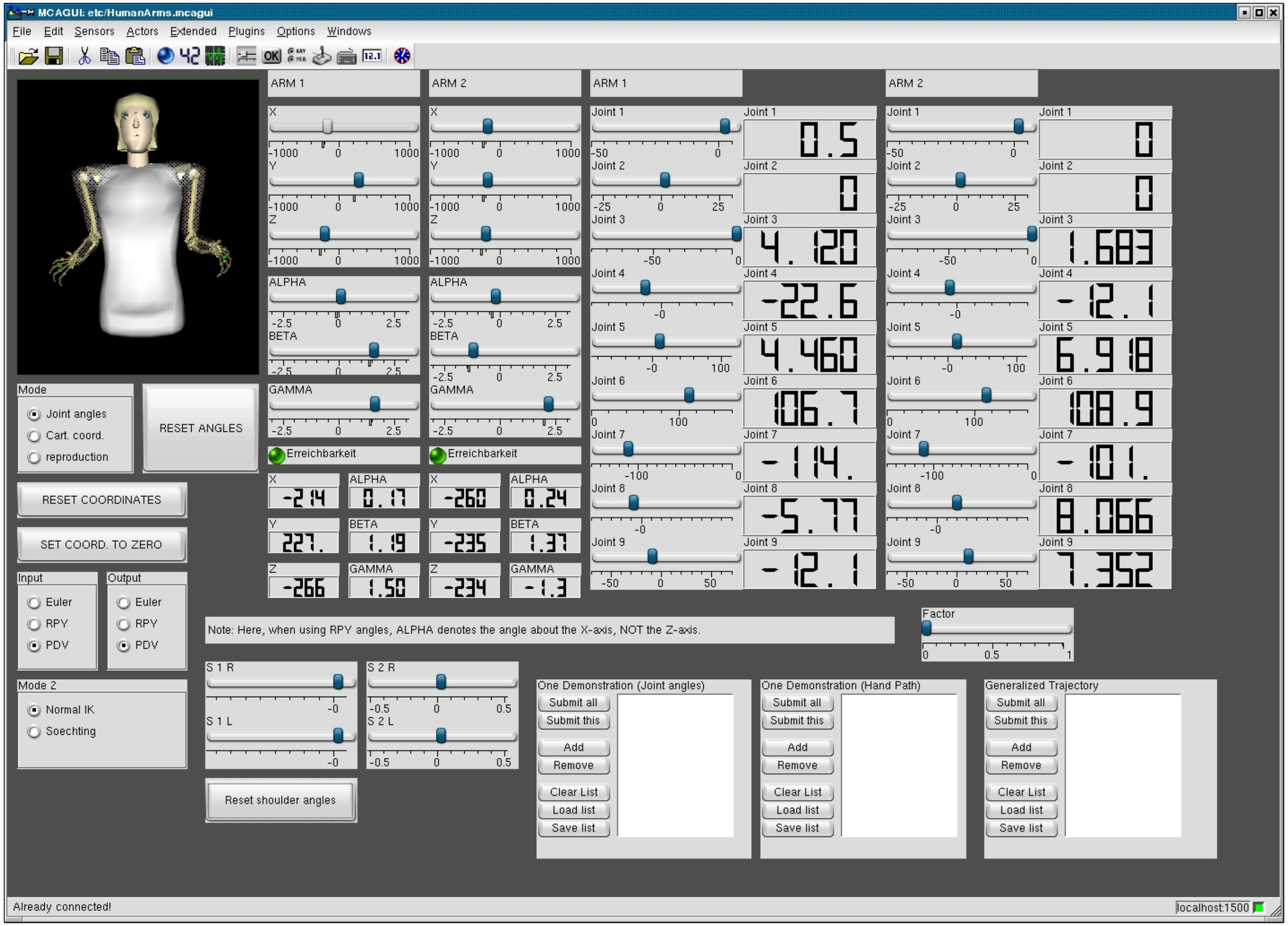
- Early version of MMM

- Features:
  - Each arm has 9 DOFs
  - Model directly calculated + inverse Kinematic
  - Rotation is represented by Euler-Angle or RPY-Angle
  - Implemented in MCA
  - Visualization with OpenInventor

Reproduction: The Kinematic Model (2)



Reproduction: MCA-User Interface



Problem: No Joint Angles

- Problem: Joint angles were not present (in 2006), which are necessary for the approach!
- Solution?
  - Now: KIT human motion database and MMM
  - In 2006: How to create human-like joint angles from end-effector trajectories?

Sensorimotor transformation model for human-like arm positions according to Soechting and Flanders

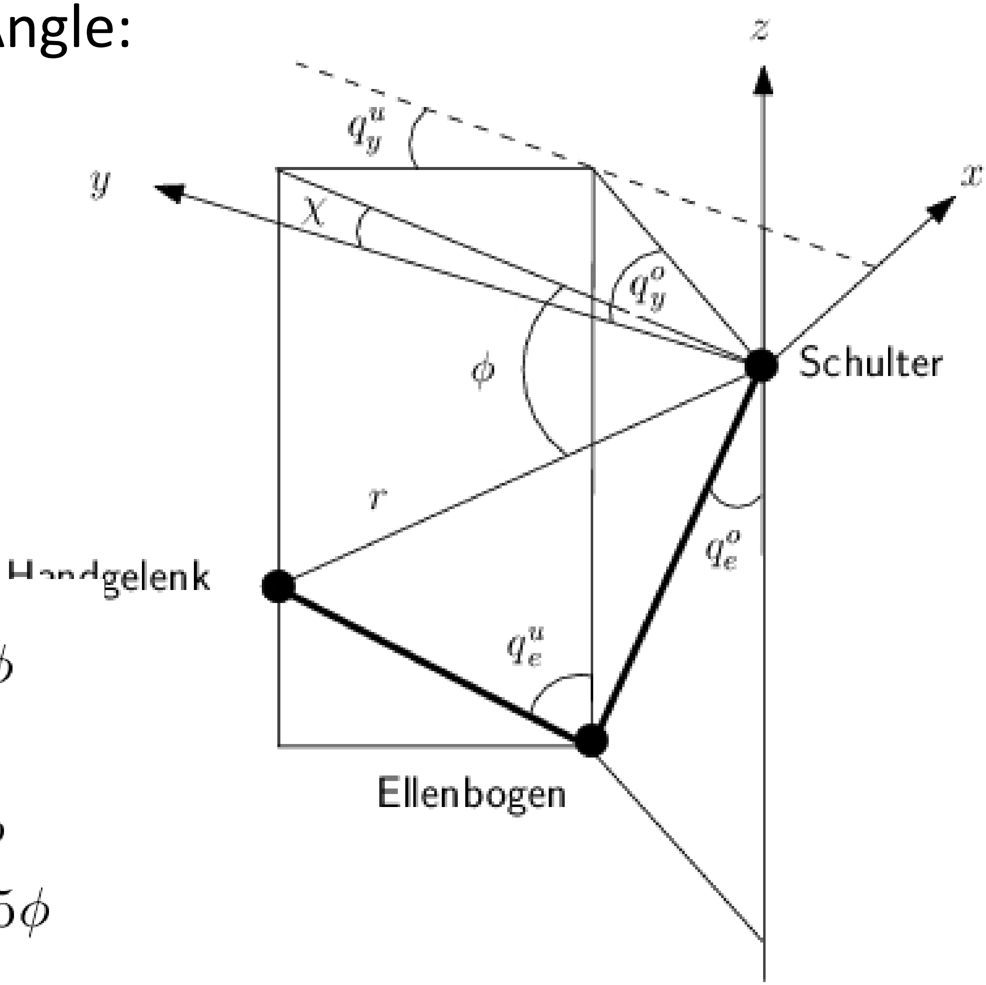
- J.F. Soechting, M. Flanders. Errors in Pointing are Due to Approximations in Targets in Sensorimotor Transformations. Journal of Neurophysiology, 62(2):595-608, 1989.
- J.F. Soechting, M. Flanders. Sensorimotor Representations for Pointing to Targets in Three-Dimensional Space. Journal of Neurophysiology, 62(2):582-594, 1989.

Soechting-Angle (1)

- Calculation of Soechting-Angle:

$$
\begin{aligned}
r^2 &= x^2 + y^2 + z^2 \\
tan\ \chi &= \frac{x}{y} \\
tan\ \phi &= \frac{z}{\sqrt{x^2 + y^2}}
\end{aligned}
$$

$$
\begin{aligned}
q_e^o &= -4.0 + 1.10r + 0.90\phi \\
q_e^u &= 39.4 + 0.54r - 1.06\phi \\
q_y^o &= 13.2 + 0.86\chi + 0.11\phi \\
q_y^u &= -10.0 + 1.08\chi - 0.35\phi
\end{aligned}
$$



Soechting-Angle (2)

How to get angles from the soechting angles?

- Set θ_1 and θ_2 to 0
- Calculate θ_3 - θ_6 from Soechting-Angles:

$$\theta_3 = \arcsin\left(\frac{\sin(q_e^o) \cdot l_o \cdot \sin(q_e^u)}{\cos(\theta_4) \cdot l_o}\right)$$

$$\theta_4 = \arcsin\left(\frac{\sin(q_e^o) \cdot l_o \cdot \cos(q_e^u)}{l_o}\right)$$

$$\begin{aligned}\theta_6 \; &= \arccos(\sin(q_e^o) \cdot \sin(q_y^o) \cdot \sin(q_e^u) \cdot \sin(q_y^u) + \\ &\quad \sin(q_e^o) \cdot \cos(q_y^o) \cdot \sin(q_e^u) \cdot \cos(q_y^u) - \cos(q_e^o) \cdot \cos(q_e^u))\end{aligned}$$

Soechting-Angle (3)

$$\vec{p} = (\vec{l_u} \times \vec{l_o}) \times \vec{l_o}$$

$$\vec{q} = (\vec{l_{u'}} \times \vec{l_o}) \times \vec{l_o}$$

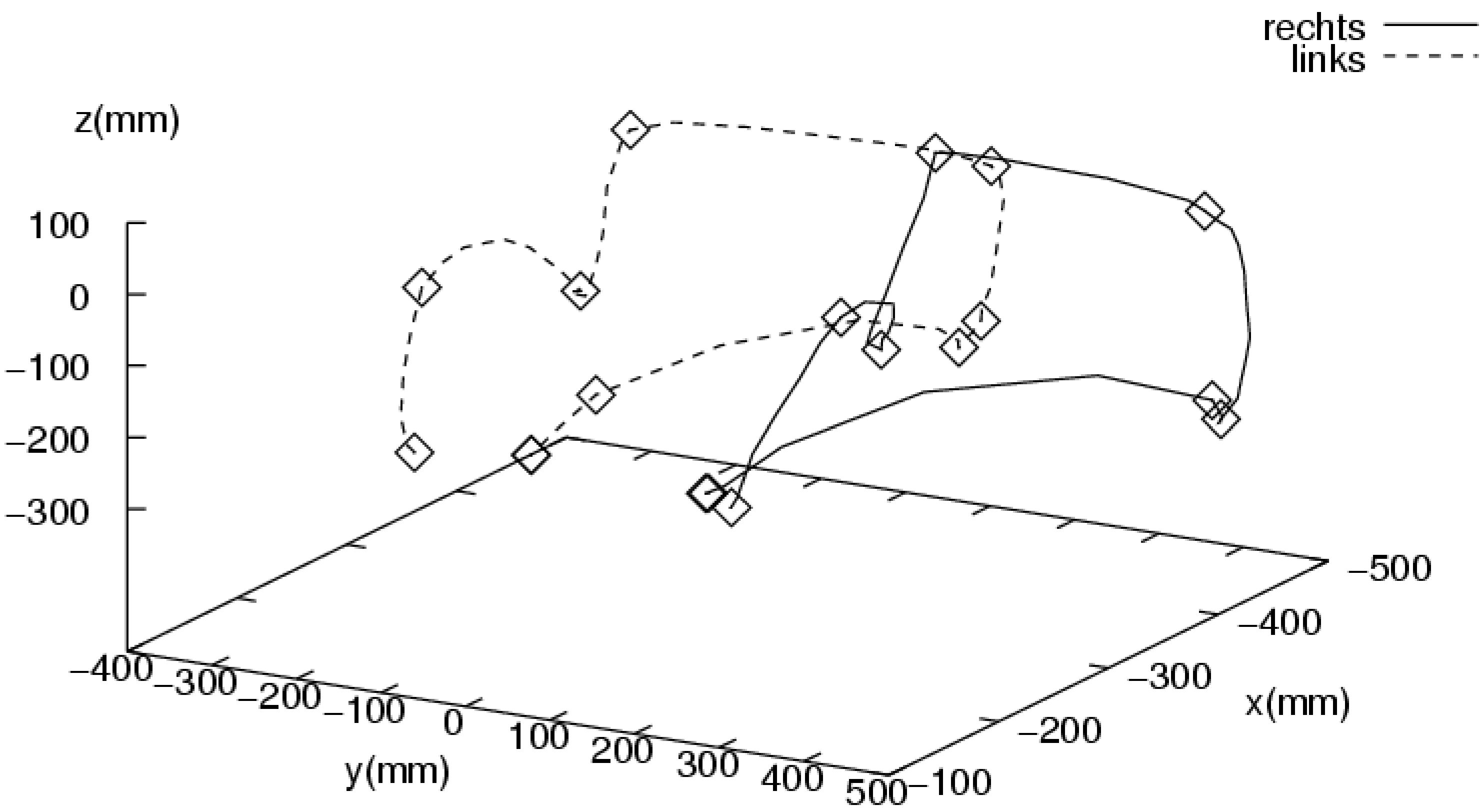
$$\theta_5 = \arccos\left(\frac{\langle \vec{p}, \vec{q} \rangle}{||\vec{p}|| \cdot ||\vec{q}||}\right)$$

- Joint angle θ_7 - θ_9 can be calculated by inverse kinematics. If the target position is not reached, θ_3 - θ_6 are changed step by step until an acceptable solution is found.

Experiment (1)

Sample demonstration pick-and-place with box

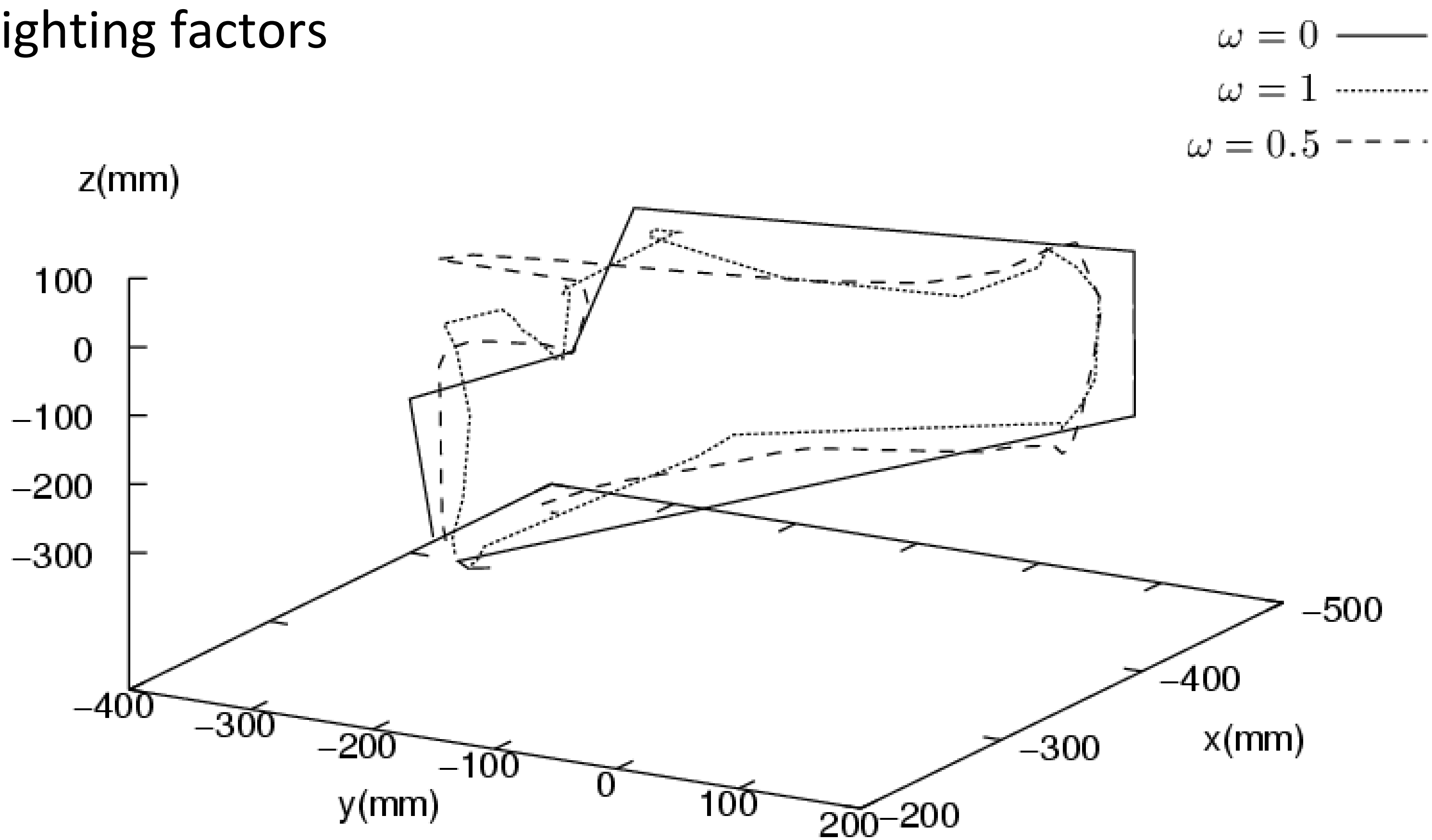
(Key Points are also drawn)



Experiment (2)

$$\tilde{\theta}_i^j = \omega \cdot \hat{\theta}_i^j + (1-\omega) \cdot \check{\theta}_i^j \qquad \omega \in [0,1]$$

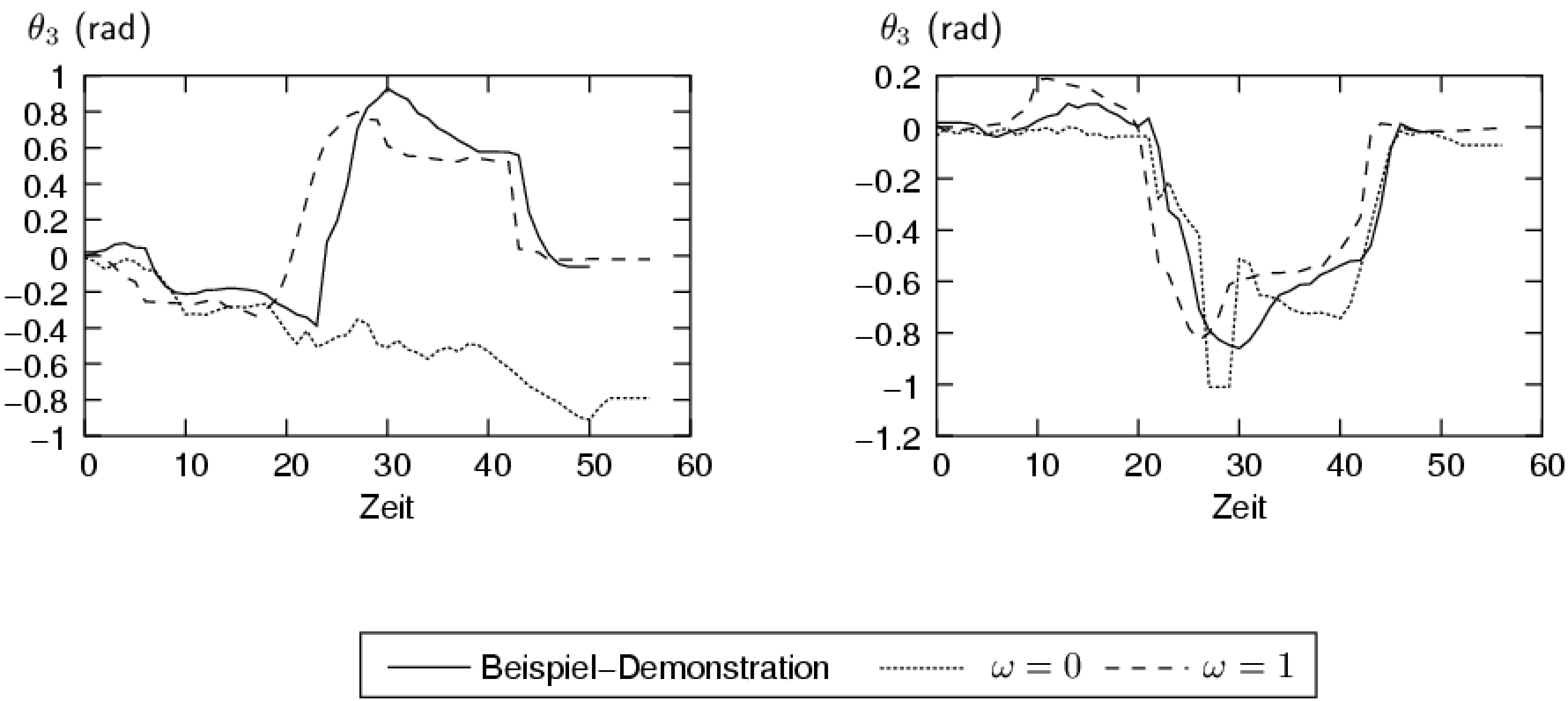
Comparison of the generalized trajectories of the left hand for different weighting factors



Experiment (3)

$$\tilde{\theta}_i^j = \omega \cdot \hat{\theta}_i^j + (1-\omega) \cdot \check{\theta}_i^j \qquad \omega \in [0,1]$$

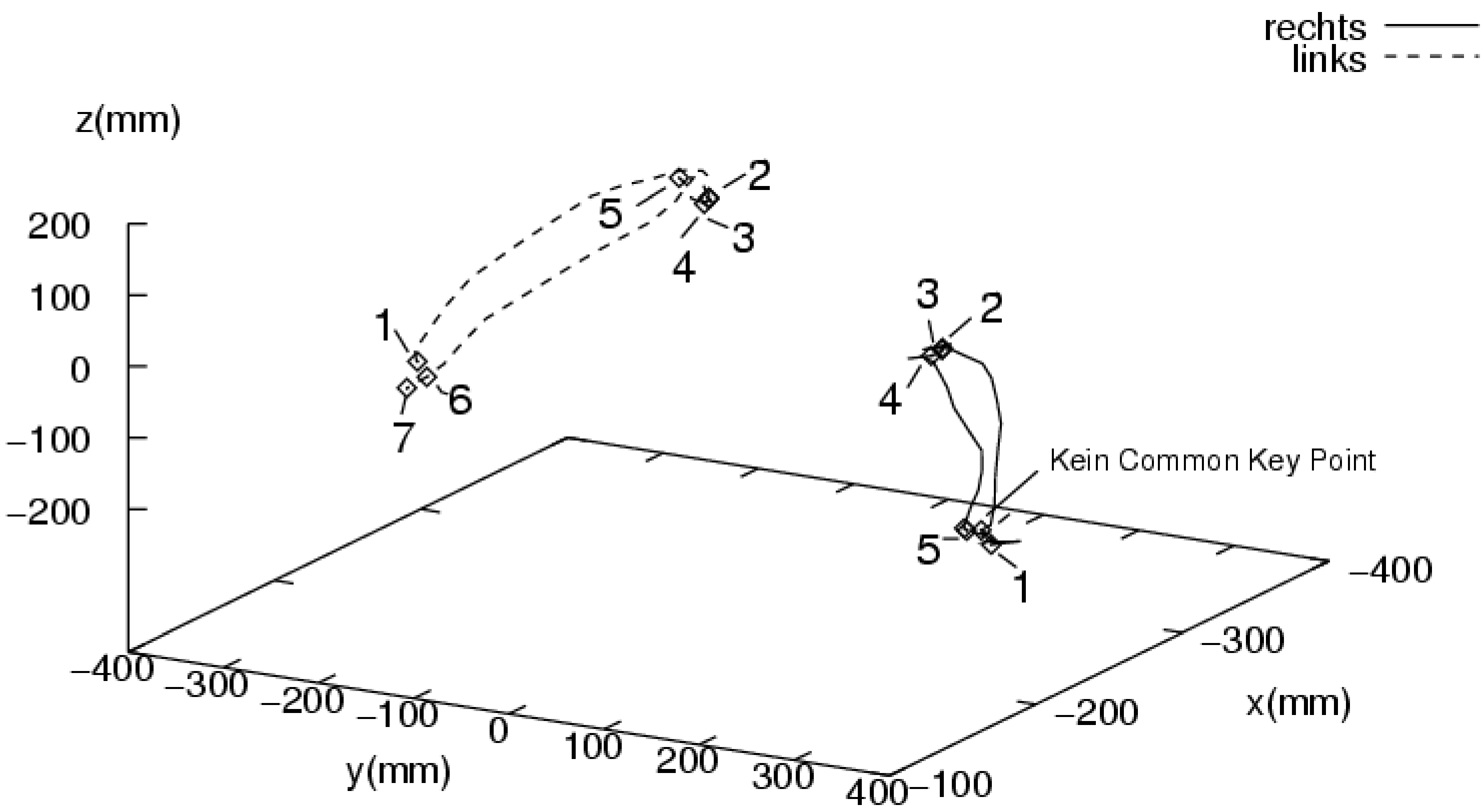
Joint angle trajectories for a selected joint(θ_3)



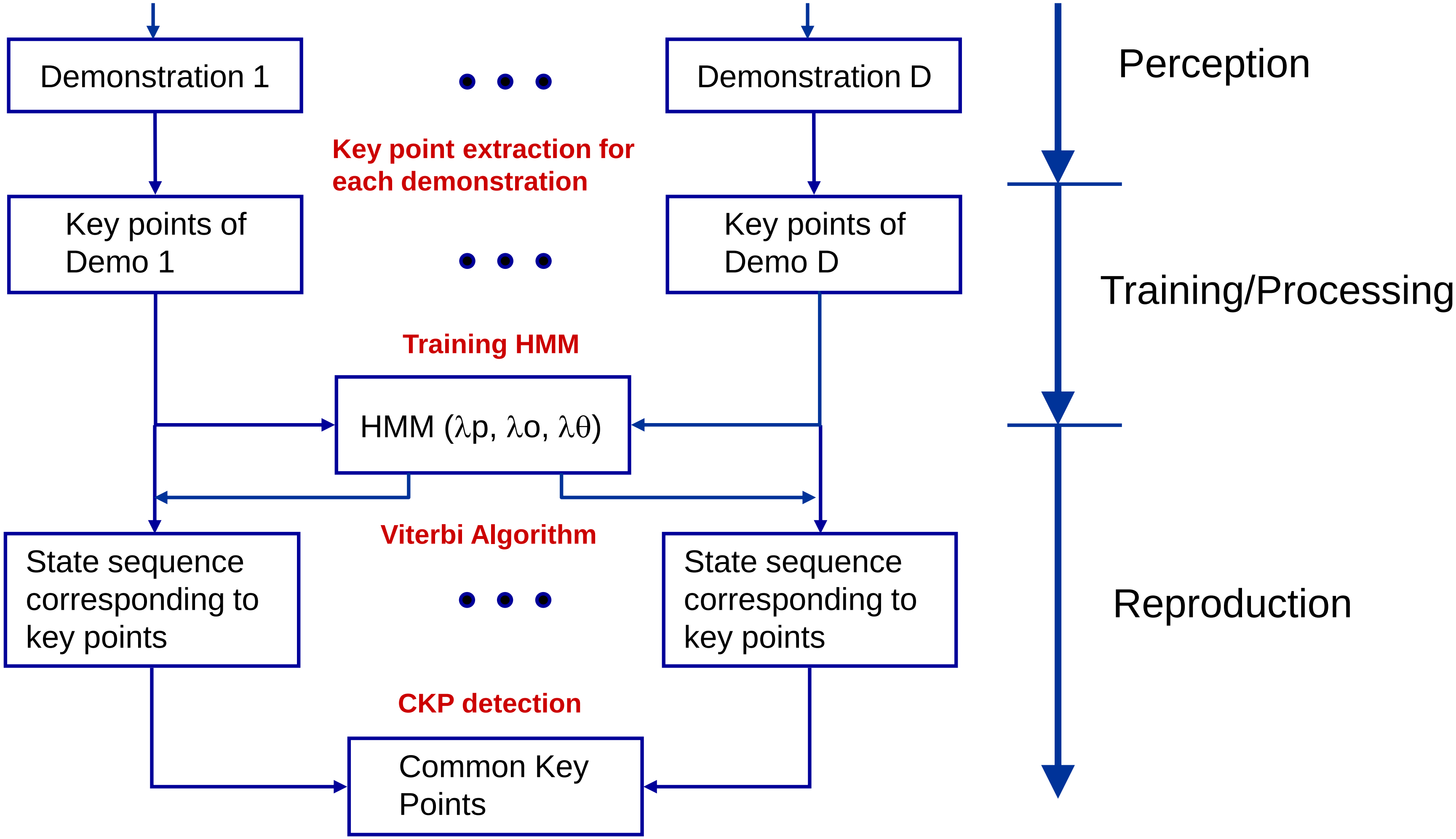
Experiment (4)

$$\tilde{\theta}_i^j = \omega \cdot \hat{\theta}_i^j + (1-\omega)\cdot \check{\theta}_i^j \qquad \omega \in [0,1]$$

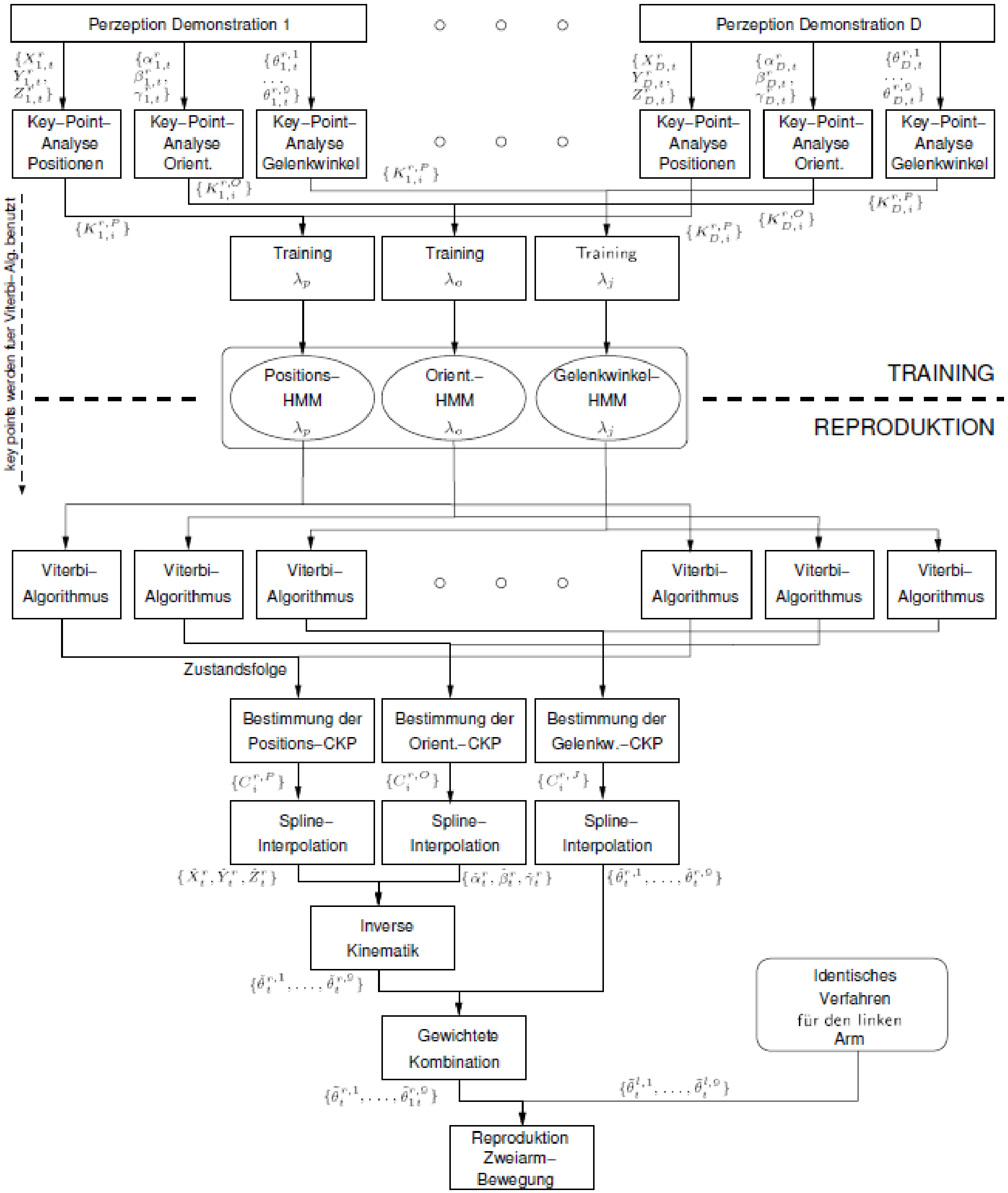
Insertion movement (right hand holds glass, left hand bottle).



Approach: from perception to reproduction



Approach: from perception to reproduction



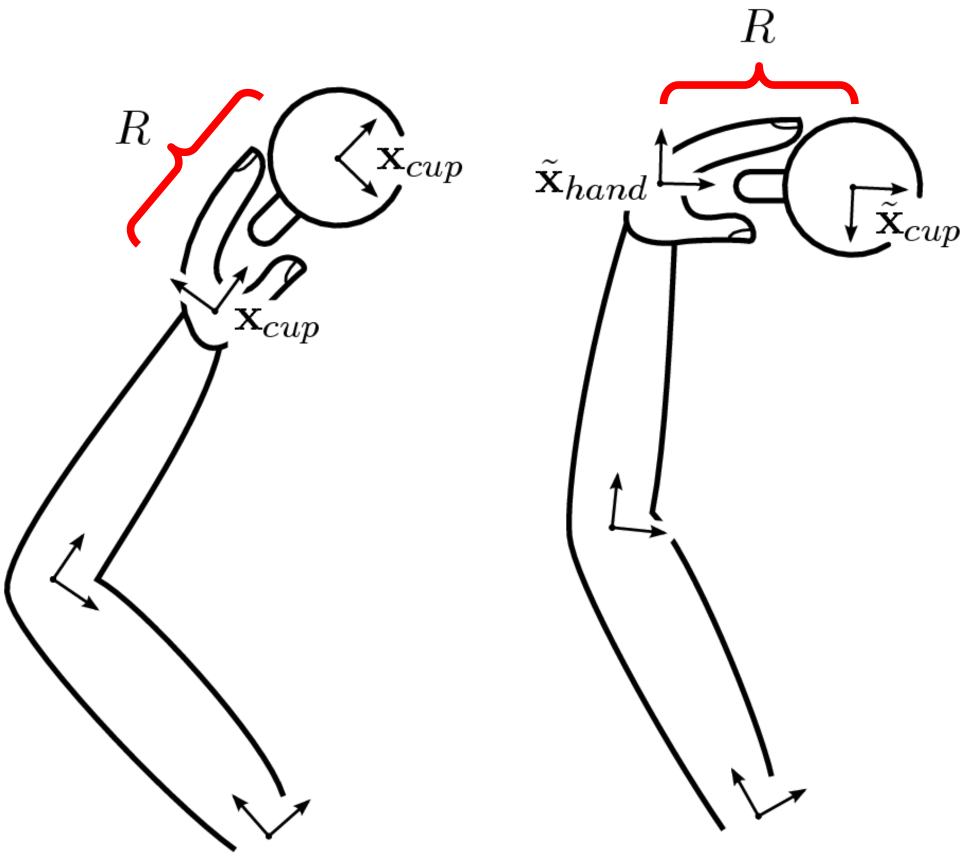
What can / should we learn?

- Classic motor learning: learn the control policies

- At a “more cognitive level”
  - Learn the goal states of an action.
  - Learn about the state of the environment, for example preconditions that need to be fulfilled to execute the action primitives.

This learning could be done for example by gathering the statistics about the state of the environment before and after the action execution.

What can / should we learn?



- Long term goal
  - Not a dancing robot
  - A learning robot
  - Interaction with objects
- Perfect imitation
  - absolutely same relation **R** to an object
  - similar arm posture

What can / should we learn?

- Example of learning control policies
  - Learning desired trajectories with HMMs
    - Representation by a sequence of keypoints.
    - Can be used for action recognition.
    - Can be used for action reproduction.

- Limitations:
  - The current method considers only the movement information
  - Do not directly relate to the goal states of an action.
  - No feedback control possible.

Incorporating the goals in the reproduction

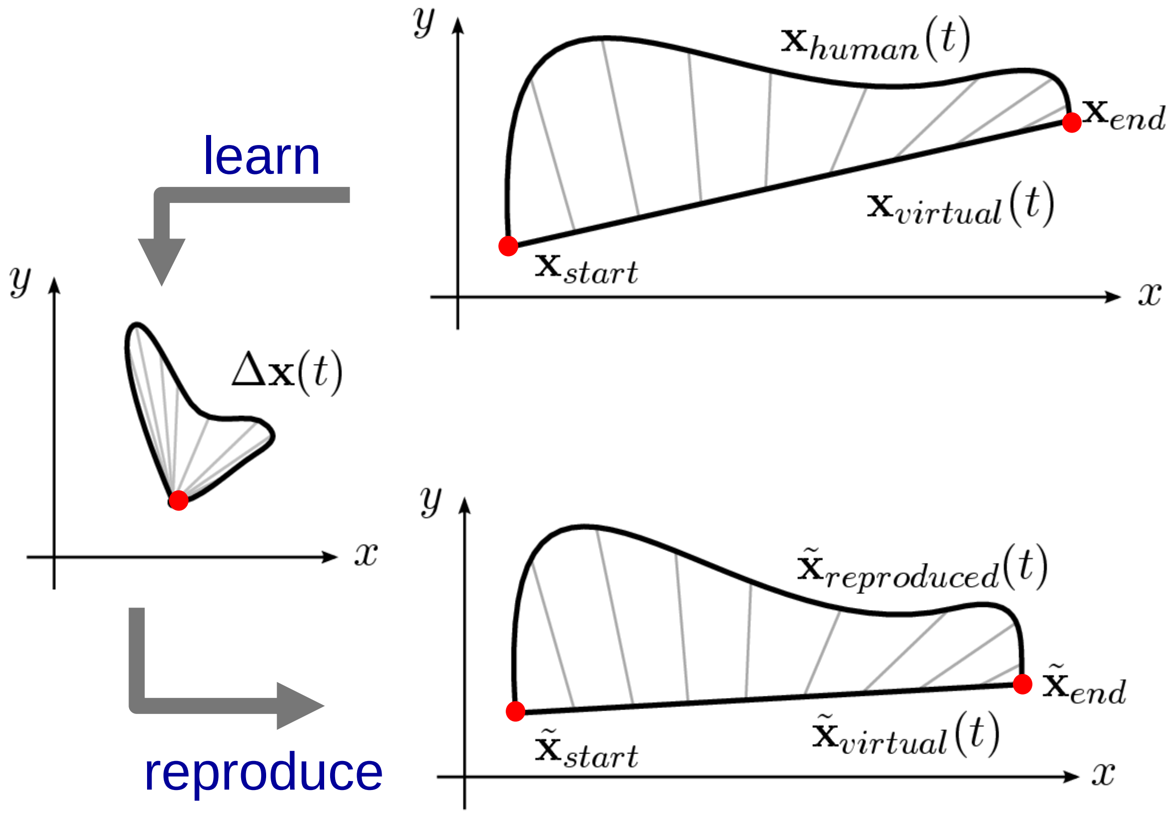
- HMMs are nice because they account both the reproduction and classification, but we need to account also for the goals of an action and other types of control than feed-forward control.

Incorporating goals in the reproduction

- Adaptation of movements to the given situation
- Interpolation between movements in a motion library

Idea: Adaptation of movements to the given situation

- Low-level imitation:
  Parameters only at start and end points
- Learn relative path
- Reproduction with respect to changed world
- Limitation:
  Only small changes can be captured.



Red: relevant for the exam

Task Segmentation Based on Object-Hand Relations

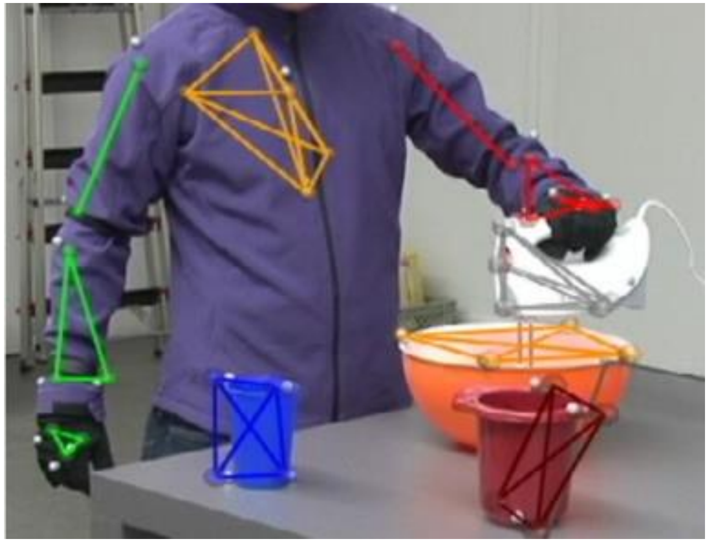
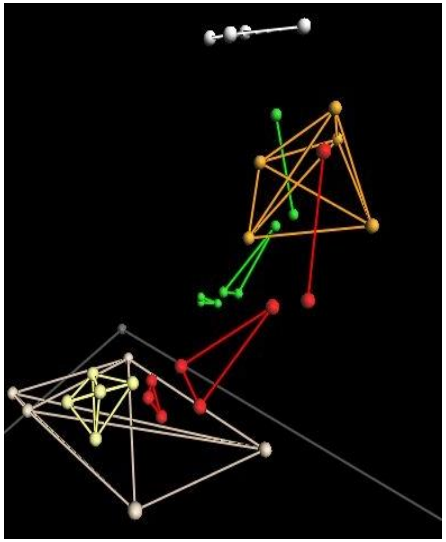
Hierarchical Segmentation

- *M. Wächter and T. Asfour, Hierarchical Segmentation of Manipulation Actions based on Object Relations and Motion Characteristics, International Conference on Advanced Robotics (ICAR), July, 2015*

- *M. Wächter, S. Schulz, T. Asfour, E. Aksoy, F. Wörgötter and R. Dillmann, Action Sequence Reproduction based on Automatic Segmentation and Object-Action Complexes, IEEE/RAS International Conference on Humanoid Robots (Humanoids), October, 2013*

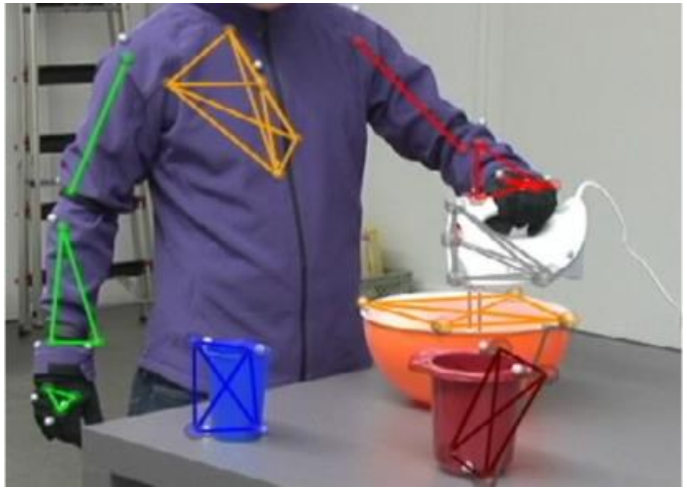
Capturing of the human demonstration

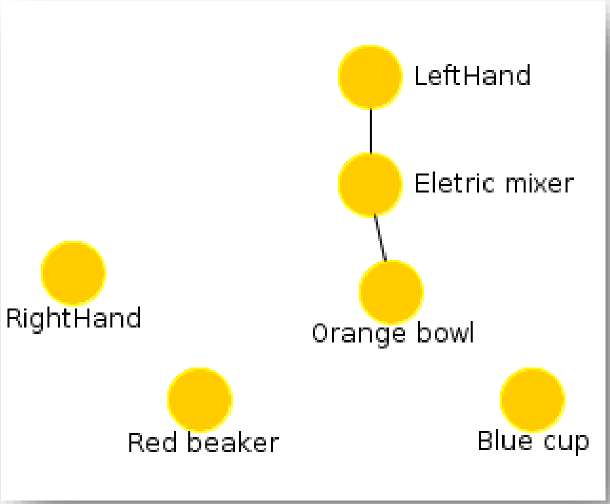
- Human motion capturing with VICON
- The agent and all objects have several markers attached
- All markers are labeled and grouped by the attached object
- Extraction of marker trajectories

Action Segmentation: Idea

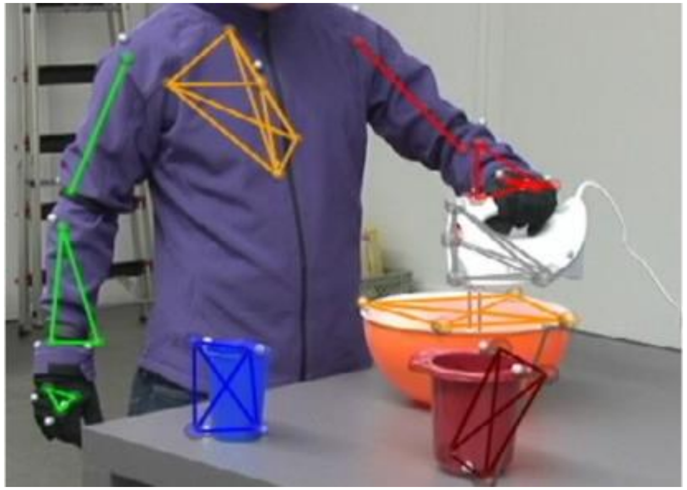
- Use contact/non-contact relations to segment the action, i.e. to extract important key frames out of object interactions
- Cartesian distance of markers employed to detect contact and key frames
- Resulting in segmentation of an action sequence into several action primitives
  - Example: „Preparing a dough" is divided into grasping, pouring, placing etc.
- But these action primitives are still unlabeled

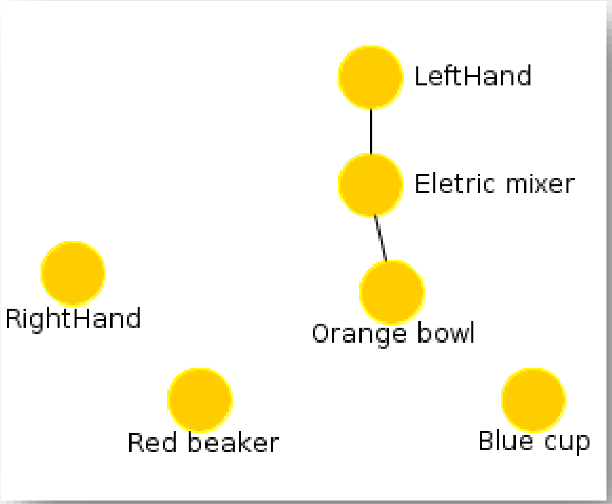




Action Segmentation: Idea

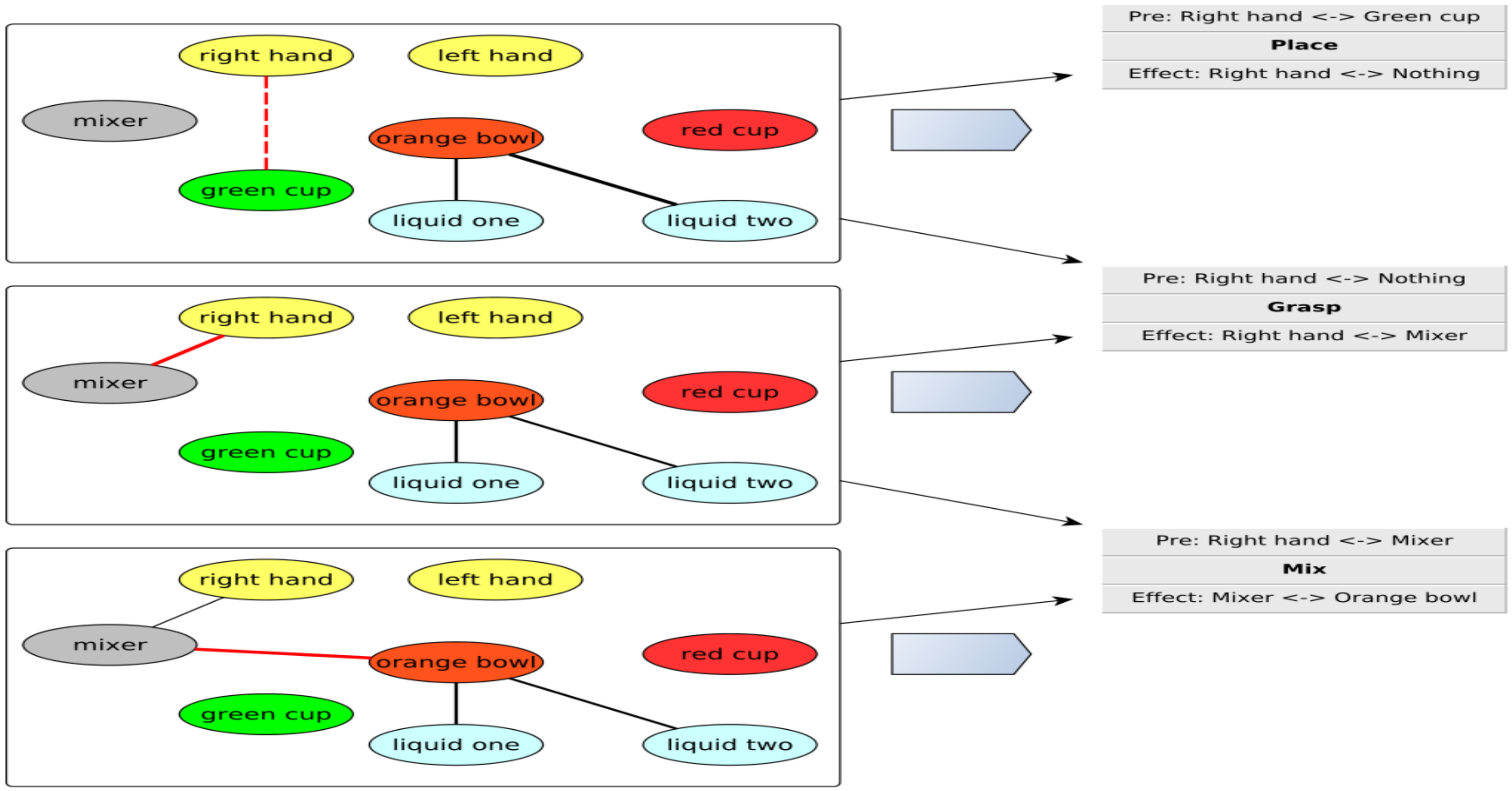
- Object relations support the extraction of general pre/postconditions of an action primitive
  - Grasping postcondition: *LeftHand touches RedCup*
  - ➔ *RedCup* is grasped





Action Segmentation

- World state and pre/postconditions



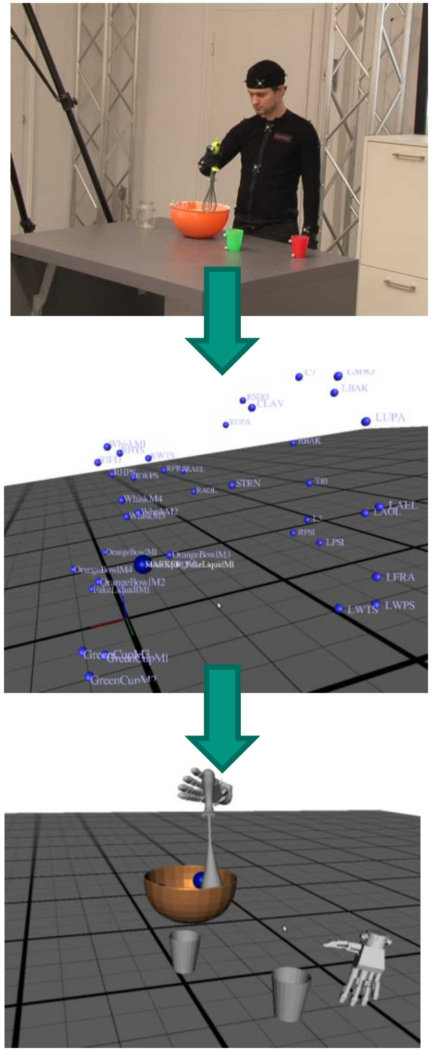
The whole process

Detection contact between hand and object

- Using vision: difficult
- Using haptics: require contact sensors on the human hand

Instead:

- Mapping of marker representation into geometric model representation
  - 3D mesh model for all objects with virtual markers
  - Registration of recorded markers with virtual markers for 6D pose estimation
  - 6D trajectories analyzed by mesh collision detection algorithms

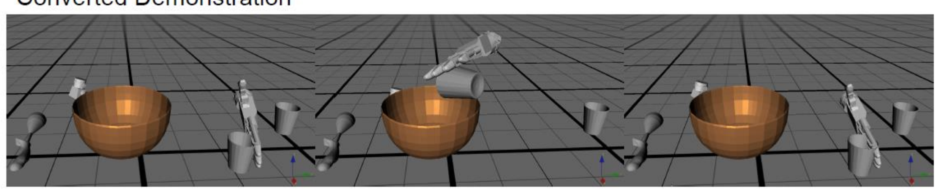


Hierarchical Action Segmentation

- Extension of previous semantic segmentation (Wächter et al., 2013)
- Semantic segmentation provides relevant information about key frames but actions without observable effects cannot be detected
- Segmentation of human demonstration on two levels
  - Semantic segmentation based on object relation changes
  - Motion segmentation based on trajectory characteristics

Human Demonstration

Converted Demonstration



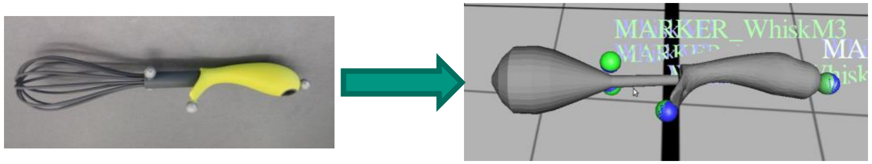
Hierarchical Segmentation

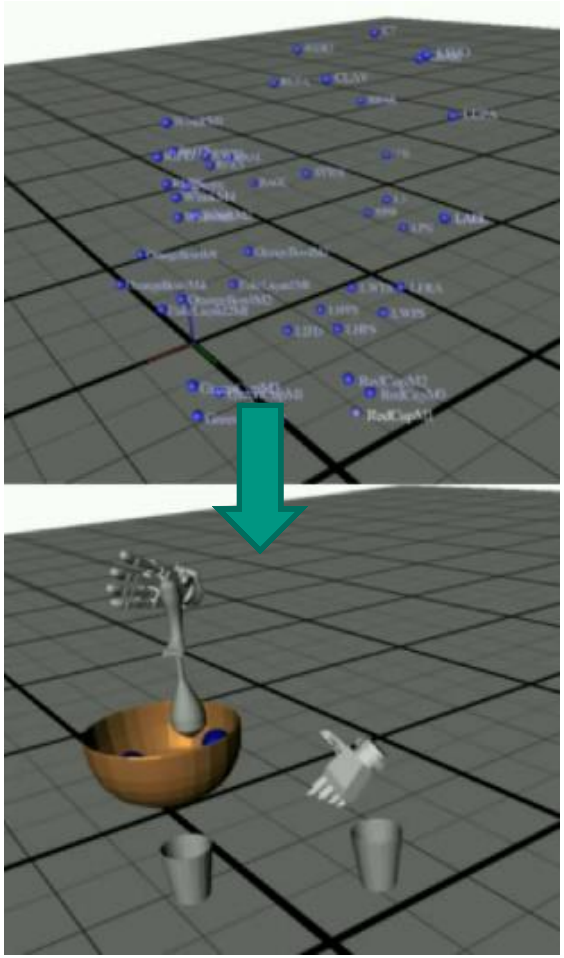
No contact	Cup in left hand			No contact
Grasp	Lift	Pour	Place	Retreat

Wächter and Asfour, 2015

Marker-based Motion Capture to 6D Object Trajectories

- Recordings of action sequences
  with marker-based motion capture
- Conversion to 6D object pose trajectories
  with simplified MMM models
- Input for segmentation algorithm



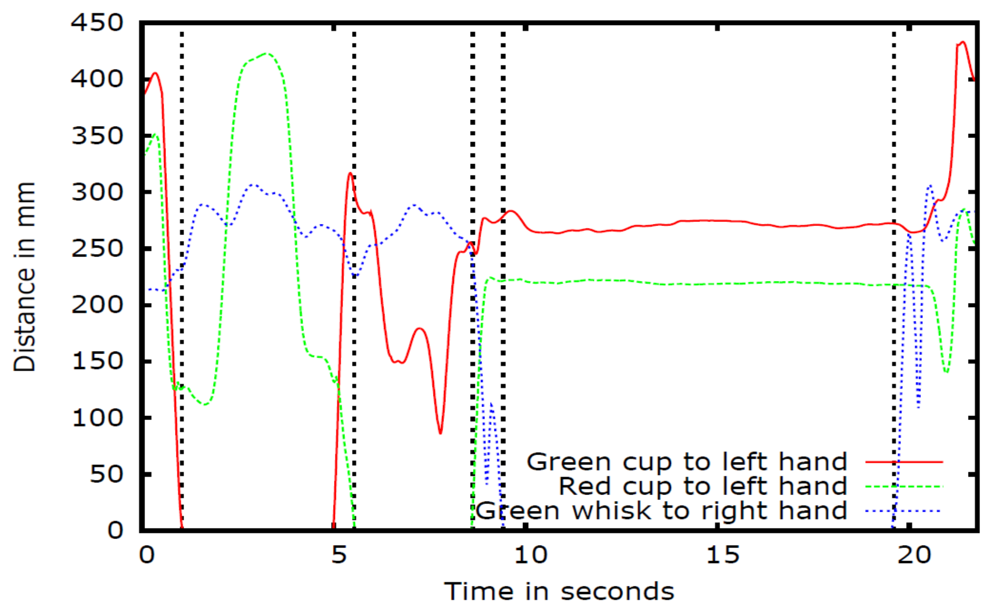


Top level: Semantic Segmentation

Hierarchical Segmentation

Object Relations	No contact	Cup in left hand			No contact
Motion trajectories	Grasp	Lift	Pour	Place	Retreat

- Extraction of contact relation changes between hand-objects and object-object using 3D mesh models and Simox[1] collision detection algorithms



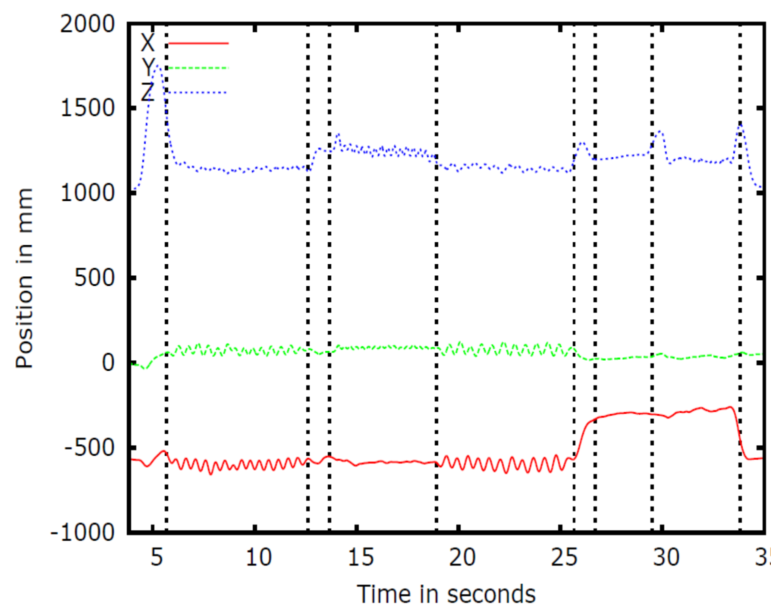
[1]: simox.sourceforge.net

Bottom Level: Motion segmentation based on trajectory characteristics

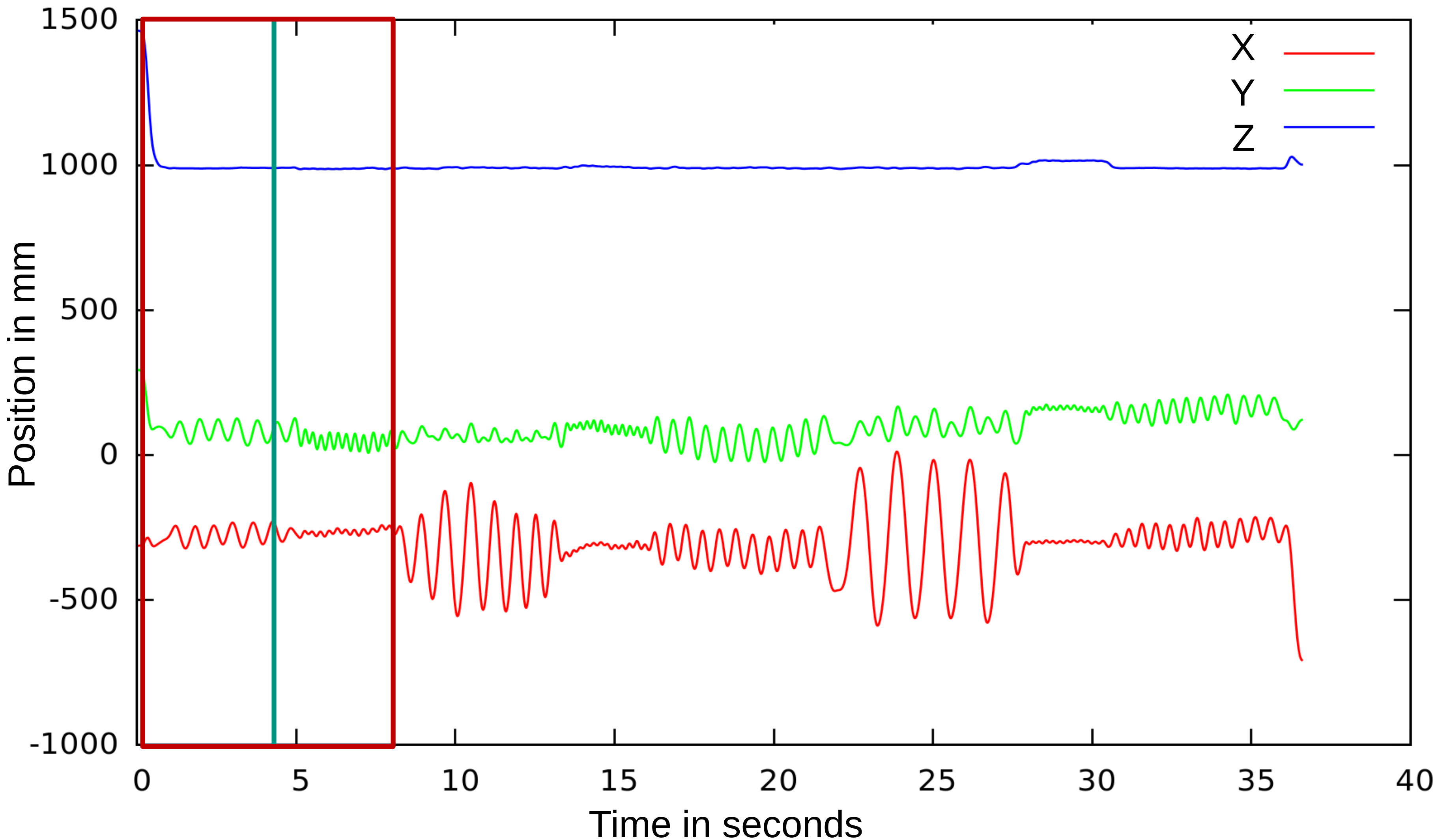
Hierarchical Segmentation

Object Relations	No contact	Cup in left hand			No contact
Motion trajectories	Grasp	Lift	Pour	Place	Retreat

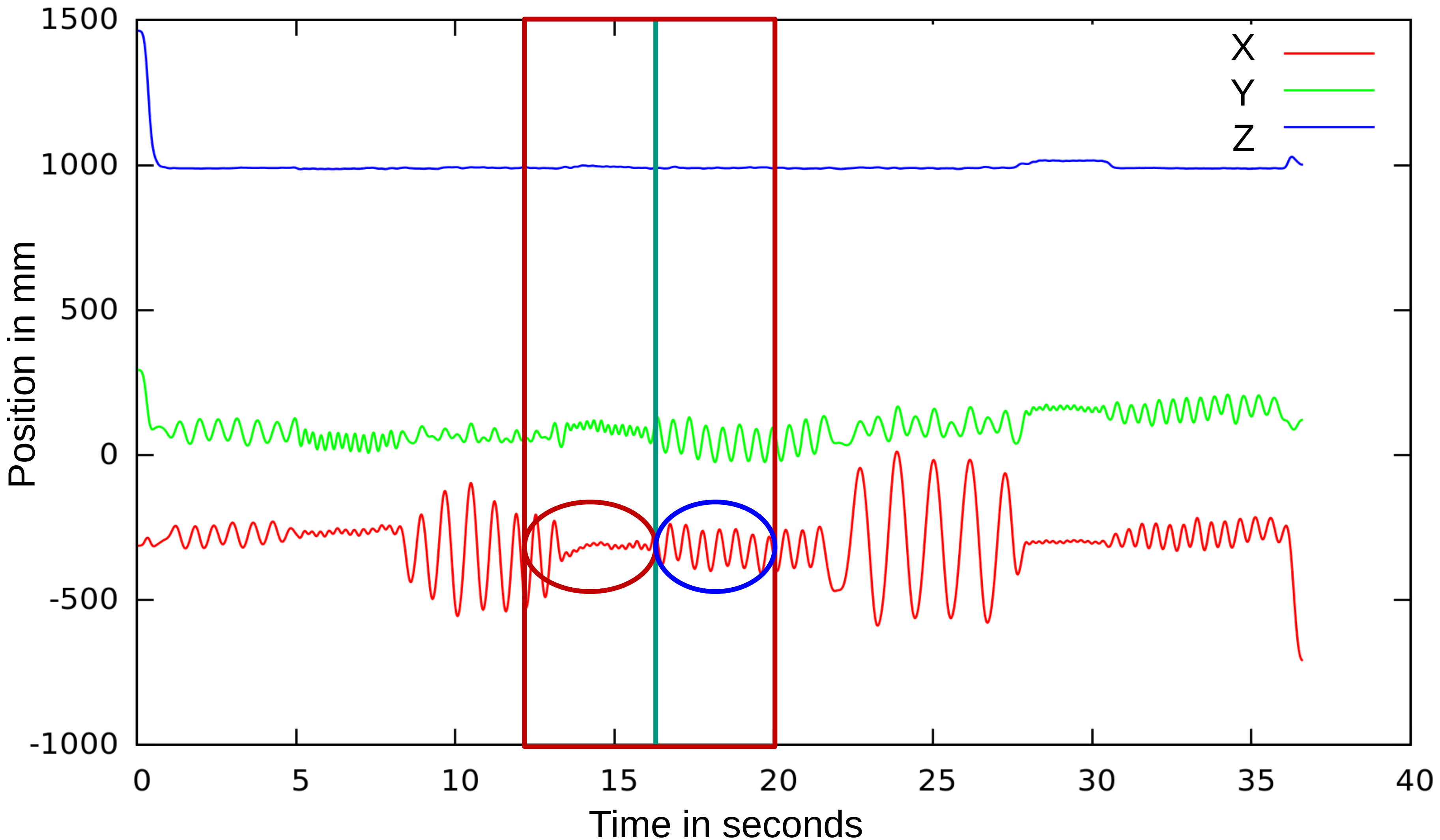
- Segmenting of semantic segments into **most distinctive parts** based on the motion characteristic
  - New **heuristic** based on **acceleration profile** to segment motion
  - Iterative search for best key frame in current semantic segment
  - Recursive segmentation until segments **too small** or **too similar**



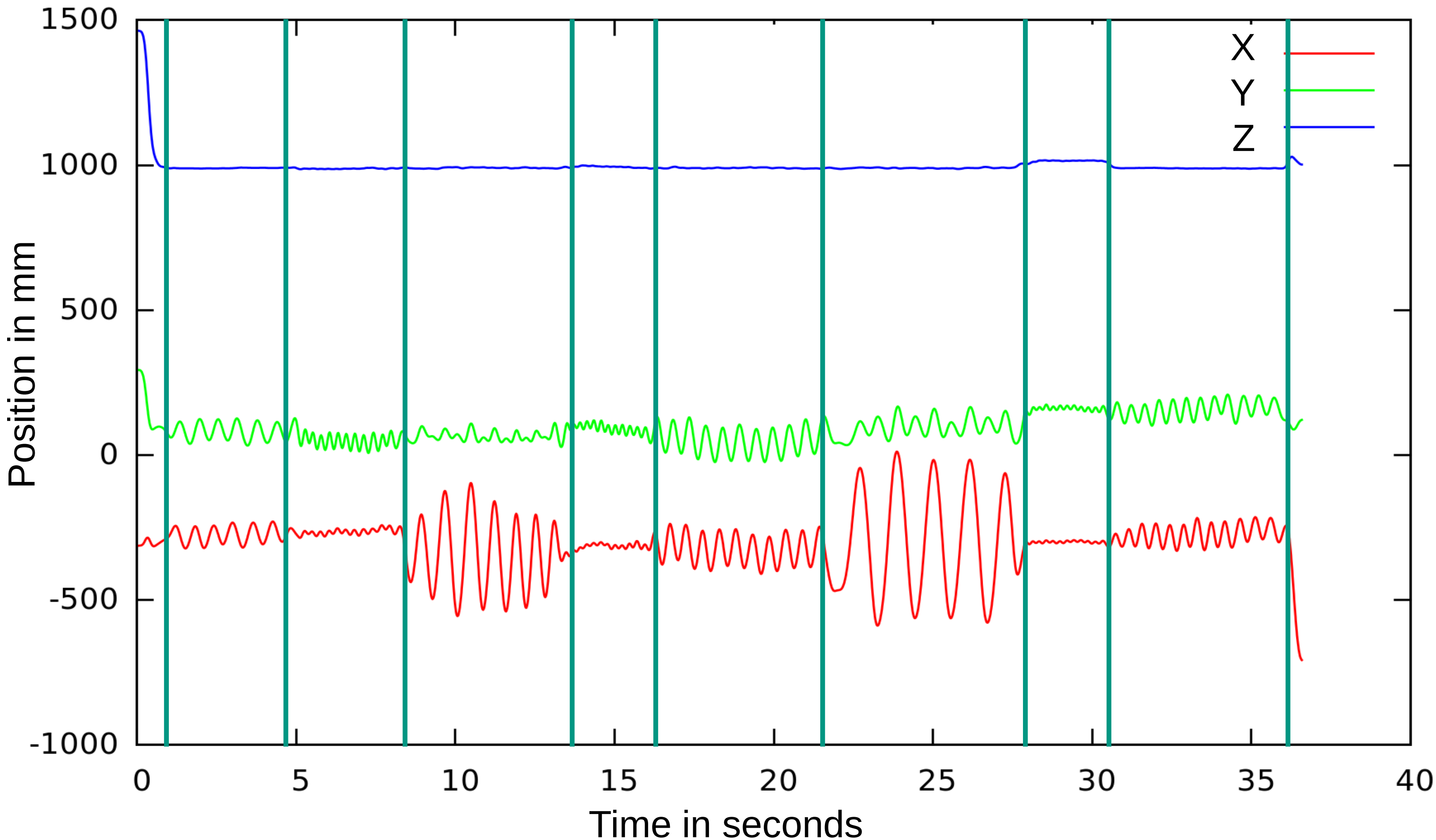
Motion Characteristics Heuristic



Motion Characteristics Heuristic



Motion Characteristics Heuristic



Motion Characteristics Heuristic

- Heuristic based on the acceleration profile
  - Measuring the length of the acceleration curve
  - $A_d(t) = |\, a_d(t+1) - a_d(t)|$

$$s_{l,d}(t_c) = \sum_{t=t_c-\frac{w}{2}}^{t_c-1} A_d(t)\left(\frac{\hat{U}_l}{\hat{U}_r}\right)^2$$

$$s_{r,d}(t_c) = \sum_{t=t_c}^{t_c+\frac{w}{2}-1} A_d(t)\left(\frac{\hat{U}_r}{\hat{U}_l}\right)^2$$

- Iterative search for best candidate key frame with a sliding window
  - Comparison of segments left and right of candidate with score function *s*
  - Key frame quality: $q_d = \begin{cases} \frac{s_{l,d}}{s_{r,d}} & s_{l,d} > s_{r,d} \\ \frac{s_{r,d}}{s_{l,d}} & s_{l,d} \leq s_{r,d} \end{cases}$
- Recursive subdividing until segment size or quality too small

Evaluation of segmentation results

- New metric for assessment of segmentation results
  - Mean squared error with penalties for missing and additional key frames

$$e = (m + f) * p + \sum_i \min_j (k_{r,i} - k_{f,j})^2$$

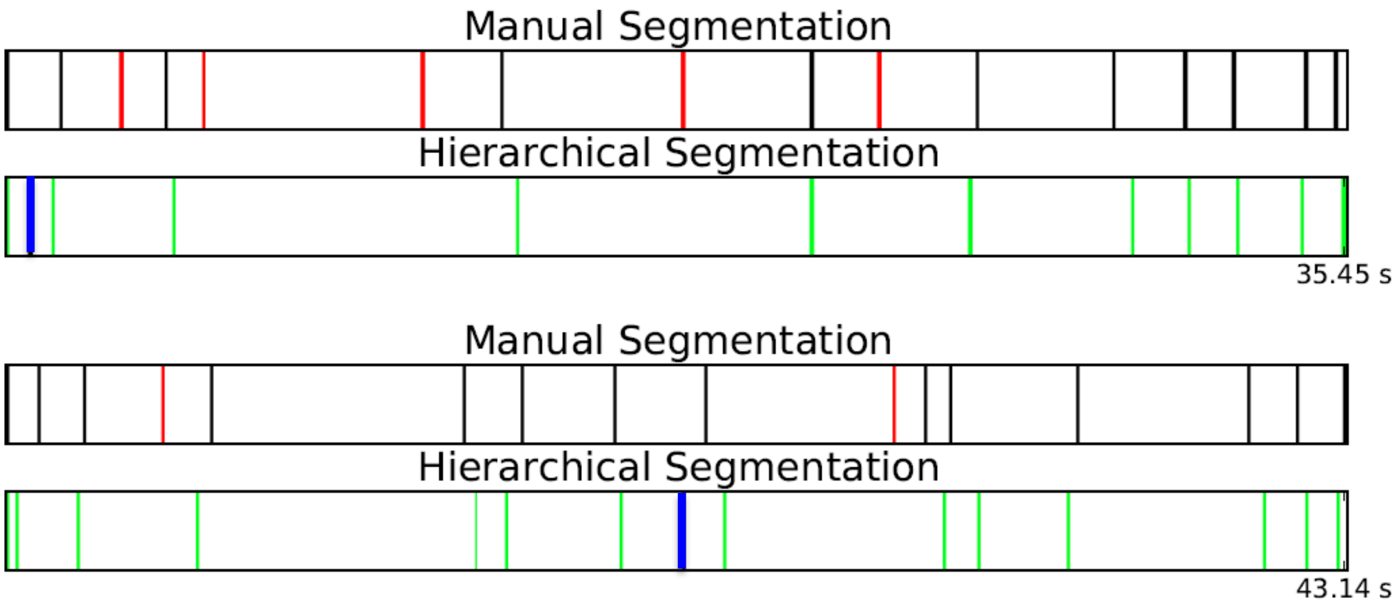
m: missed key frames
f: false positives
p: penalty

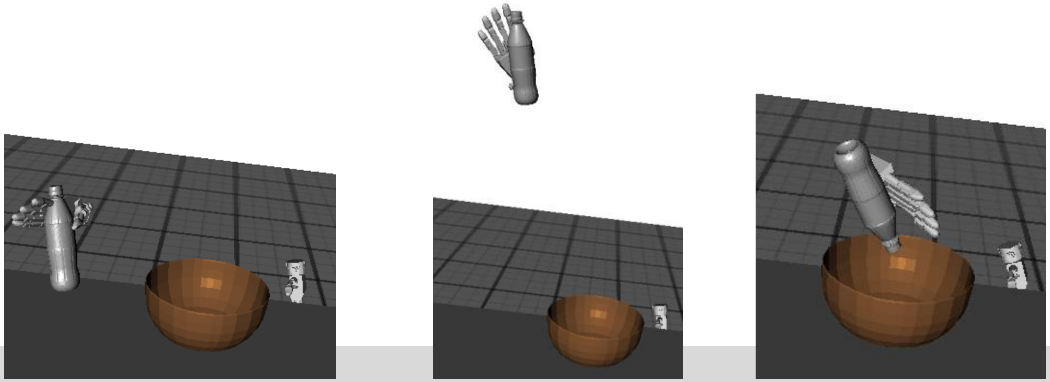
- Comparison to manual reference segmentation
  - HS: Hierarchical Segmentation
  - ZVC: Zero Velocity Crossing
  - PCA: Principal Component Analysis
- 13 action sequences à ~30 seconds:

Average Results	HS	ZVC	PCA
Error	3.35 s^2	7.01 s^2	20.18 s^2
Accuracy	0.27 s	0.1 s	0.36 s
Unmatched key frames	2	0.6	27.9
Missed key frames	3.54	12.27	3.5

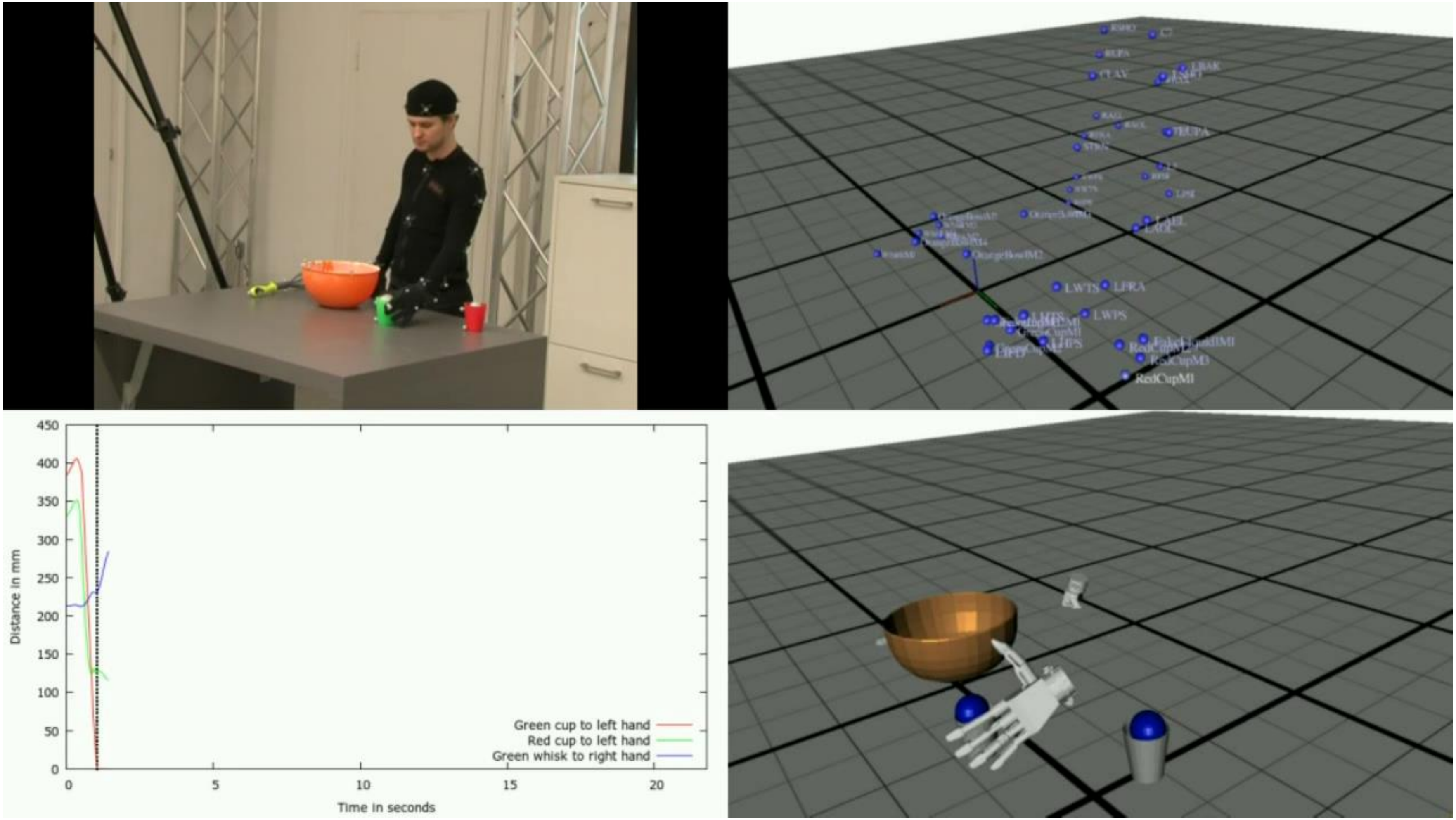
Evaluation: Comparison to reference segmentation

Two repetitions of action sequences of pouring-shaking actions

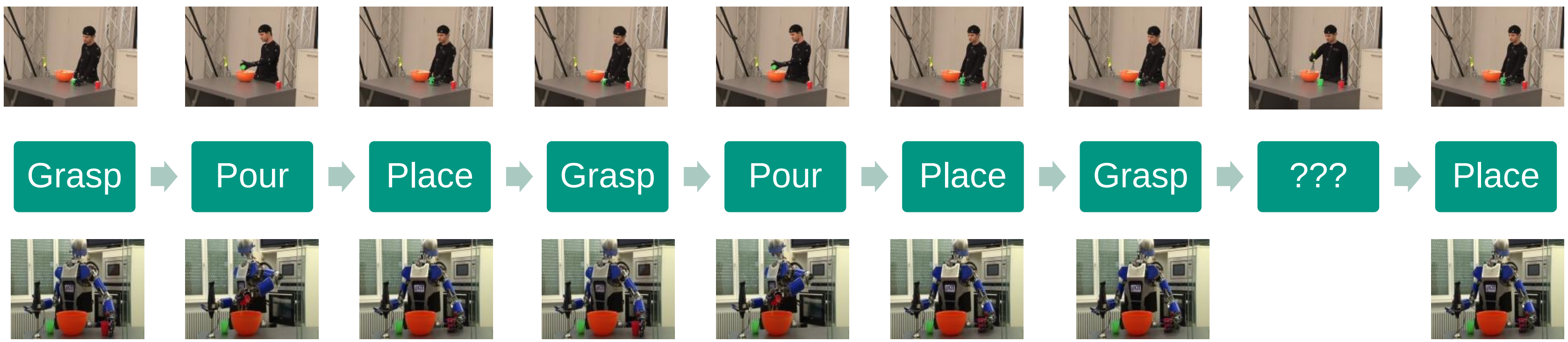


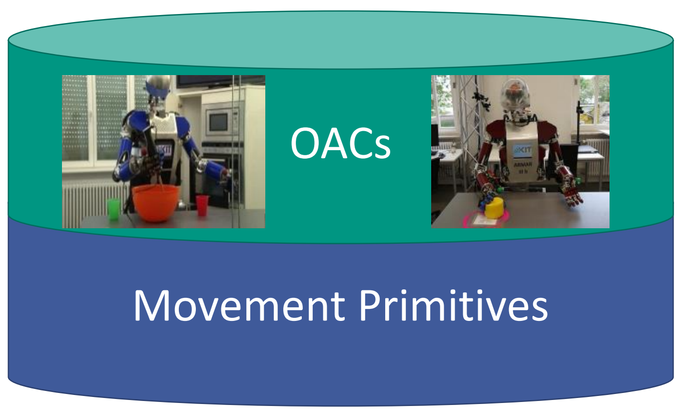


Hierarchical Action Segmentation

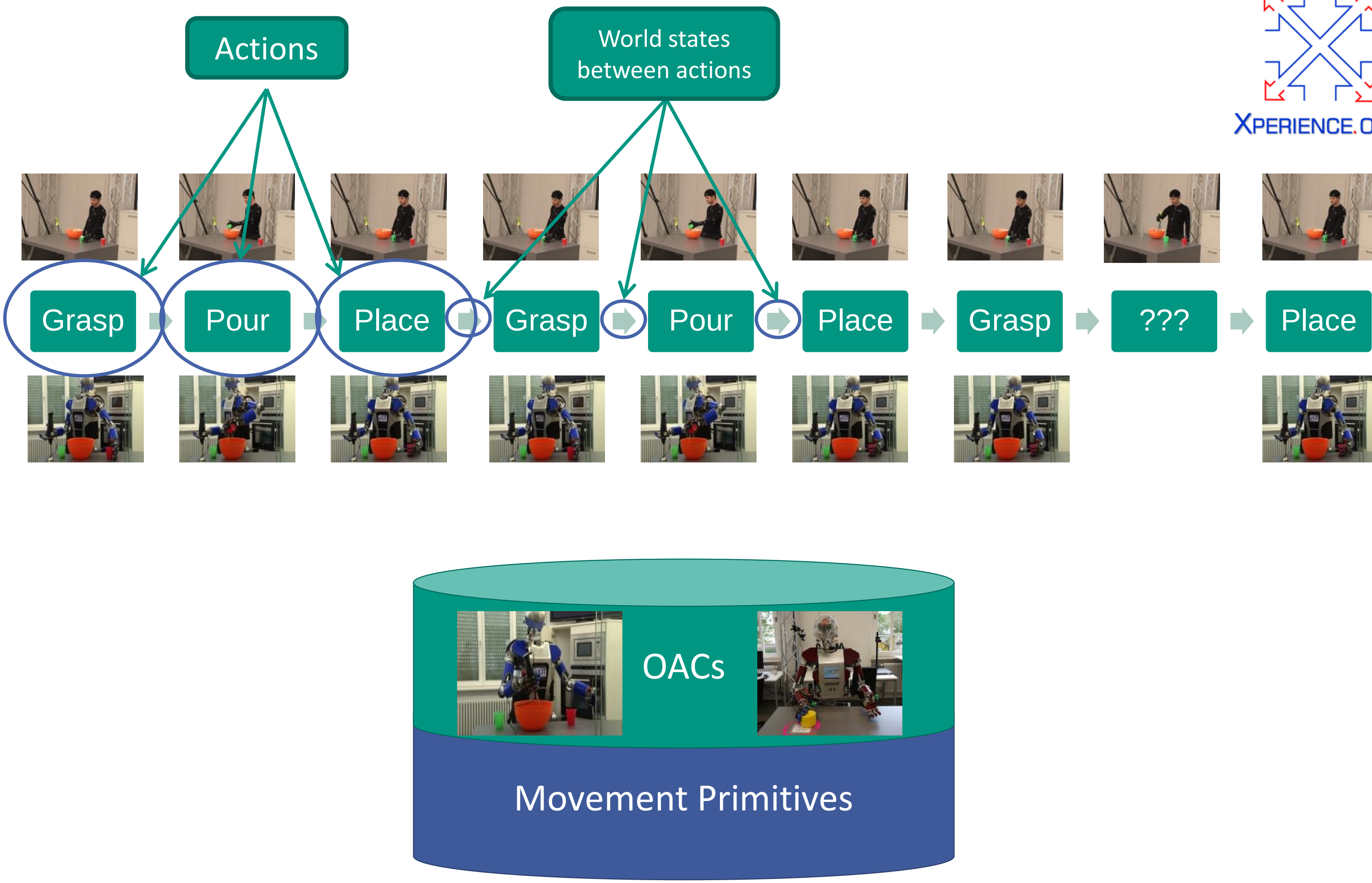


Understanding human demonstration

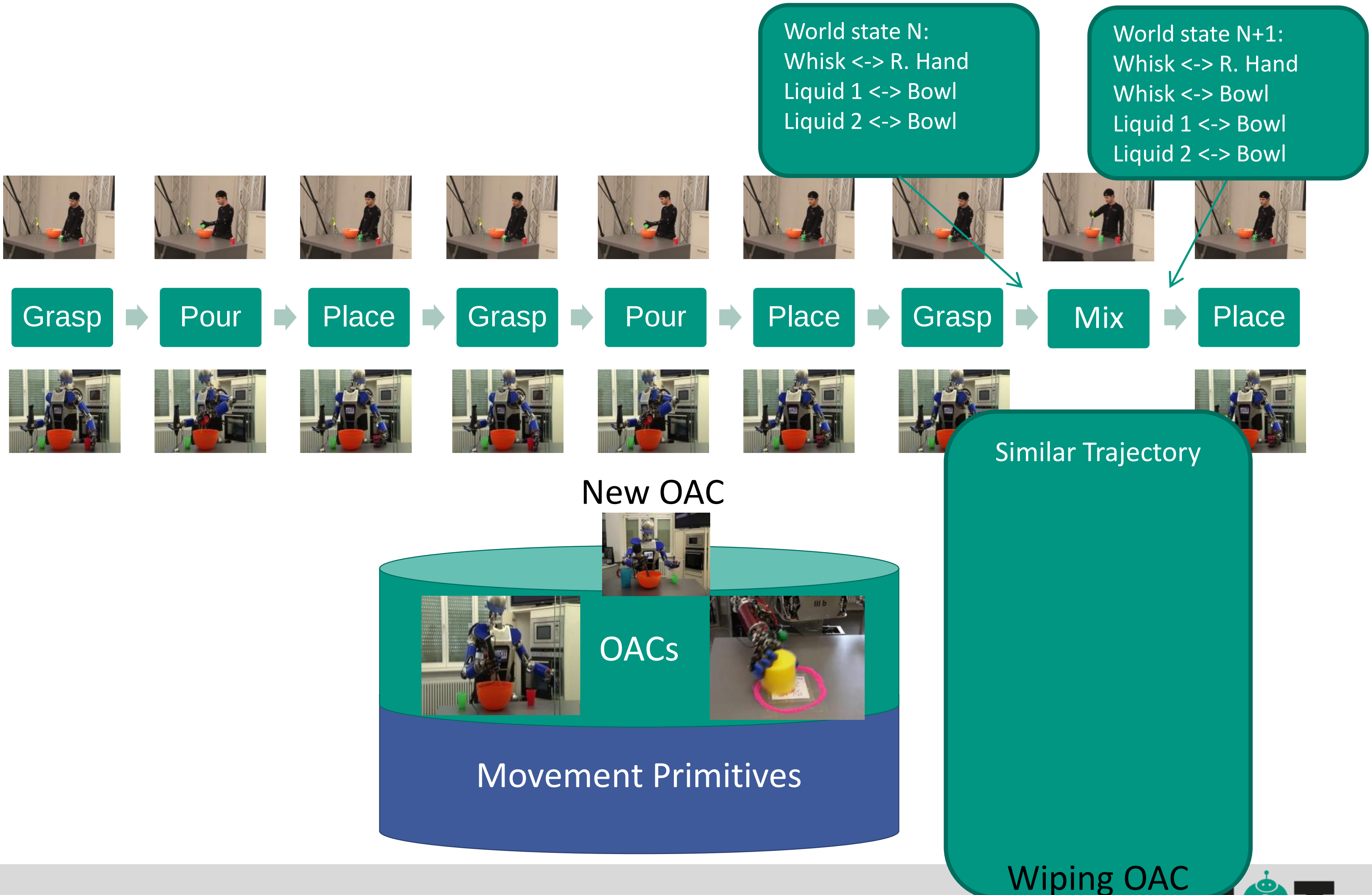




Understanding actions and their effects

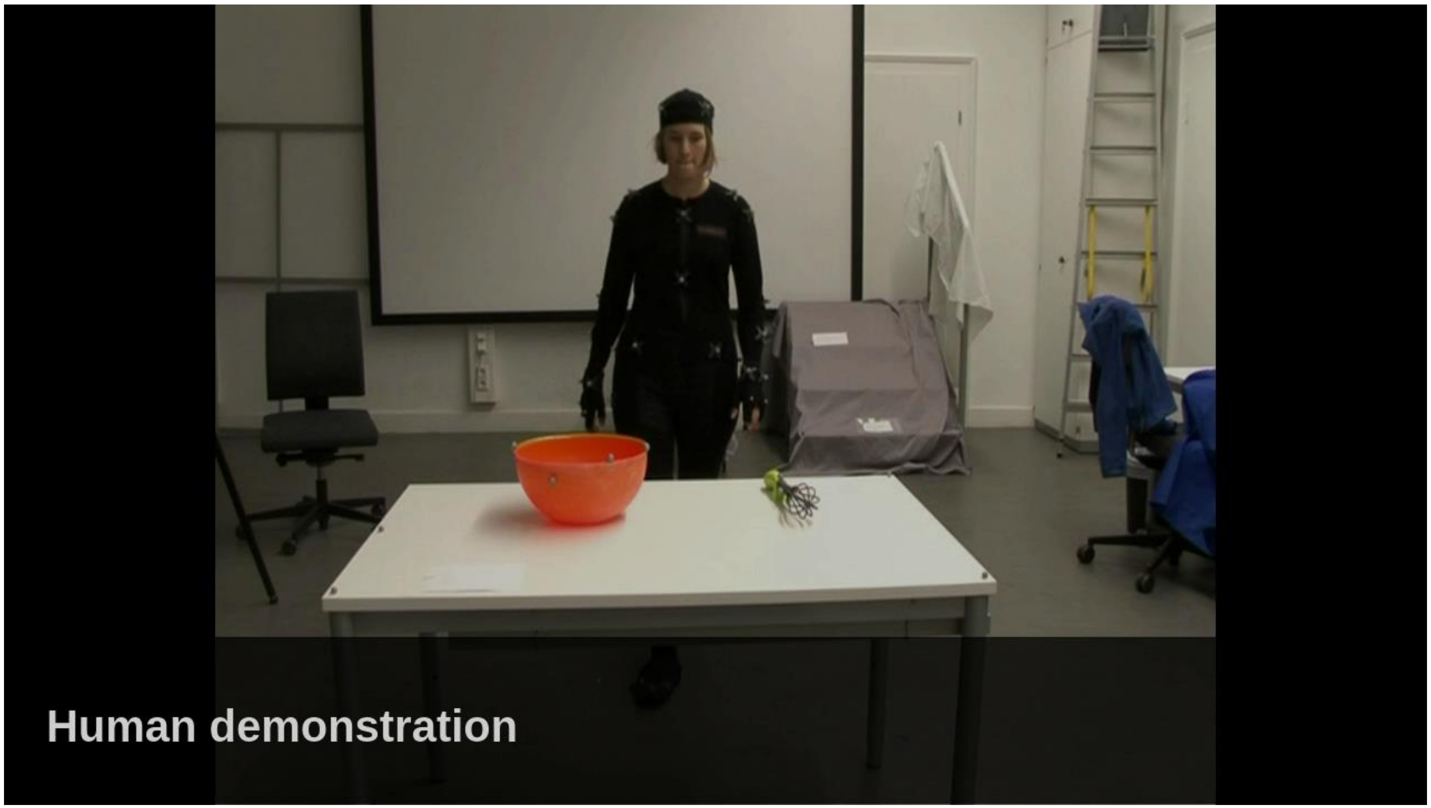


Action replacement from a motion library



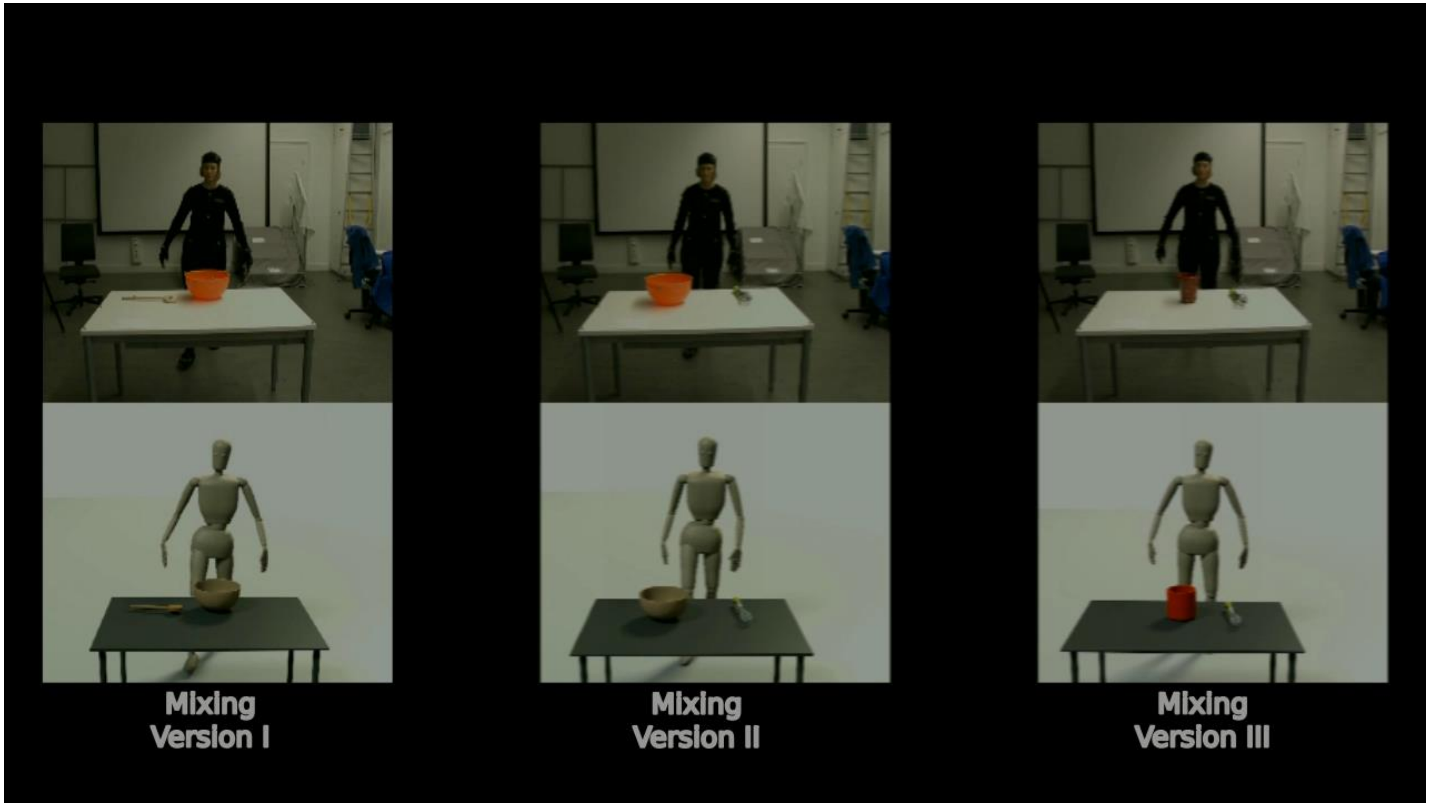
Learning from observation – prepare the dough

KIT Manipulation Action Dataset



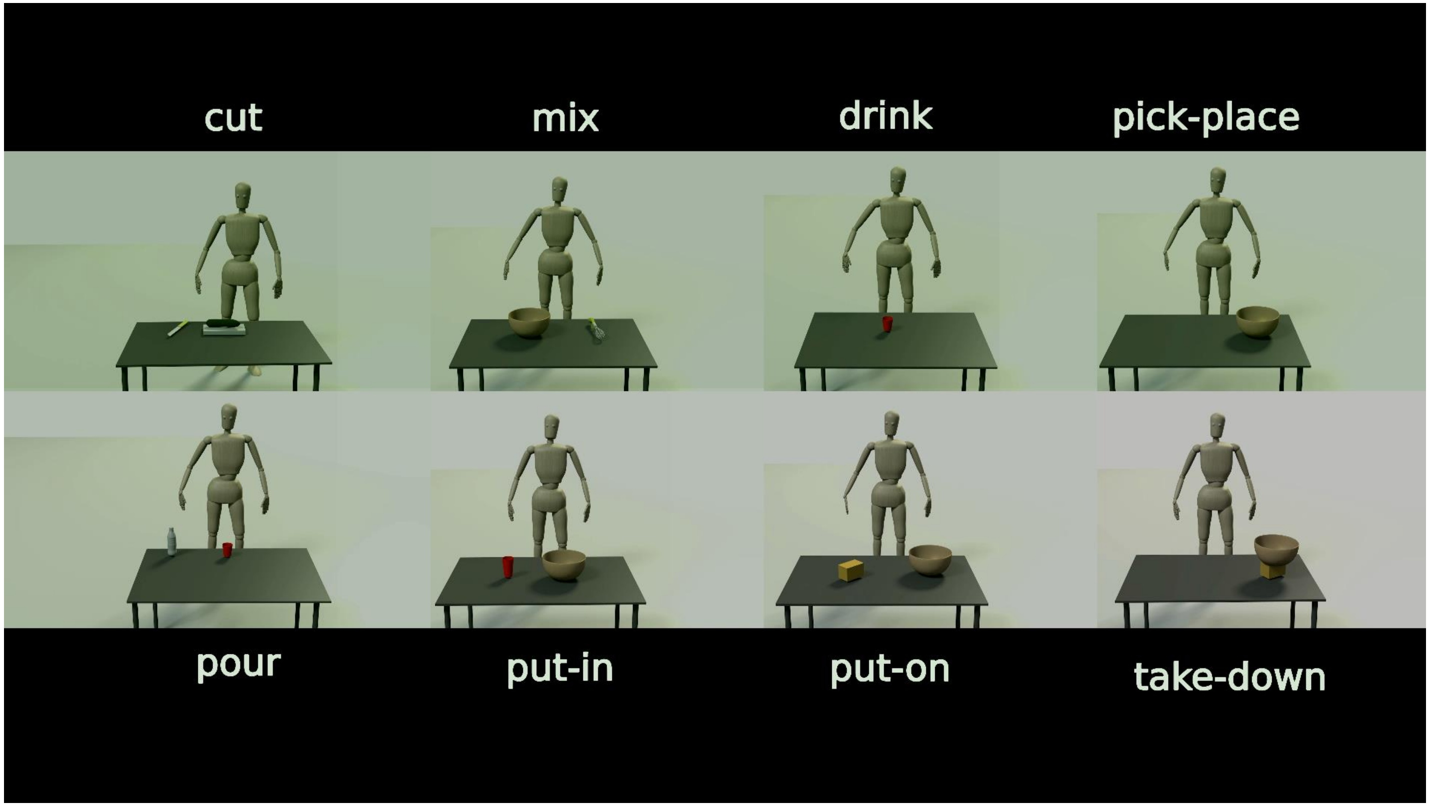
Human demonstration of ***mixing*** action (recoded @100Hz)

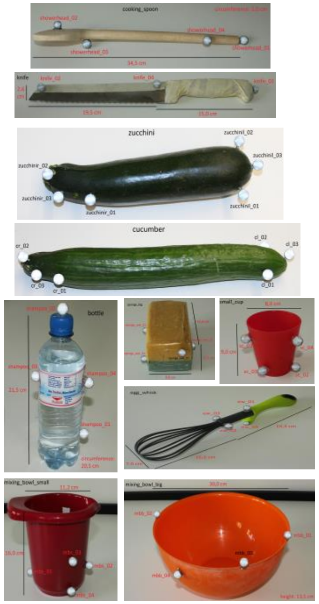
KIT Manipulation Action Dataset



Different demonstrations of ***mixing*** action (recoded @100Hz)

KIT Manipulation Action Dataset



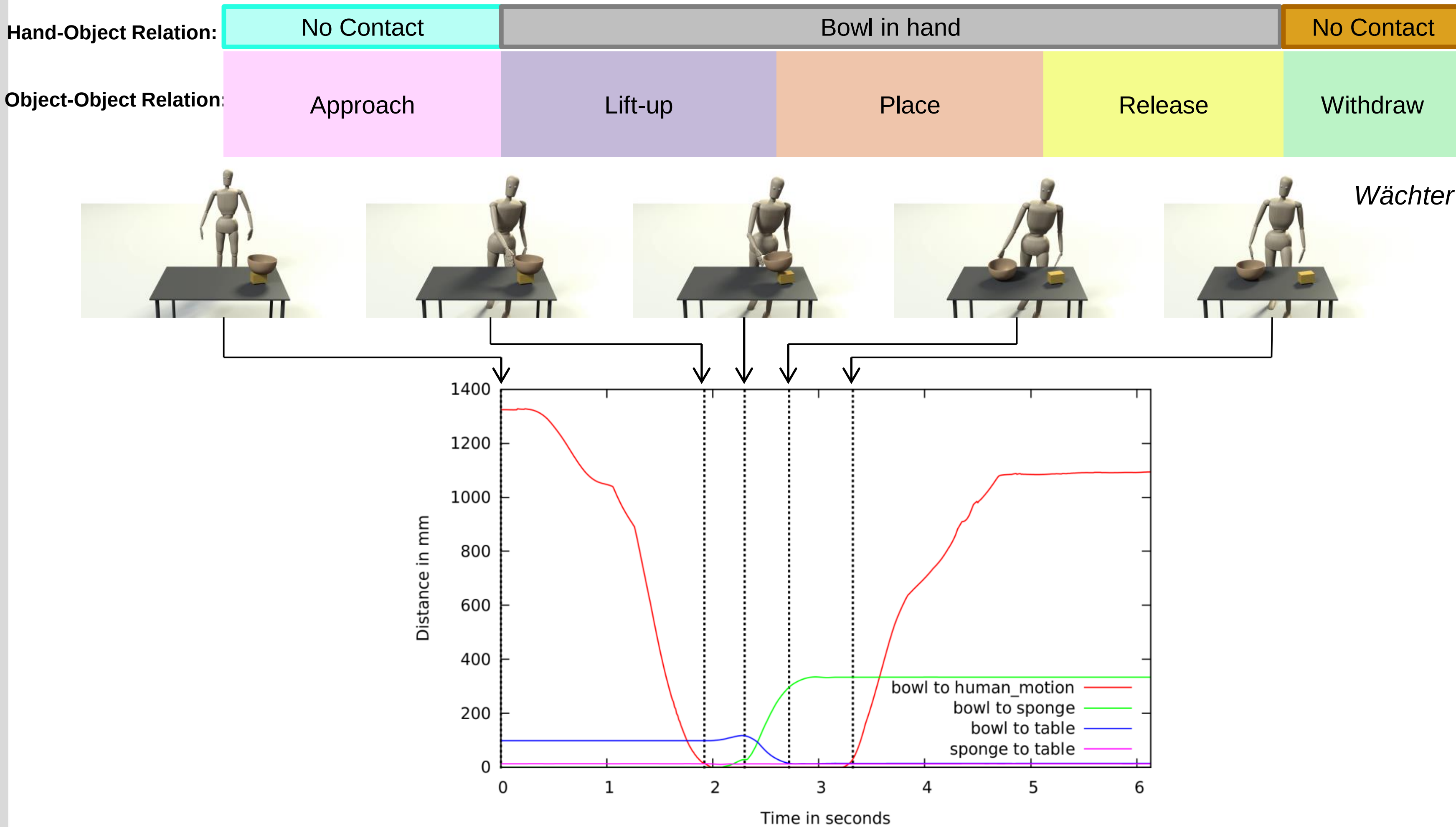


10 Different Objects

In total 70 demonstrations of 8 different manipulation actions

Level I : Semantic Segmentation

Hierarchical Segmentation

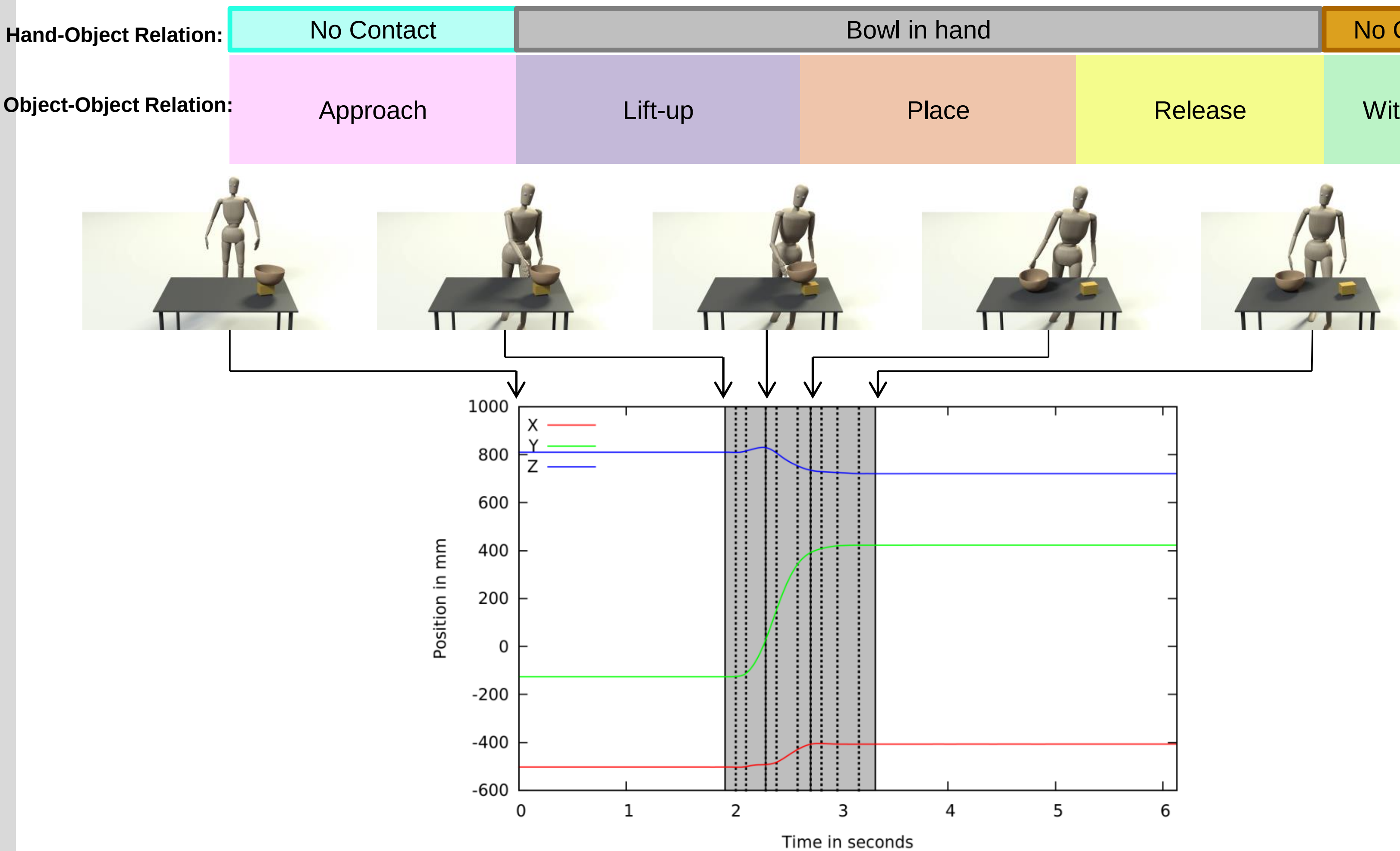


Wächter et al. 2015

Semantic Distance Profile

Level II : Motion Segmentation

Hierarchical Segmentation

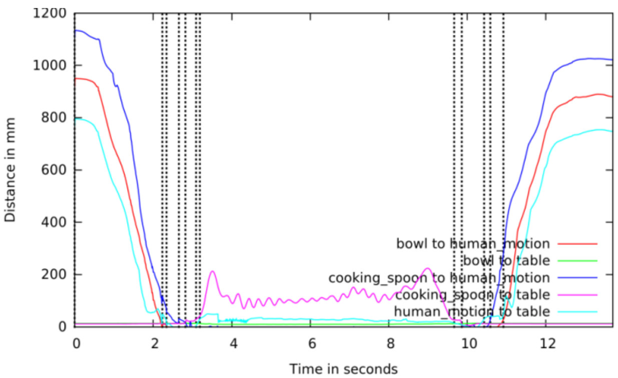


Wächter et al. 2015 ICAR

Trajectory of the Bowl

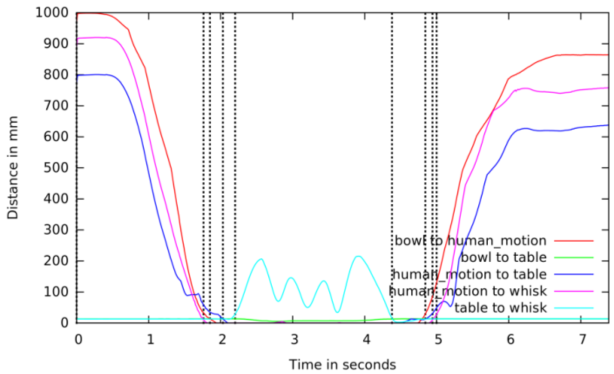
Semantic Action Similarity

Human Demonstration:
Mixing with a *spoon*



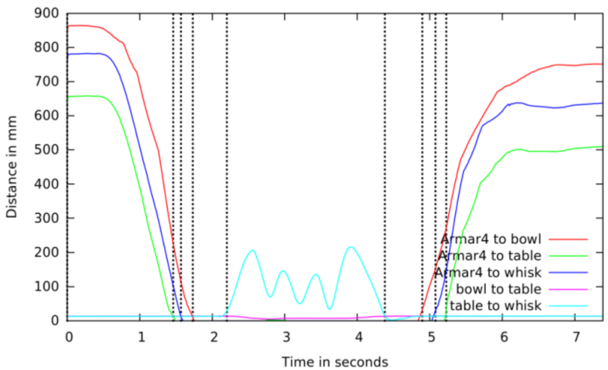
Result of hierarchical segmentation

Human Demonstration:
Mixing with a *whisk*



Result of hierarchical segmentation

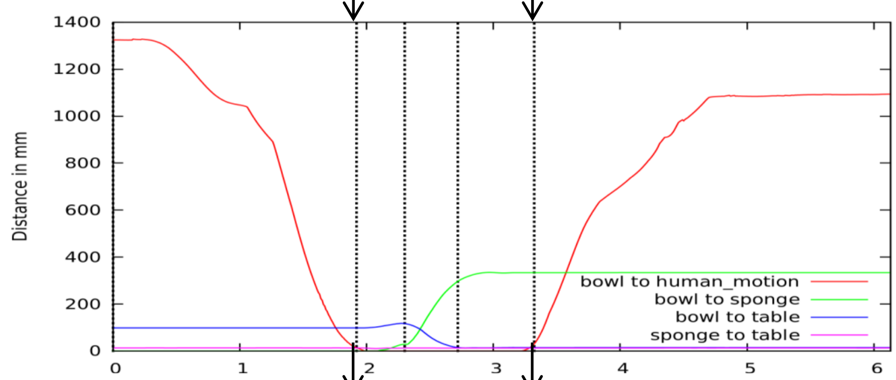
ARMAR-4 Imitation:
Mixing with a *whisk*



Result of hierarchical segmentation

Perception of Time: Put-on Action

No Contact	Bowl in hand	No Contact



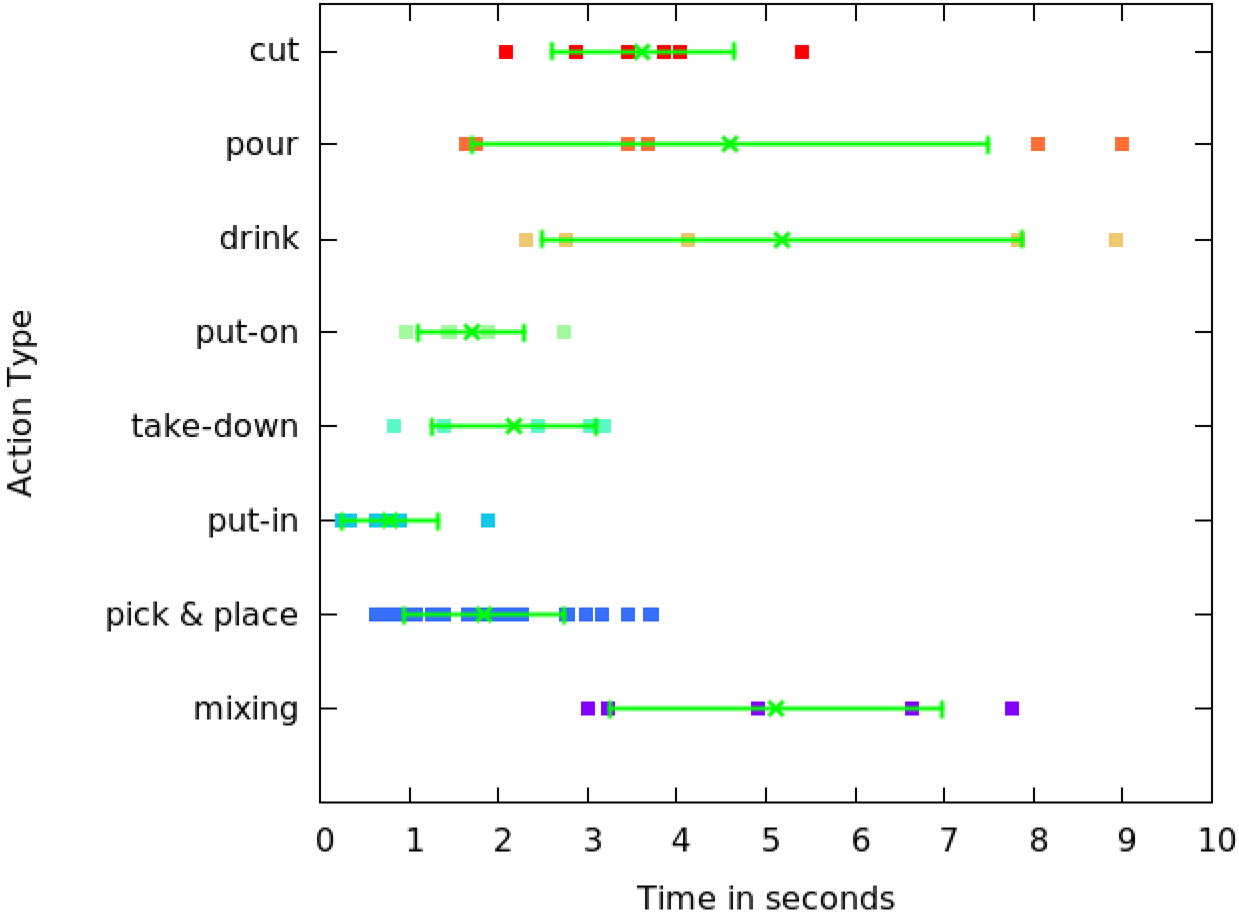
Put-on Action

							Δt
Version I:	1.9	0.4	0.4	0.6	2.8	⇨	1.4 sec
Version II:	2.1	1.0	0.6	0.9	3.2	⇨	2.5 sec
Version III:	2.8	0.9	1.2	1.0	3.0	⇨	3.1 sec
Version IV:	1.7	0.4	0.4	0.1	2.1	⇨	0.9 sec
Version V:	1.8	0.7	1.2	1.1	3.1	⇨	3.0 sec

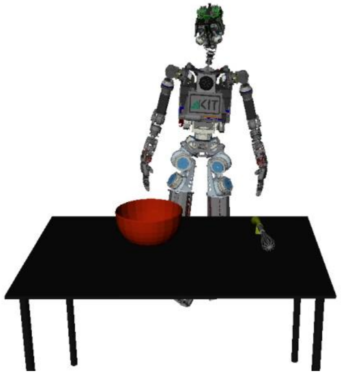
Δt

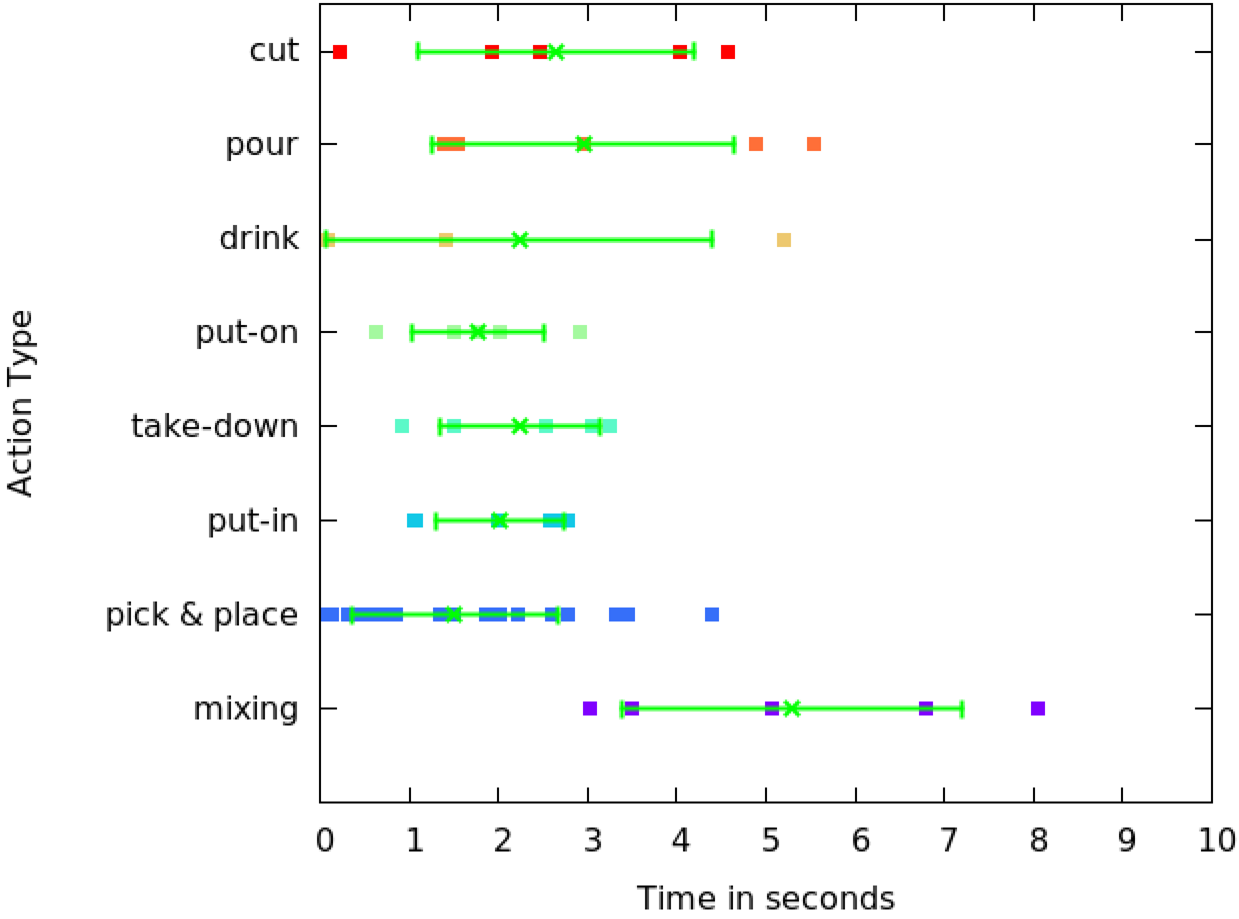
Mean : 2.1 sec Std: 0.9

Perception of Time: Human Demonstration

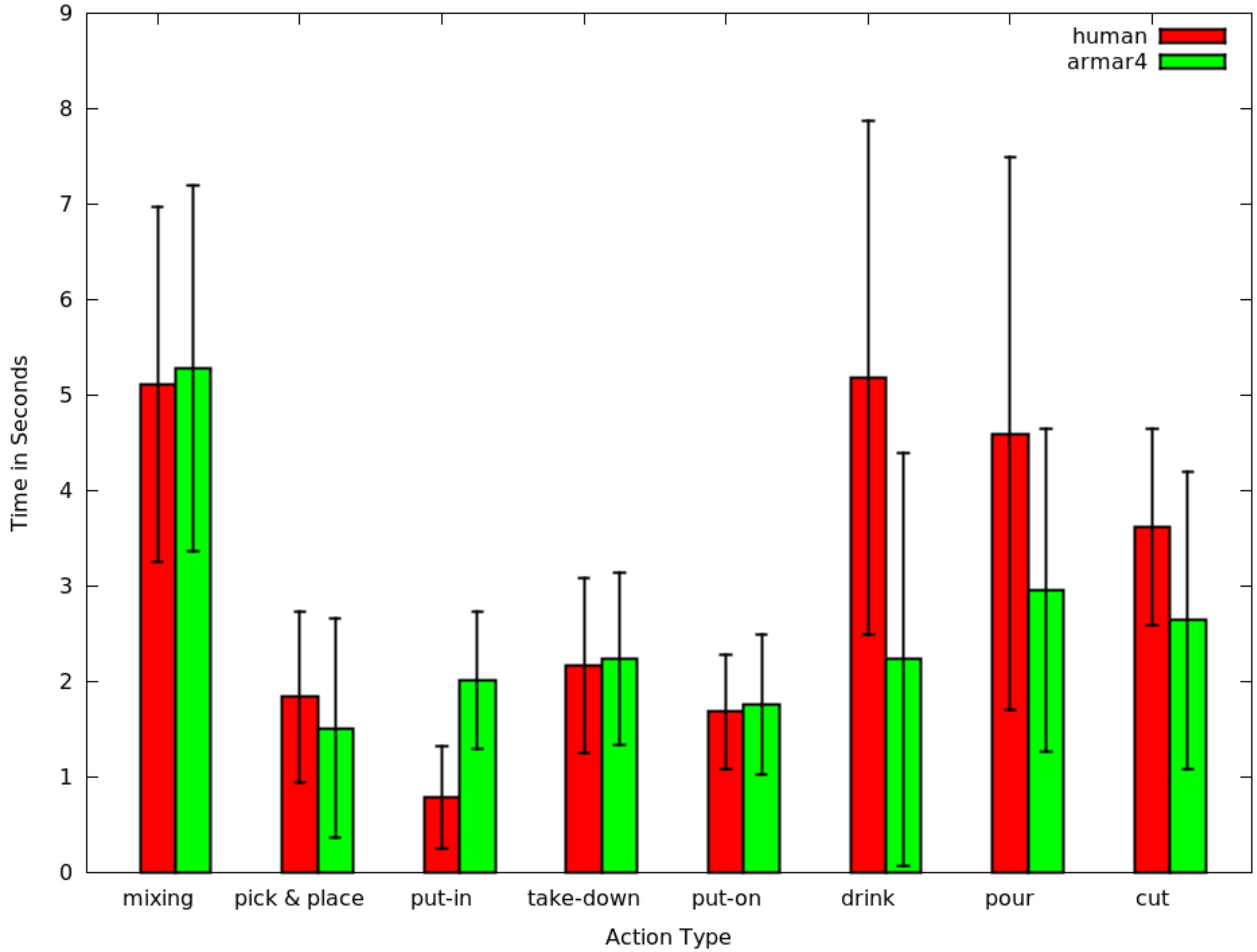


Perception of Time: ARMAR-4 Imitation



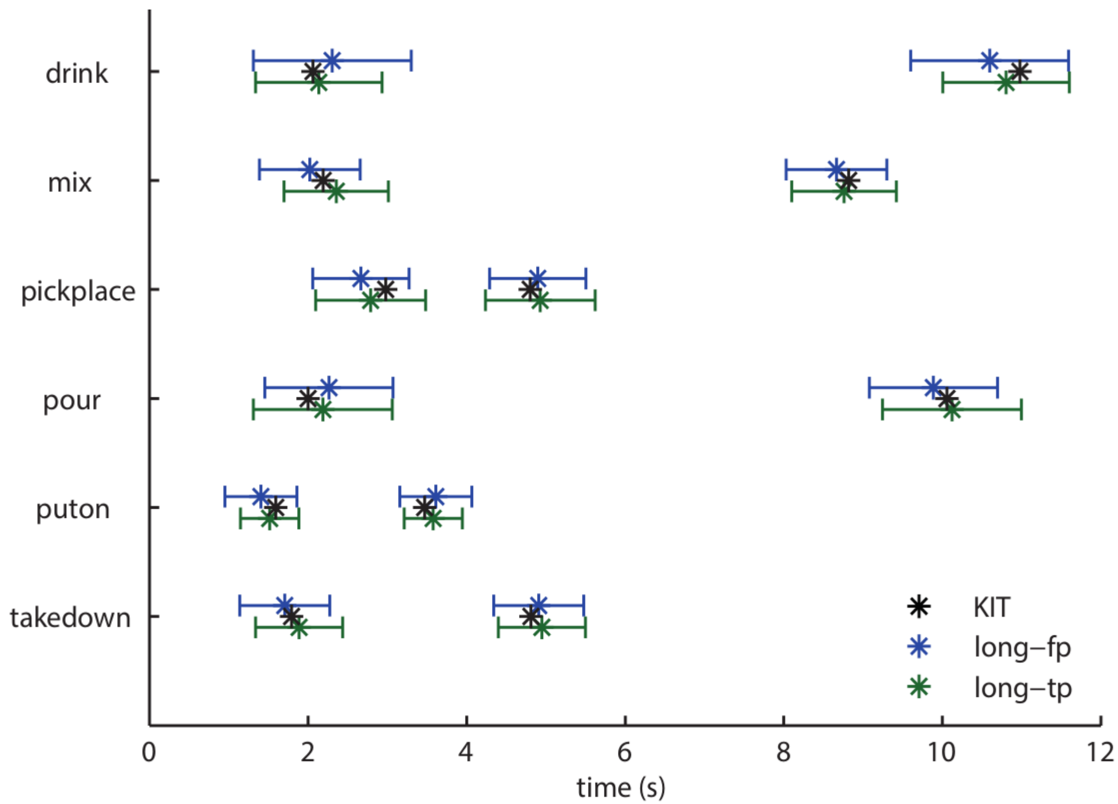


Perception of Time: Human vs ARMAR-4



Perception of Time: Psychological Experiments

Psychological experiments support our action segmentation approach (Collaboration with University of Groningen)



Temporal Scaling

Karlsruhe Institute of Technology

Karlsruhe Institute of Technology

Enriched Manipulation Action Semantics for Robot Execution

Eren Erdal Aksoy, You Zhou, Mirko Wächter and Tamim Asfour

Institute for Anthropomatics and Robotics - High Perfomance Humanoid Technologies Lab (H2T)

KIT – University of the State of Baden-Wuerttemberg and National Laboratory of the Helmholtz Association

www.kit.edu

E. E. Aksoy, Y. Zhou, M. Wächter and T. Asfour, Enriched Manipulation Action Semantics for Robot Execution of Time Constrained Tasks, IEEE/RAS International Conference on Humanoid Robots (Humanoids), pp. 109 - 116, 2016

H2T

Action representation

- Hidden Markov Models (HMM) Humanoids 2006, IJHR 2008
  - Extract key points (KP) in the demonstration
  - Determine key points that are common in multiple demonstrations (common key points: CKP)
  - Reproduction through interpolation between CKPs
- Dynamic movement primitives (DMP) ICRA 2009, T-RO 2010
  - Ijspeert, Nakanishi & Schaal, 2002
  - Trajectory formulation using canonical systems of differential equations
  - Parameters are estimated using locally weighted regression
- Spline-based representations Humanoids 2007
  - fifth order splines that correspond to minimum jerk trajectories to encode the trajectories
  - Time normalize the example trajectories
  - Determine common knot points so that all example trajectories are properly approximated. Similar to via-point, key-points calculation.

Red: relevant for the exam

Karlsruhe Institute of Technology

Dynamic Movement Primitives

- A. J. Ijspeert, J. Nakanishi, and S. Schaal. Learning Attractor Landscapes for Learning Motor Primitives. In Advances in Neural Information Processing Systems 15 (NIPS), 2003.
- A. J. Ijspeert, J. Nakanishi, and S. Schaal, “Movement imitation with nonlinear dynamical systems in humanoid robots,” in Proc. IEEE Int. Conf. Robotics and Automation, Washington, DC, 2002, pp. 1398–1403.
- A. J. Ijspeert, J. Nakanishi, and S. Schaal, “Learning rhythmic movements by demonstration using nonlinear oscillators,” in Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, Lausanne, Switzerland, 2002, pp. 958–963
- P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal (2009) Learning and generalization of motor skills by learning from demonstration. In Proc. IEEE Int. Conf. Robotics and Automation, Kobe, Japan, pp. 763-769.
- Ales Ude, Andrej Gams, Tamim Asfour, and Jun Morimoto. Task-Specific Generalization of Discrete and Periodic Dynamic Movement Primitives. IEEE Transactions on Robotics, 26(5): 800-815, 2010
- J. Ernesti, L. Righetti, M. Do, T. Asfour, and S. Schaal (2012). Encoding of Periodic and their Transient Motions by a Single Dynamic Movement Primitive, Humanoids 2012 - Best paper award Finalist -

H2T

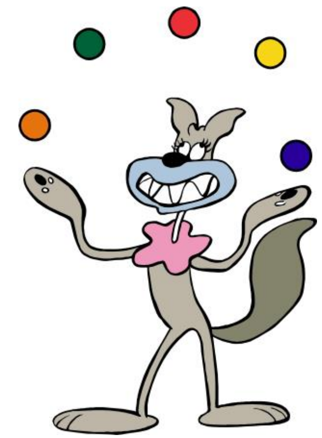
DMPs

See also Chapter 7 in Mechano-Informatics

The following slides are not part of the material of the lecture and thus not relevant for the exam

Action representation – biological systems

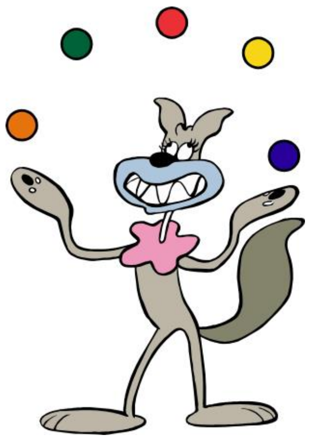
- Motor control in biological systems
  - Biological systems exhibit a continuous stream of movements in their daily activities.
  - Movements can be
    - Rhythmic (e.g. locomotion, juggling)
    - Discrete (e.g. tennis swing)



- Assumption: Movement sequences consist of segments (→ motion primitives), which can be executed either in sequence or with overlap.

Dynamic Movement Primitive (DMP)

- Two types of motions
  - Discrete (e.g. tennis swing)
  - Periodic/Rhythmic (e.g. locomotion, juggling)



- Different stability requirements
  - Discrete: A discrete movement is stable, if the movement shows point attraction behavior, which means that it will stop somewhere when time goes infinite.
  - Periodic: A periodic movement is stable, if the movement shows path attraction behavior, which means that it will keep the same path when time goes infinite.

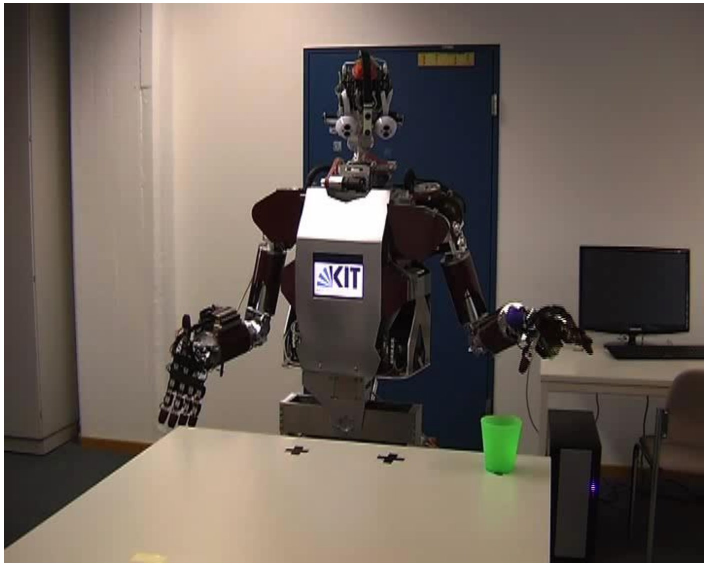
Action representation - robotics

- Mathematical models to represent robot motions are needed → action representation
- Desired properties of such an action representation:
  - Reusability: motion primitives can be composed to action sequences in order to solve complex tasks
  - Flexibility: execution under different conditions, such as changing start and goal positions, velocity, acceleration
  - Adaptivity: goal-directed execution in dynamic environments (e.g. obstacle avoidance)
  - Portability: use the same action representation on different robots
- To attain these properties, models are needed that are capable of generalizing specific situations.
  - → Dynamic movement primitives (DMPs); more soon!

Dynamic Movement Primitives (DMP)

- Formalism to describe functions, e.g. robot movements, as **dynamic system**
- Learnable from **demonstration**
- Movement is **generalized** to arbitrary
  - start points and goals
  - duration

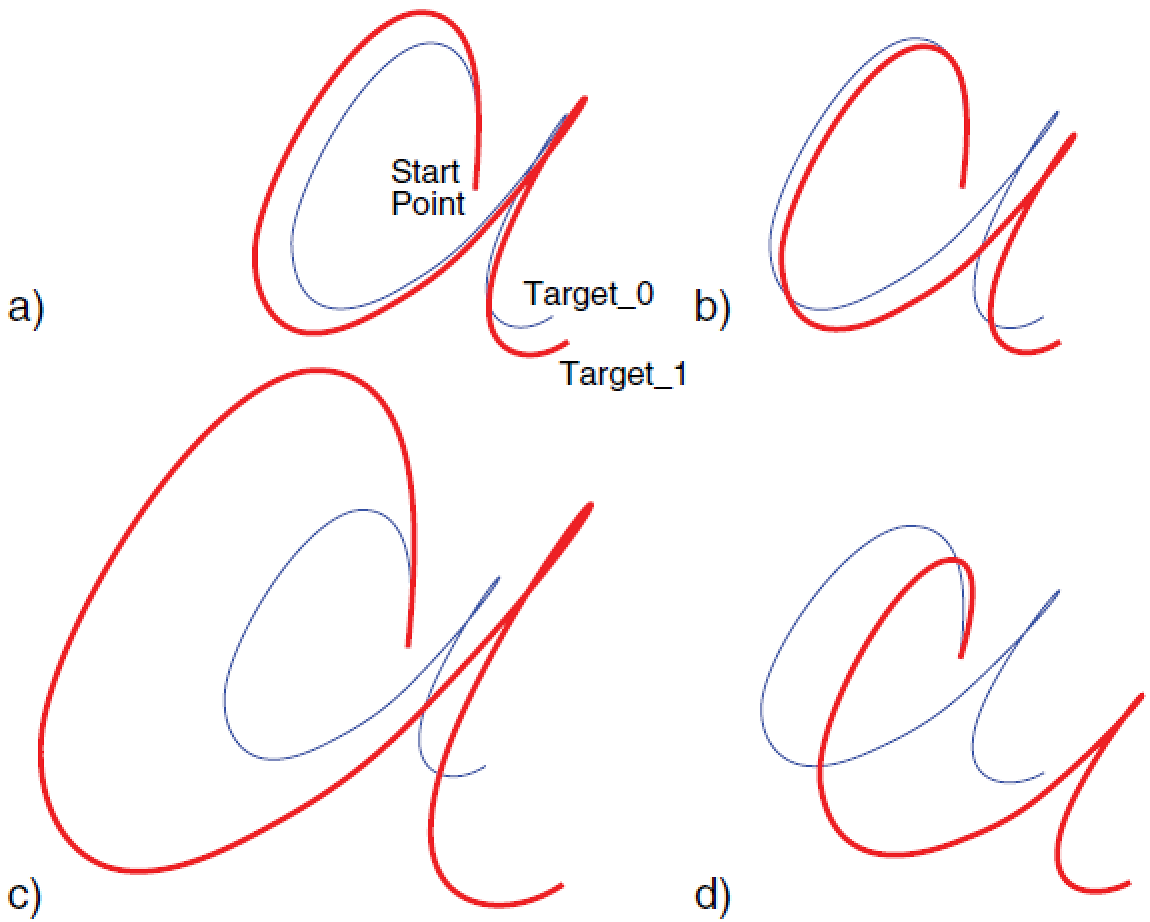
- Extendable for **online feedback**
  - Changing goals
  - Changing duration
  - Force feedback
  - Collision avoidance



Wiping

- Movement generation is a major part of robotics
- Pure imitation is not applicable to the real world
  - Adaptation to current task needed
    - Objects are at different positions
    - Execution must be quicker/slower
  - Handling of real world perturbation
    - execution is delayed

Dynamic movement primitives (DMPs): Motivation



Introduction to Dynamic Movement Primitives

- Given
  - Demonstration by human (Position & velocity for each time step): $\boldsymbol{y}_D, \boldsymbol{v}_D$
  - Arbitrary start position y_0, velocity v_0
  - Arbitrary goal Position g
  - Arbitrary duration (i.e. temporal factor τ)

- Desired
  - Trajectory from y_0 to g in duration τ with characteristic shape of demonstration $\boldsymbol{y}_D$

- Solution: Critically damped spring-mass system with perturbation

Damped spring-mass system

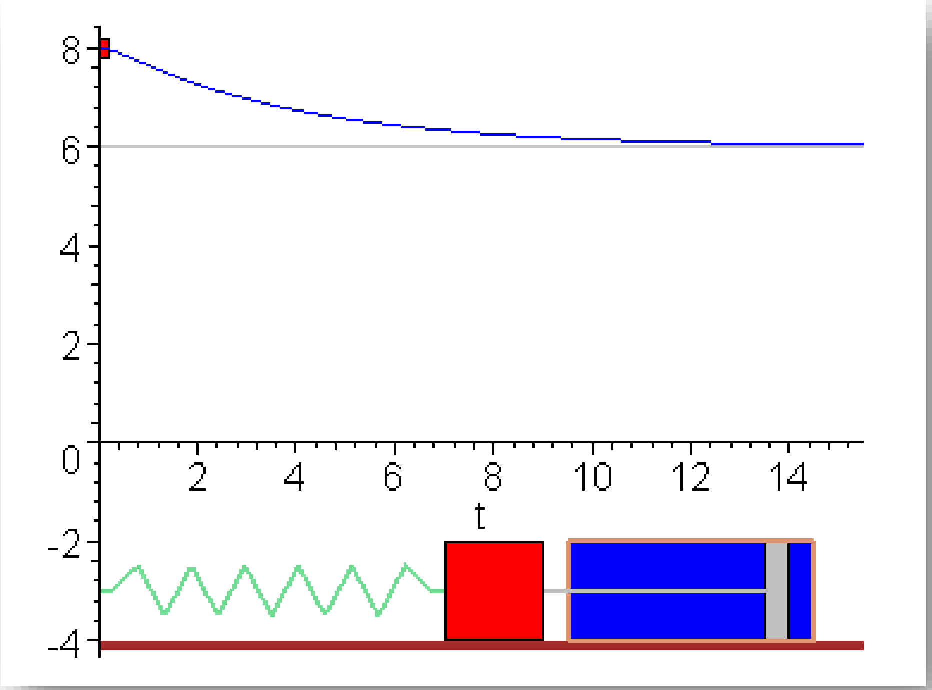
$$\ddot{x} = \boxed{K(g - x)} - \boxed{D\dot{x}}$$

Spring term:
goal attractor

Damping term:
limits acceleration

g : goal
x : current position
D: damping constant
K: spring constant

- 2nd order ordinary differential equation (ODE)
- Globally stable attractor system
  - Converges to *g*

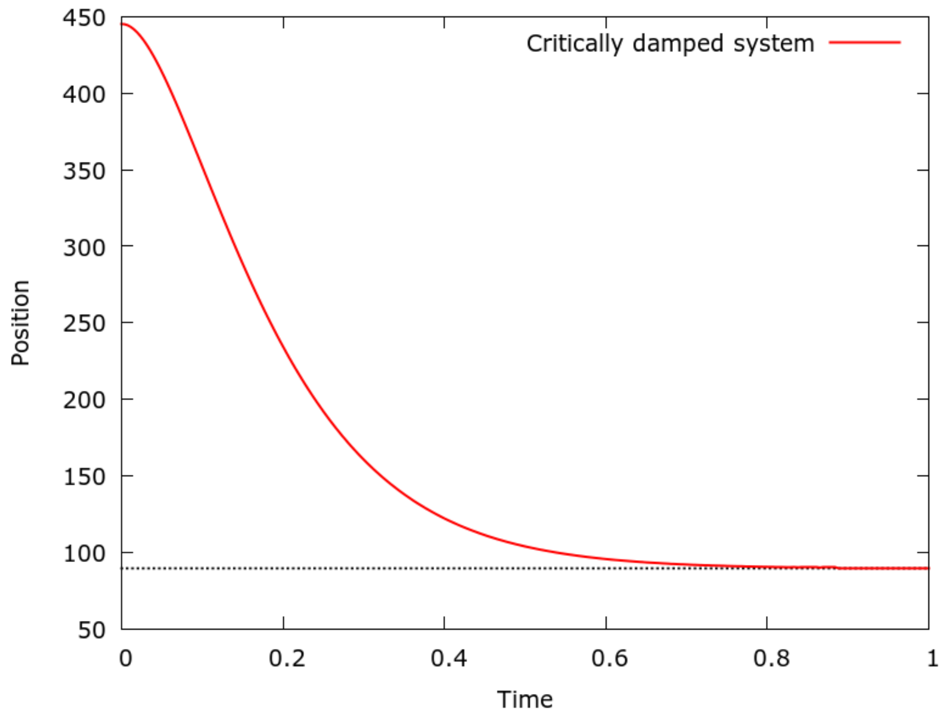


Damped spring-mass system

To easier computation the ODE is transformed into a 1st order system: Substitution of variables so that only the first derivative appears

$$\dot{\boldsymbol{v}} = K(g - x) - D\boldsymbol{v}$$
$$\dot{\boldsymbol{x}} = \boldsymbol{v}$$

g : goal
x : current position
v: current velocity
D: damping constant
K: spring constant



Temporal scaling of the system

For temporal scaling (shorten or lengthen) a simple temporal factor is introduced:

$\tau > 1$: Mass-spring is slower

$\tau < 1$: Mass-spring is faster

$$\boxed{\begin{aligned} \boldsymbol{\tau\dot{v}} &= K(g - x) - D \cdot v \\ \boldsymbol{\tau\dot{x}} &= v \end{aligned}}$$

τ : temporal factor
g : goal
x : current position
v: current velocity
D: damping constant
K: spring constant

Transformation System

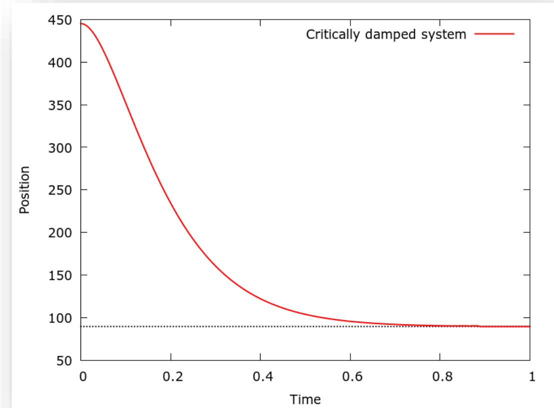
To shape the trajectory a perturbation force term is introduced:

Scaled perturbation term

$$\tau \dot{v} = K(g - x) - D \cdot v + \overbrace{(g - x_0) \cdot f \cdot u}$$

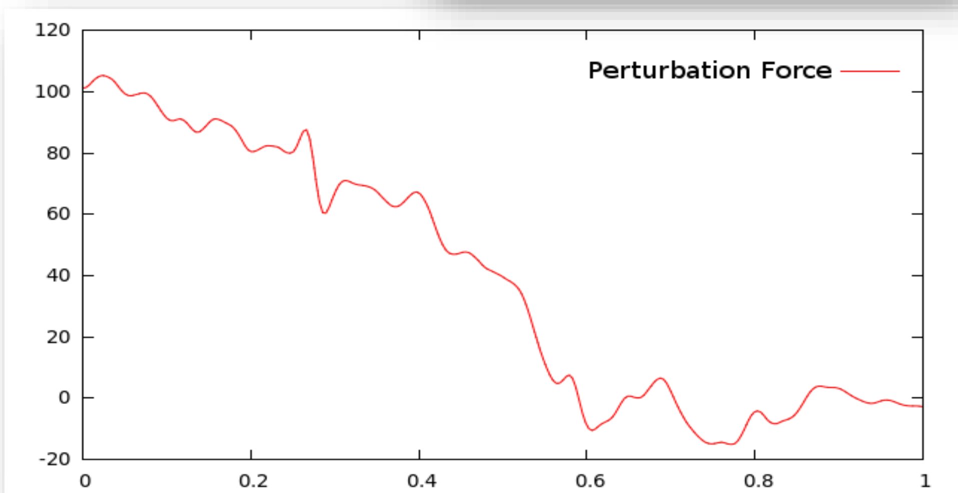
$$\tau \dot{x} = v$$

Perturbation force



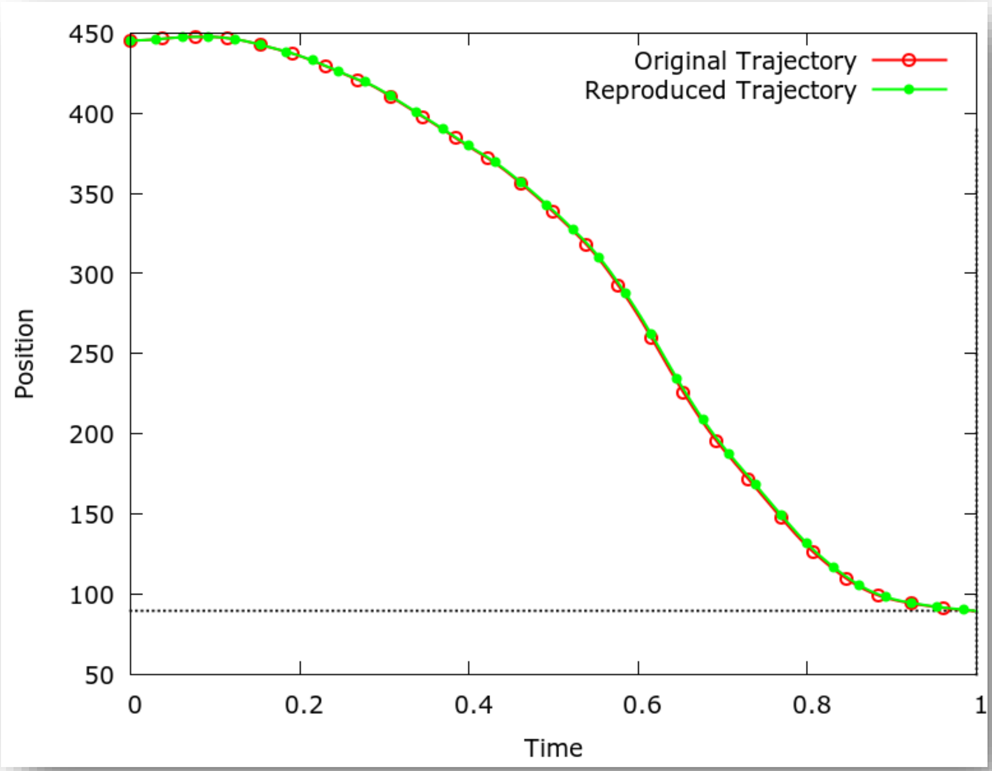
This ODE system is called the transformation system of a DMP

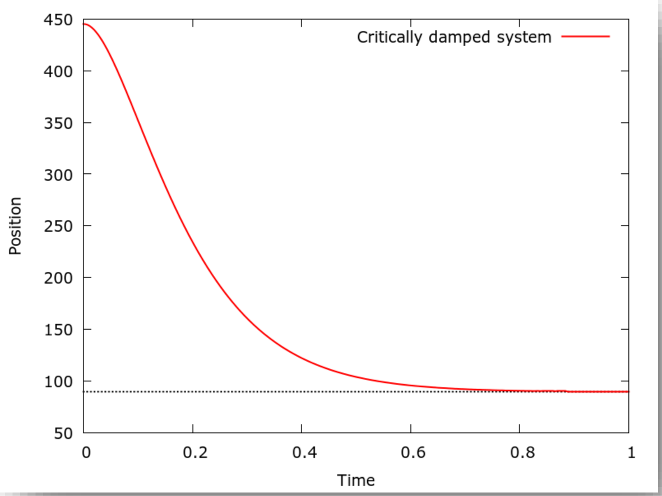
The perturbation force is learned from demonstration

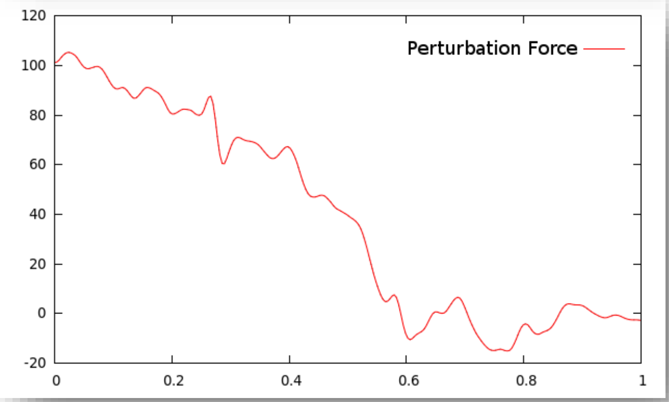


Transformation System (2)

Resulting trajectory of transformation system

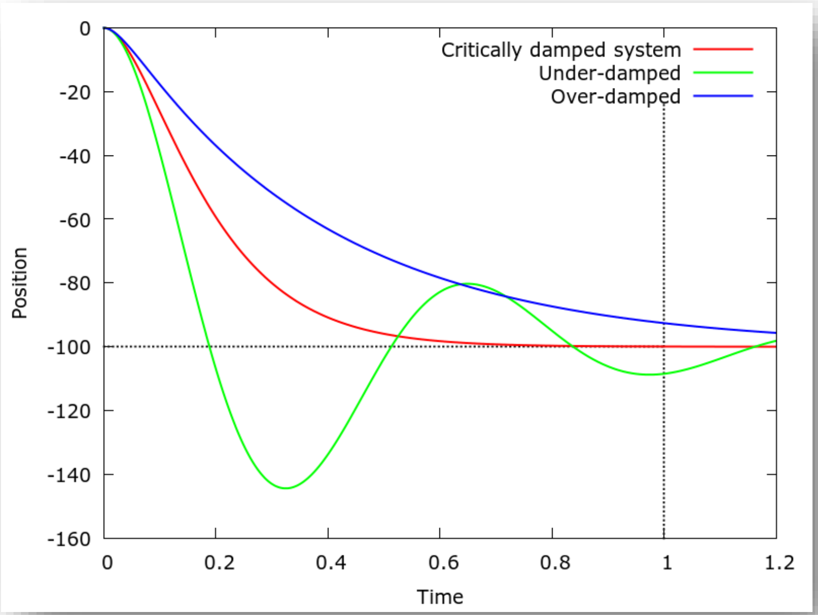






Value selection of constants *D* and *K*

- A damped spring-mass system oscillates around the goal: not desired for movement generation
- Solution: Critically damped spring-mass system
  - Special relation of spring constant *K* and damping constant *D*
  - Damping is just as strong to compensate overshooting over goal
  - But: goal is still reached as fast as possible

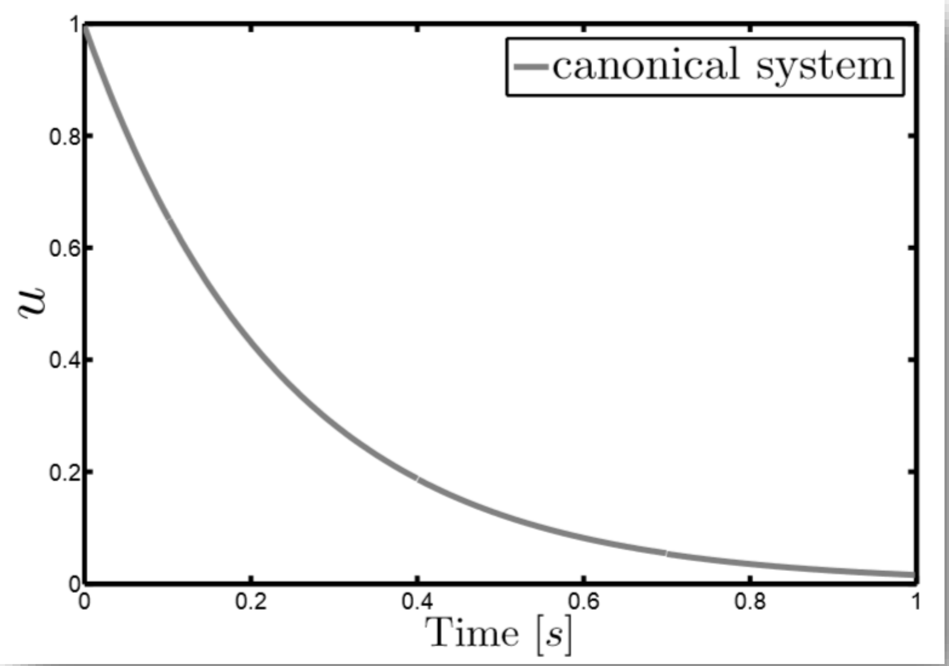


The canonical system: Inducing time independence

- If the perturbation term f depends explicitly on time, the complete execution depends on time
- Solution: Use of a canonical system (phase variable) instead of explicit time dependence

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

$$\alpha = -\log(u_{min})\,\tau$$
$$u_{min} = 0.001$$
$$u(0) = 1$$



DMP: Short Recapitulation

- The Canonical system
  - State of the DMP in time
  - Drives the perturbation to control the transformation system
- The Transformation system and the Perturbation
  - Are adapted to the demo trajectory
  - Generate the robot motion
  - Can generalize the learned trajectory to new conditions

Canonical system $$\tau \dot{u} = -\alpha u$$

Transformation system $$\tau \dot{v} = K(g - x) - Dv + (g - x_0) f(u) u$$
$$\tau \dot{x} = v$$

Perturbation: Shaping the trajectory

- The perturbation function *f* determines the shape of the trajectory
  - It "pushes" the spring-mass-system away from it's original path
  - Additional term in the acceleration calculation
- Non-linear function representing the demonstrated trajectory
- *f* depends on the canonical system *u(t)*
- Resolve the transformation system to *f*

$$f = \frac{-K(g - x) + Dv + \tau\dot{v}}{(g - x_0)u}$$

- *How do we find f and how do we store it?*

How to learn the trajectory of the demonstration

- Given: Demonstration y (sequence of position, velocities and acceleration with timestamps) of length T
- Wanted: Perturbation force for complete demonstration

- Calculate f for all timestamps with:

$$\begin{aligned} g &= y(T) \\ x_0 &= y(0) \\ x, v, \dot{v} &= y, \dot{y}, \ddot{y} \\ \tau &= T \end{aligned}$$

- These are only the perturbation forces at the given timestamps
  - Continuous representation of the perturbation force desired!

How to approximate the perturbation force

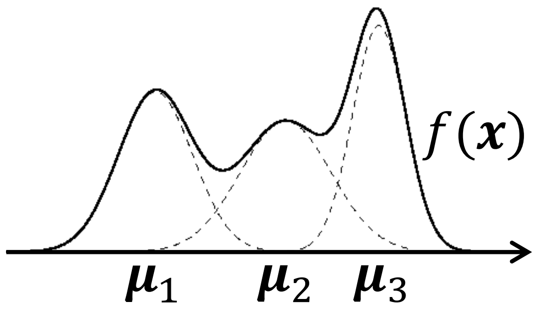
- Approximate the discrete values for perturbation function *f* with a continuous representation

- Approximation can be done with any function approximation algorithm:
  - Weighted Radial Basis Functions (RBF)
  - Splines
  - LWPR
  - …
- Here: RBF

Radial Basis Function (RBF) Networks I

- Function approximation

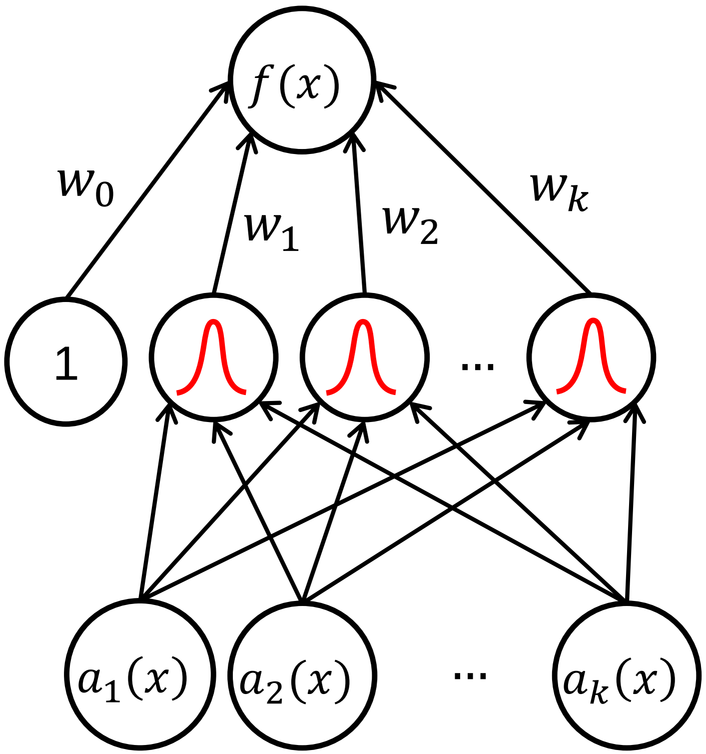
$$\hat{f}(\boldsymbol{x}) = w_0 + \sum_{u=1}^{k} w_u \cdot K_u(\boldsymbol{x})$$



- Kernel function

$$K_u(\boldsymbol{x}) = e^{-\frac{1}{2\sigma_u^2}\|\boldsymbol{\mu}_u - \boldsymbol{x}\|^2}$$

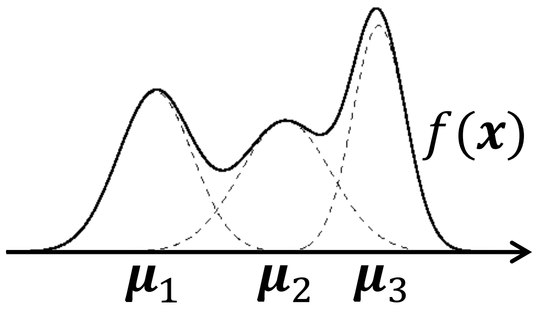
- Modelled as a neural network
  - One hidden layer
  - One output neuron
- Related to locally weighted regression

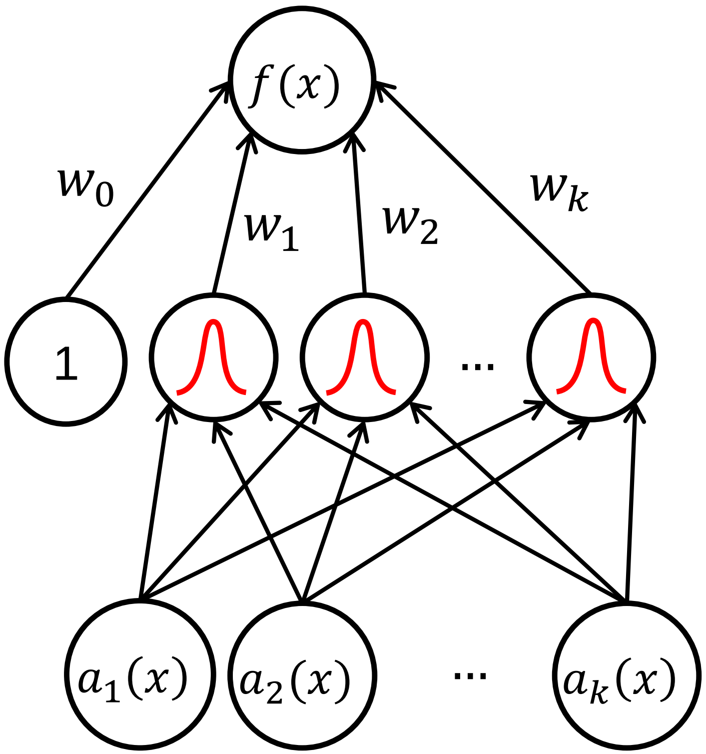


Radial Basis Function (RBF) Networks II

- Training a RBF network
- Two steps

1. determine the receptive fields
   - Number k of hidden neurons
   - Parameters μ_u and σ_u
     - One neuron per sample or
     - Uniformly distributed in X or
     - Based on clustering of the training data or
     - Expectation Maximization (EM)
       (finds optimal μ_u and σ_u)
     - …
2. determine the weights w_u
   - As in regular neural networks
   - While μ_u and σ_u are fixed





How to approximate the perturbation force

- Here: Weighted RBFs:
  - Locally weighted regression

$$f_{approx.}(u) = \frac{\sum_{i=1}^{N} w_i \psi_i(u)}{\sum_{i=1}^{N} \psi_i(u)}$$

  - RBF:

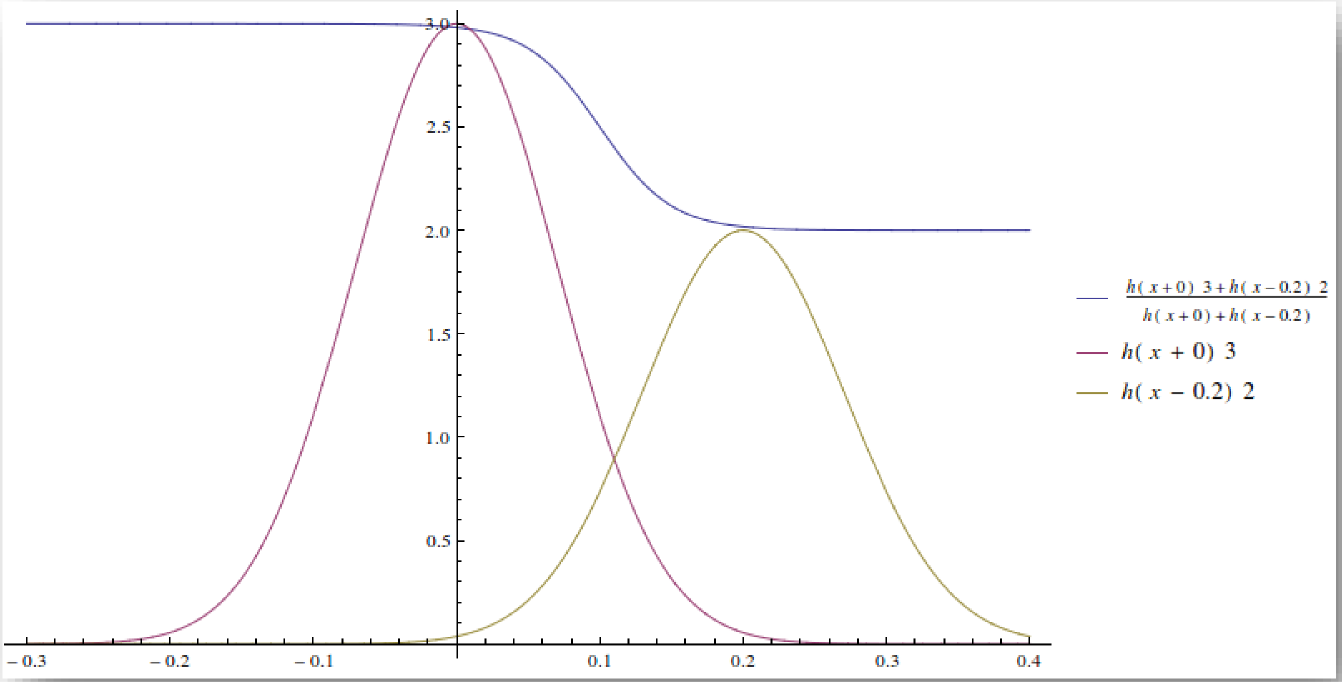
$$\psi_i(u) = e^{h_i \, (u - c_i)^2}$$

h_i: width of RBF
c_i: center of RBF
w_i: weight of RBF

  - One dimensional optimization problem to find the weights w_i with fixed h_i and c_i that fit the discrete values of f best
  - Weights w_i are stored for the execution of the DMP, h_i and c_i have fixed values

How to approximate the perturbation force (2)

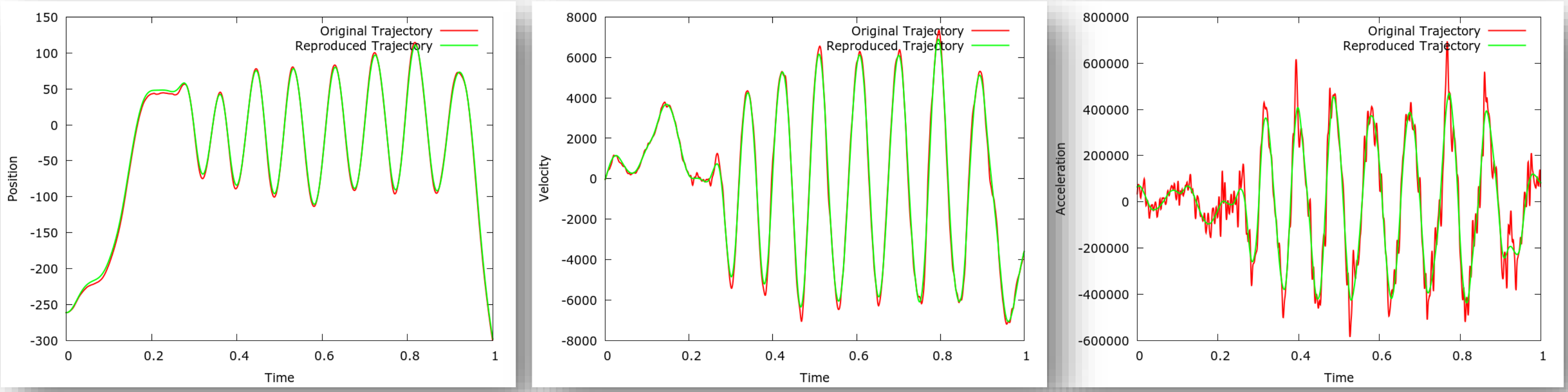
- Illustration: 2 weighted RBFs
  - Proximity to support points is determined by RBF width



Approximation accuracy depends on the number of basis functions, the more basis functions, the less approximation error

Benefits of approximation

- Weighted RBFs can be used to remove jitter from original demonstration



- Weighted RBFs reduce memory consumption
  - In this example: 70 RBFs instead of 1500 position & velocity values

DMP Formulierung

canonical system: $\tau \dot{u} = -\alpha u$

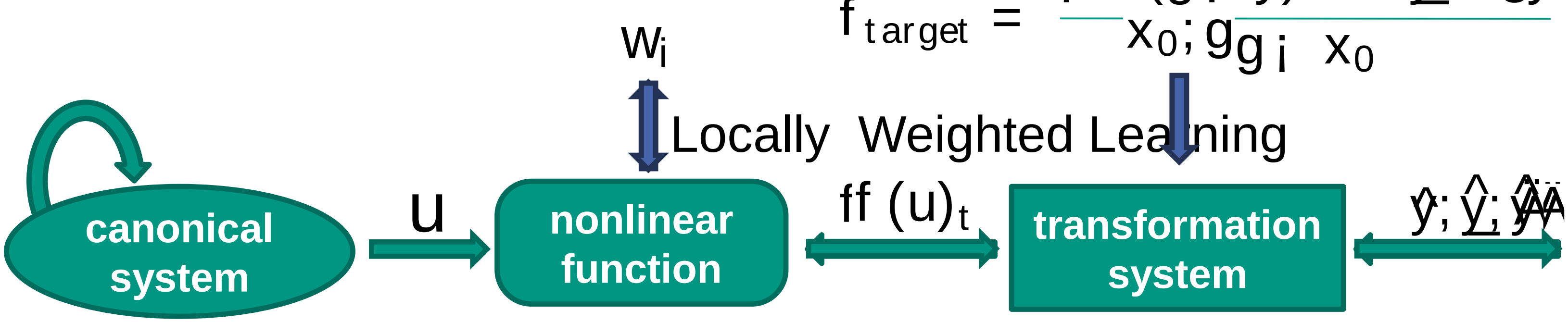
nonlinear function: $f(u) = \dfrac{\sum_i \psi_i(u) w_i u}{\sum_i \psi_i(u)}$ $\qquad \psi_i(u) = e^{-h_i (u - c_i)^2}$

transformation system: $\tau \dot{v} = K(g - x) - Dv + (g - x_0) f$

$\tau \dot{x} = v$

$$f_{target} = \frac{-K(g - y) + D\dot{y} + \tau \ddot{y}}{g - x_0}$$

$x_0; g$



n-dimensionale DMPs

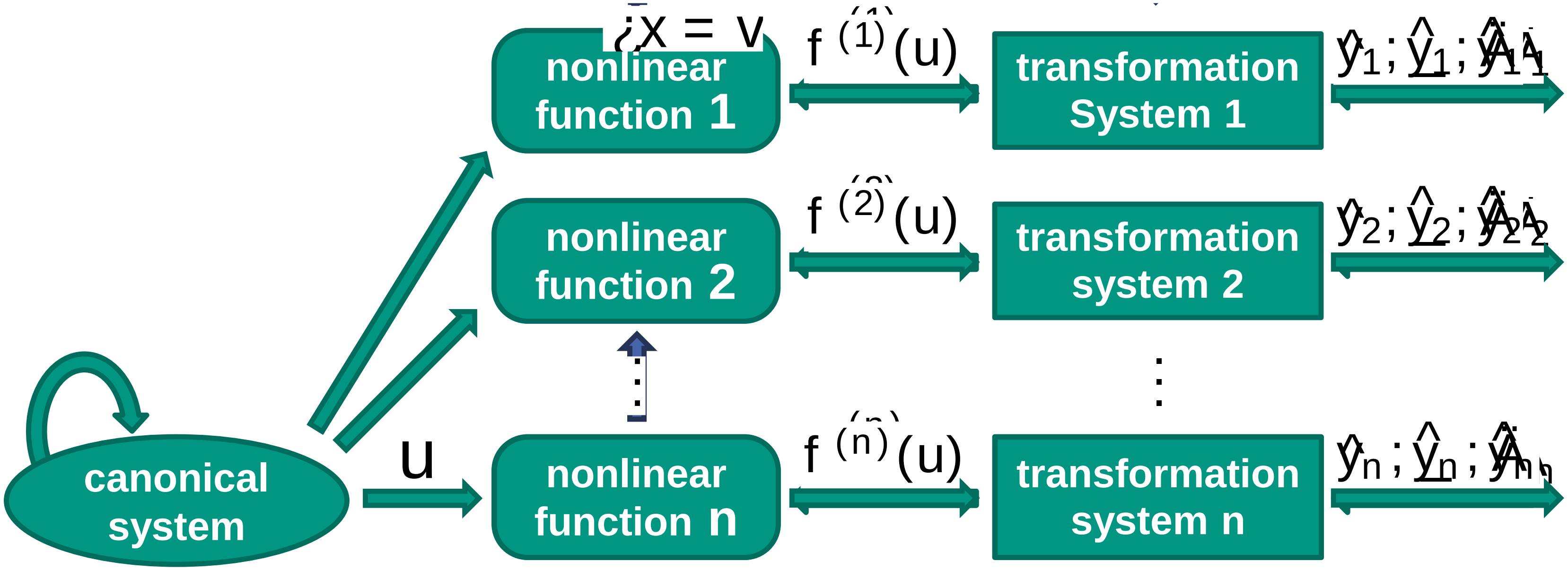
canonical system: $\tau \dot{u} = -\alpha u$

nonlinear function: $f(u) = \frac{\sum_i \psi_i(u) w_i u}{\sum_i \psi_i(u)} \qquad \psi_i(u) = e^{-h_i (u - c_i)^2}$

$w_i \qquad x_0, g$

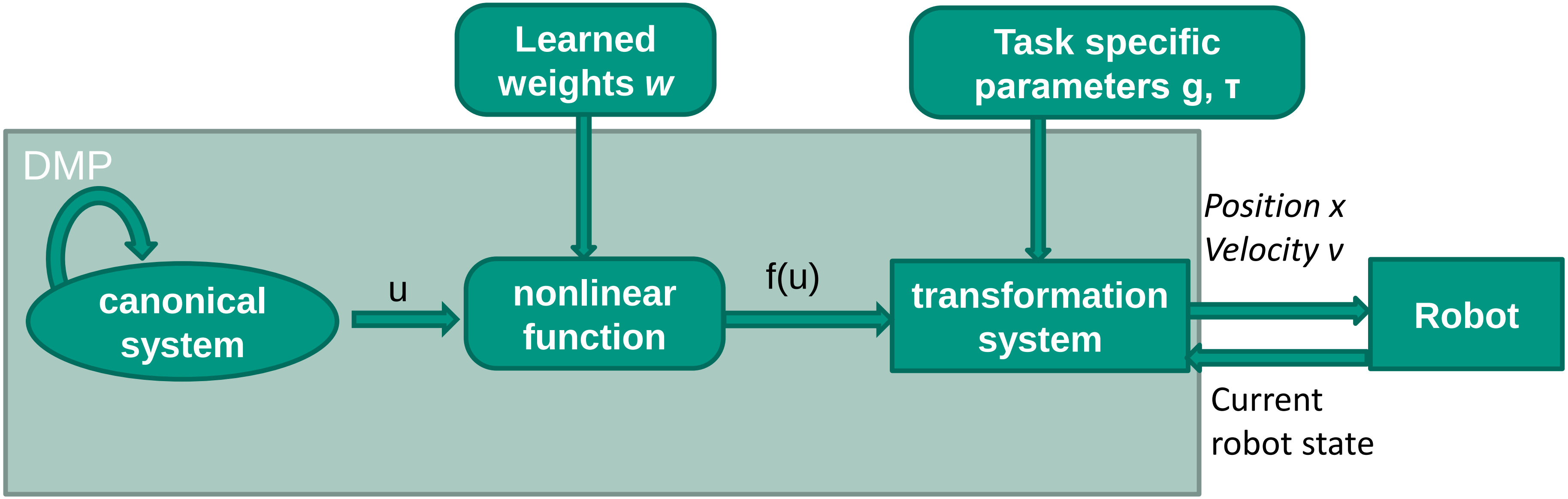
transformation system: $\tau \dot{v} = K(g - x) - Dv + (g - x_0) f$

$\tau \dot{x} = v$

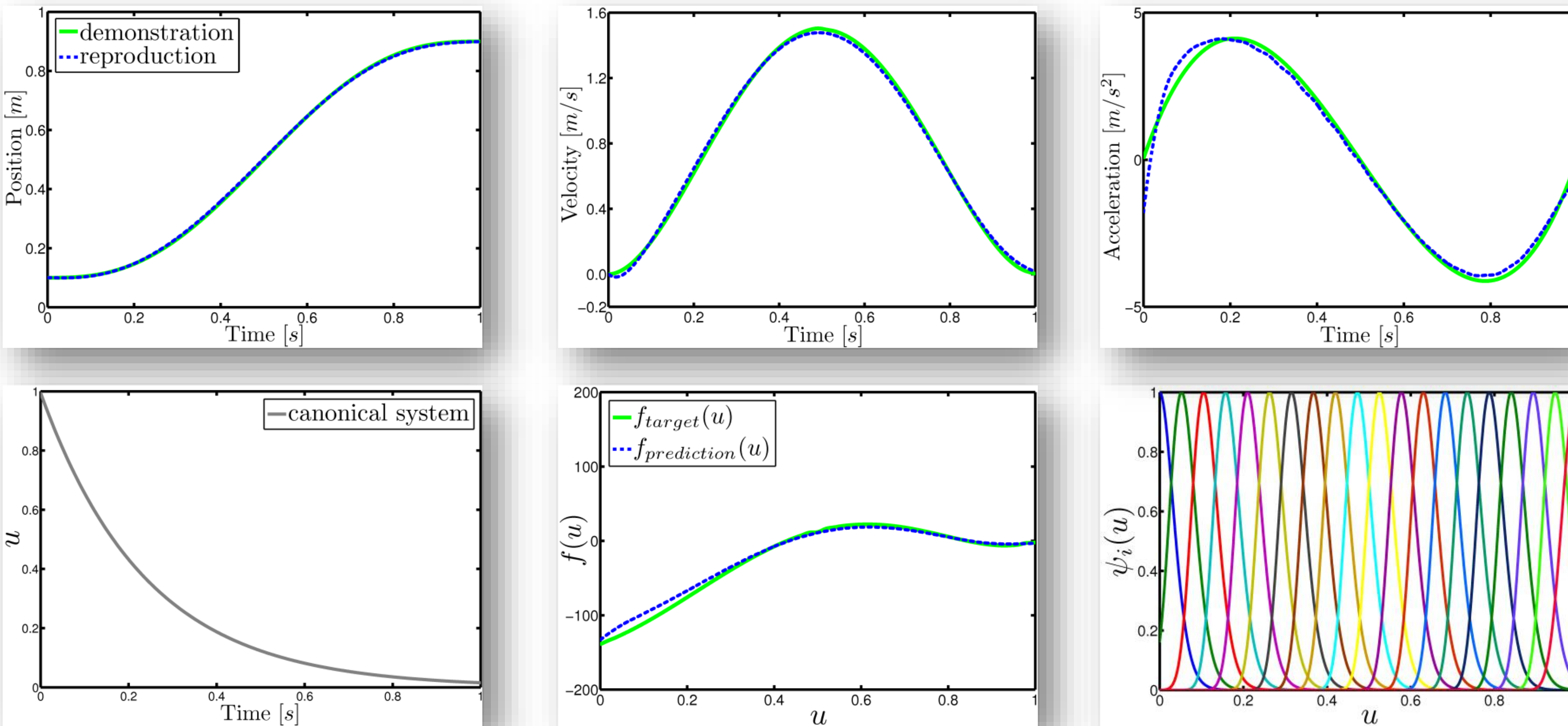


Execution of a DMP

- Parameterize the DMP
  - Initial state (position & velocity)
  - Goal position
  - Temporal factor
  - Choose a demonstration (stored as approximation of perturbation force)
- Integrate the ordinary differential equation



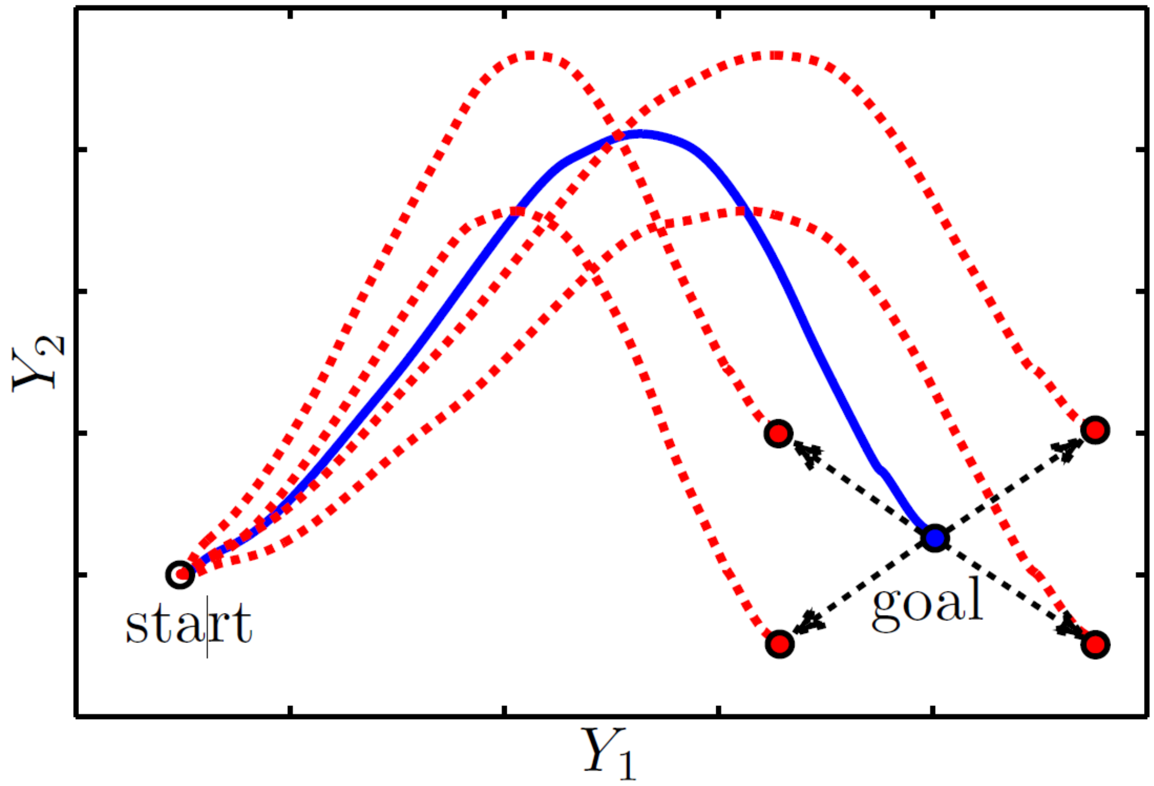
Example



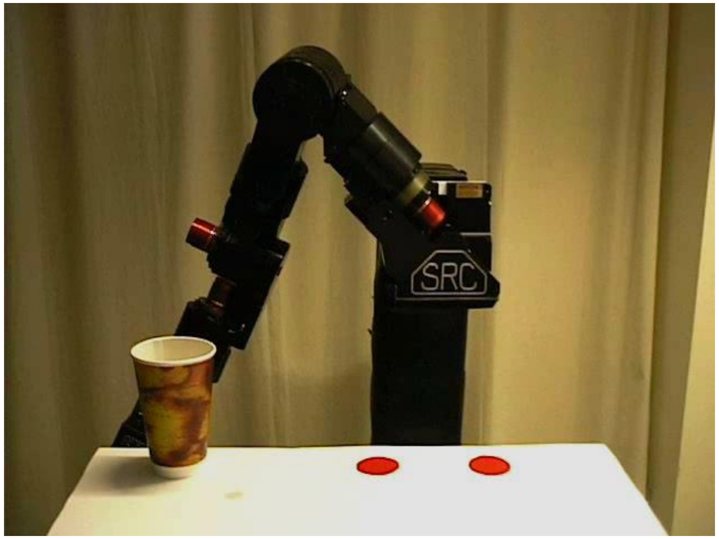
demonstration
reproduction
Position $[m]$
Time $[s]$
Velocity $[m/s]$
Time $[s]$
Acceleration $[m/s^2]$
Time $[s]$
canonical system
u
Time $[s]$
$f_{target}(u)$
$f_{prediction}(u)$
$f(u)$
u
$\psi_i(u)$
u

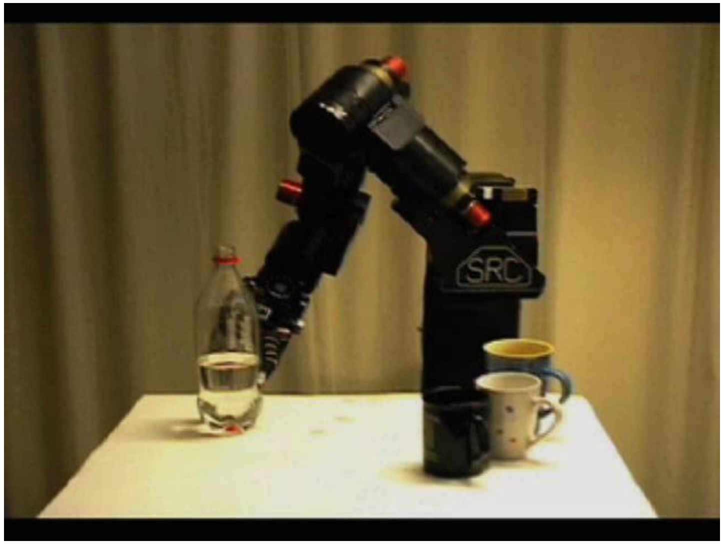
Changing the goal

- DMPs can generalize to new goal while keeping the characteristic shape of the trajectory



Examples





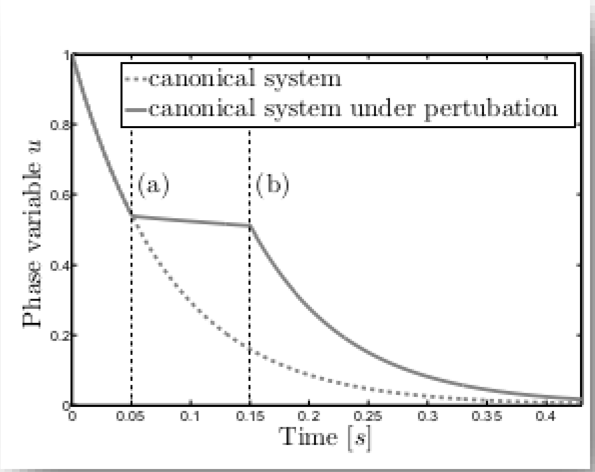
Online Feedback: External Perturbation Force

- One way to deal with an external perturbation is phase stopping
  - Phase slows down until external perturbation force is gone
  - E.g. a person holds the arm of the robot

- Modifying the canonical system for phase stopping:

$$\tau \dot{u} = \frac{-\alpha u}{1 + \alpha_{pu} |\tilde{x} - x|}$$

x: desired position
$\tilde{x}$: actual position



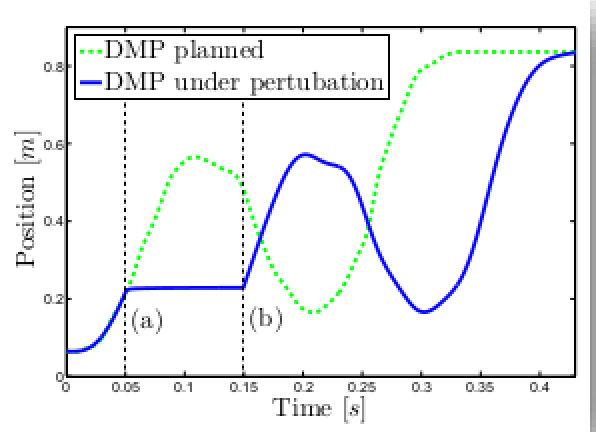
  - α_{pu} controls how fast the phase slows down
  - Increasing error in position slows the phase down

Online Feedback: External Perturbation Force

- One way to deal with an external perturbation is phase stopping
  - Phase slows down until external perturbation force is gone
  - E.g. a person holds the arm of the robot

- Similarly the transformation system is changed:

$$\tau\dot{v} = K(g - x) - Dv + (g - x_0)fu$$
$$\tau\dot{x} = v + \alpha_{px}|\tilde{x} - x|$$



x: desired position
$\tilde{x}$: actual position

An error in position moves the desired position towards the actual position

Online Feedback: Changing Goal

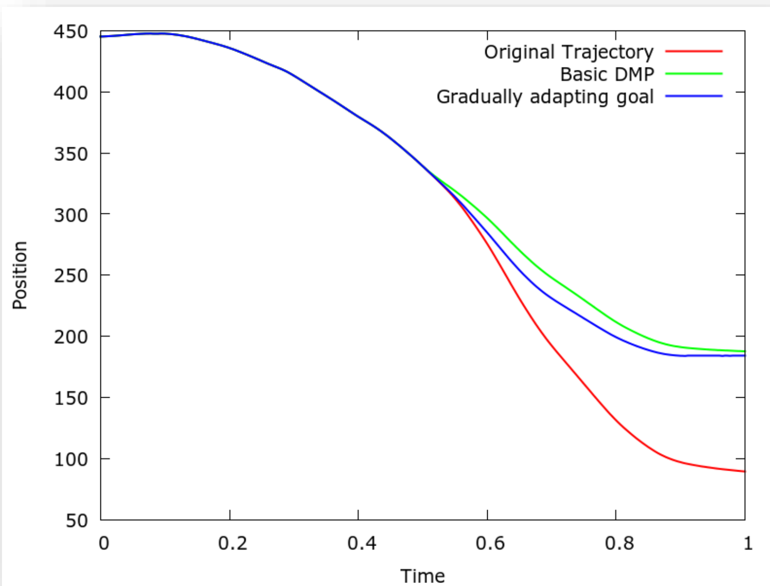
- Simple online changing of the goal would lead to a jump in the acceleration

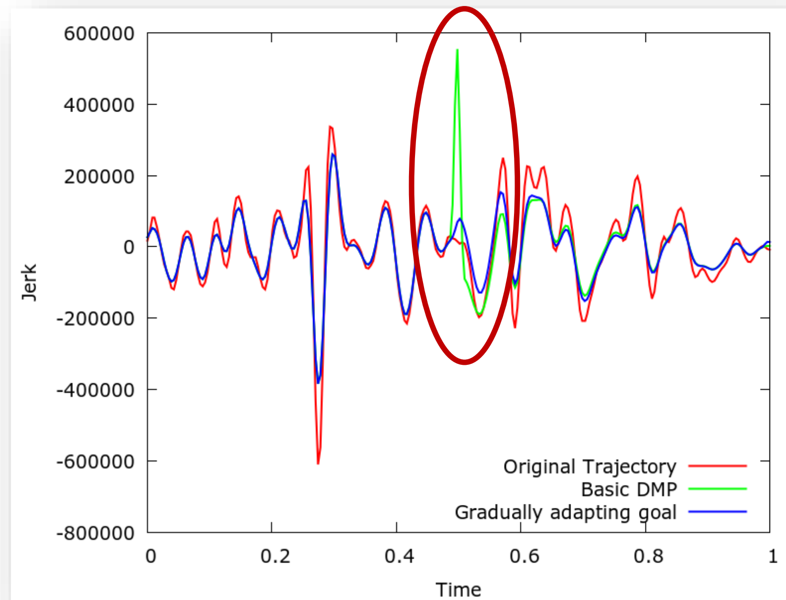
$$\tau\dot{v} = K(g - x) - Dv + (g - x_0)f(u)u$$
$$\tau\dot{x} = v$$

- Filtering the goal with the ODE:

$$\dot{g} = \lambda(g - g_d)$$

g_d: Desired goal
λ: Goal change rate factor





Periodic DMPs

Observation:

- Most periodic motions start with a discrete part:
  - Stirring: first move the hand to the vessel containing the liquid.
  - Wiping: first move the hand to the surface to be wiped.
  - Peeling: first bring the peeler to the potato.
  - Cutting: first move the hand to the object.
  - Walking – first step vs. all other steps.
  - Juggling – bringing the balls into the air vs. juggling itself.
- We call this non-periodic part the transient.

Periodic DMPs with transients

- A periodic motion does not have a fixed start point
  - which phase should be started at?
- Different transients possible for one periodic motion, e.g. juggling:
  - 2 Balls in the left hand vs.
  - 1 Ball in the left, 1 Ball in the right hand
- The transients cannot be reconstructed from the period itself
  - Have to be learned!
- No transient $\Rightarrow$ no well-defined start of periodic motion

Periodic DMPs

- Some movements repeat themselves periodically
  - Wiping, waving, stirring, walking, screwing
- Change
  - Canonical system with: $\dot{\phi} = \Omega$ with Ω as frequency of periodic movement
  - τ with $\frac{1}{\Omega}$
  - Perturbation force approximation:

$$f(\phi) = \frac{\sum_{i=1}^{N} w_i \Gamma_i(\phi)}{\sum_{i=1}^{N} \Gamma_i(\phi)} r$$

$$\Gamma_i(\phi) = e^{h_i \left(\cos(\phi - c_i) - 1\right)}$$

  - Resulting transformation system:

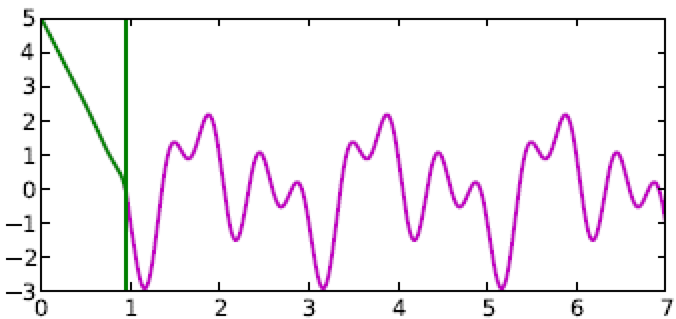
$$\dot{v} = \Omega K(g - x) - Dv + (g - x_0) f(\phi) \phi$$

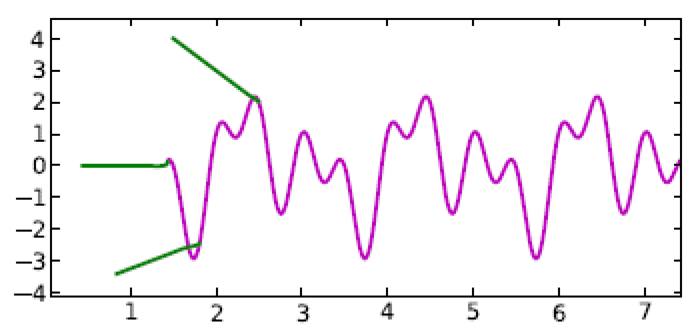
$$\dot{x} = \Omega v$$

Extension of periodic DMP

- Extend the periodic DMP formulation to
  - encode the periodic pattern
  - and all corresponding transients
- into a single dynamical system.

- Example target functions:





One DMP for everything

Solution: one DMP for everything.

- We want to have a DMP which is able to
  - encode the periodic movement itself and
  - all corresponding transients
- into a single dynamical system.

DMPs in General

A DMP consists of two parts:

$$\begin{cases} \dot{s}(t) = \text{Canonical}(t, s), & (1) \\ \dot{y}(t) = \text{Transform}(t, y) + \text{Perturbation}(s). & (2) \end{cases}$$

- The Canonical system
  - State of the DMP in time
  - Drives the perturbation to control the transformation system
- The Transformation system
  - Generates the motion of the robot according to the state of the canonical system

The Canonical System

Other DMP works addressing periodic movements use a one-dimensional Canonical System, i.e. $s(t)$ in $\boldsymbol{R}$.

We use a two-dimensional Canonical System,
i.e. $s(t) := (\varphi(t), r(t))$ in $\boldsymbol{R}\times(0,\infty)$ for φ, r solution of

$$\begin{cases} \dot{\phi} = \Omega, \\ \dot{r} = \eta(\mu^{\alpha} - r^{\alpha})r^{\beta}, \\ \phi(0) = \phi_0,\ r(0) = r_0 \end{cases} \tag{3}$$

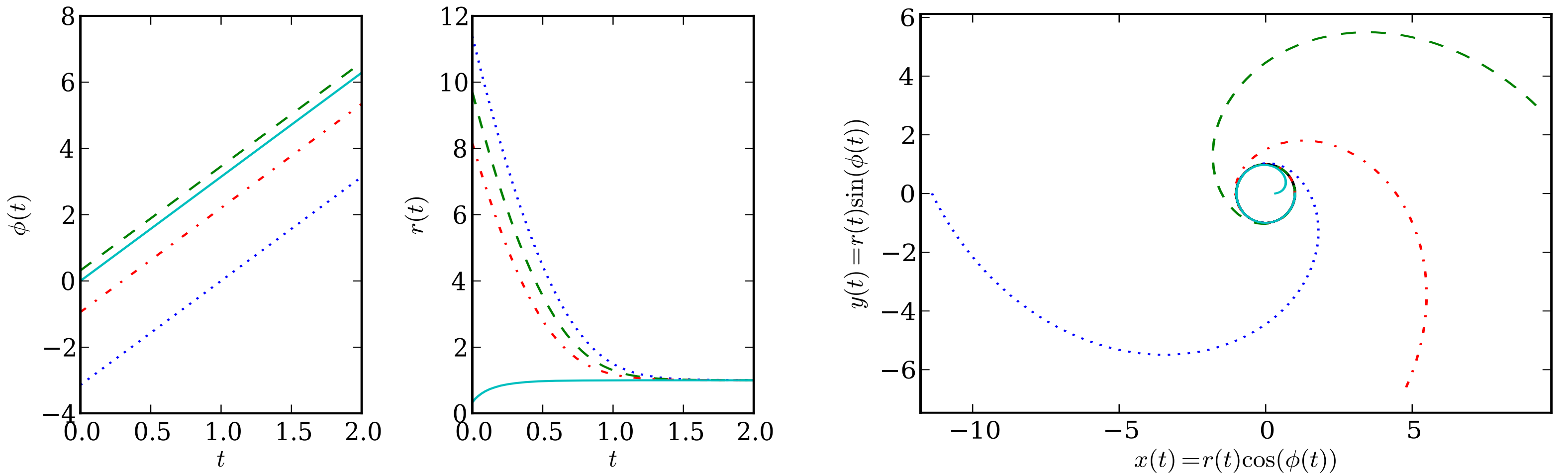
with $\mu, \eta, \alpha, \beta > 0$ constant, $\Omega > 0$ phase frequency,
(φ_0, r_0) in $\boldsymbol{R}\times(0,\infty)$ initial conditions.

The Canonical System - Properties

Properties of the Canonical System:

- $r(t) \rightarrow \mu$ for $t \rightarrow \infty$ (monotonously)
- α, β, η define the speed and shape of the convergence
- Interpret (φ, r) as polar coordinates

Example runs:



The Transformation System

Well known transformation system.

We use the transformation system introduced by [Ijspeert et al. 2002]

$$\begin{cases} \dot{z} = \Omega\Big(\alpha_z\big(\beta_z(g-y)-z\big)+f(\phi,r)\Big), \\ \dot{y} = \Omega z \end{cases} \tag{4}$$

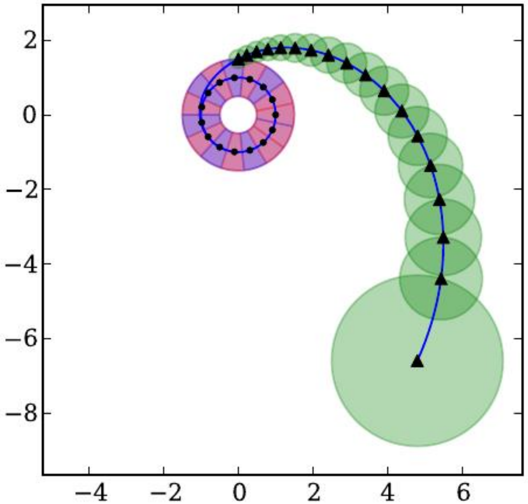
with α_z, $\beta_z > 0$ constants, g in $\boldsymbol{R}$ the anchor point.

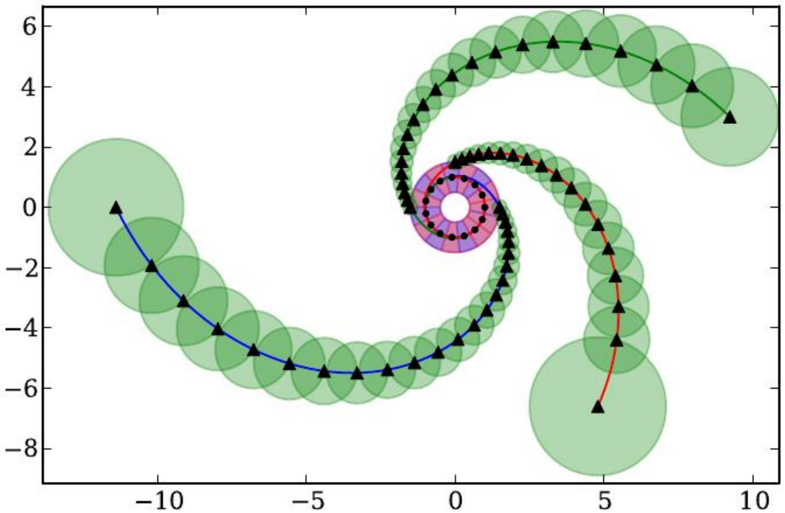
- The shaping term $f(\varphi,r)$ takes a two dimensional input.

The Canonical System - Idea

Two different types of basis functions in the phase plane

- ψ_j “living” outside the limit cycle (transient)
- φ_i “living” on the limit circle (periodic pattern).





Each transient has its separate set of basis functions ψ_j.

The Shaping Term

We extend the perturbation force used by [Gams et al. 2009]

- Using the introduced functions, we set

$$f(\phi, r) = \frac{\sum_{j=1}^{M} \psi_j(\phi, r)\tilde{w}_j + \sum_{i=1}^{N} \varphi_i(\phi, r) w_i}{\sum_{j=1}^{M} \psi_j(\phi, r) + \sum_{i=1}^{N} \varphi_i(\phi, r)}, \qquad (10)$$

- where $W := (w_1, \ldots, w_N, \tilde{w}_1, \ldots, \tilde{w}_M)^T$
  contains the weights which can be adjusted to fit the desired trajectory.
- Solve Eq. (4) for f and use LWR to calculate W.

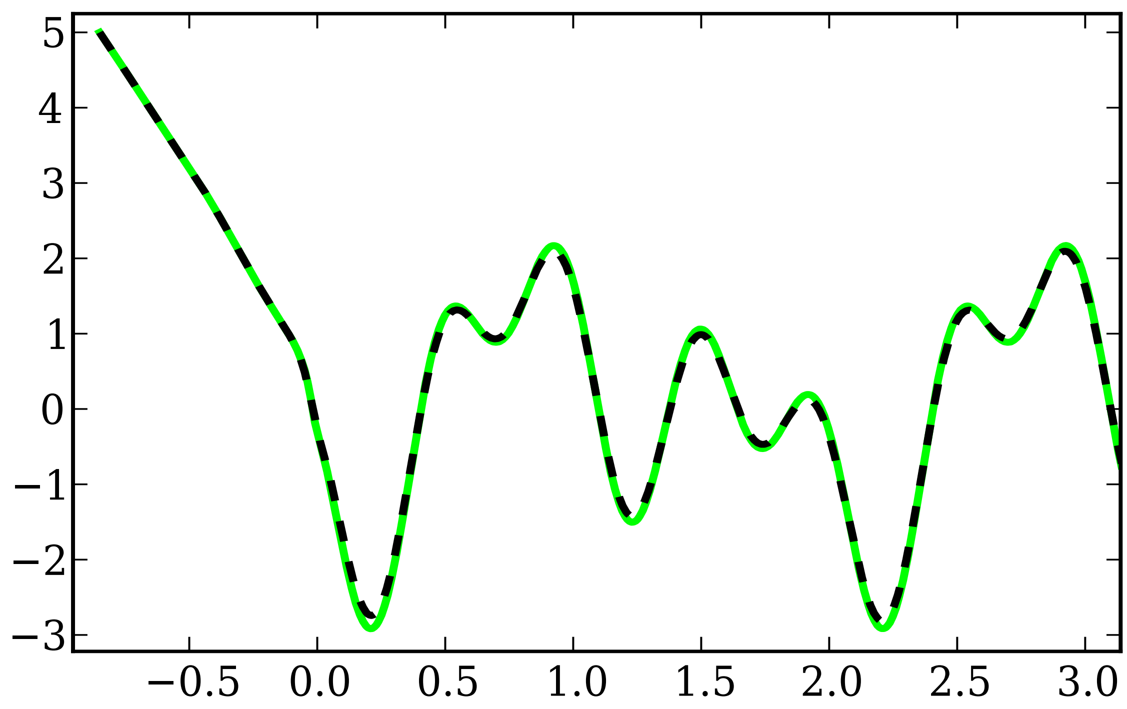
Proof of concept evaluation

- What can we do with that?

Evaluation - One (simple) Transient

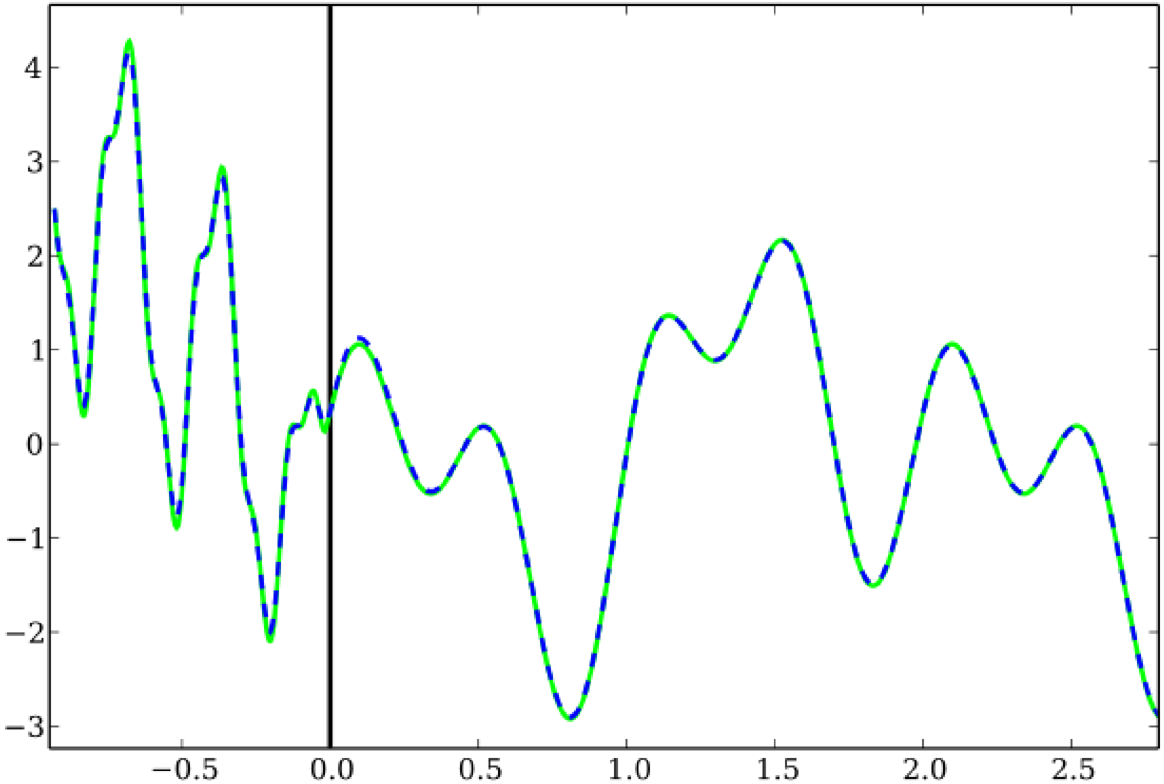
Simple-shaped transient

$$\text{transient}(t) = 5 - 5t$$

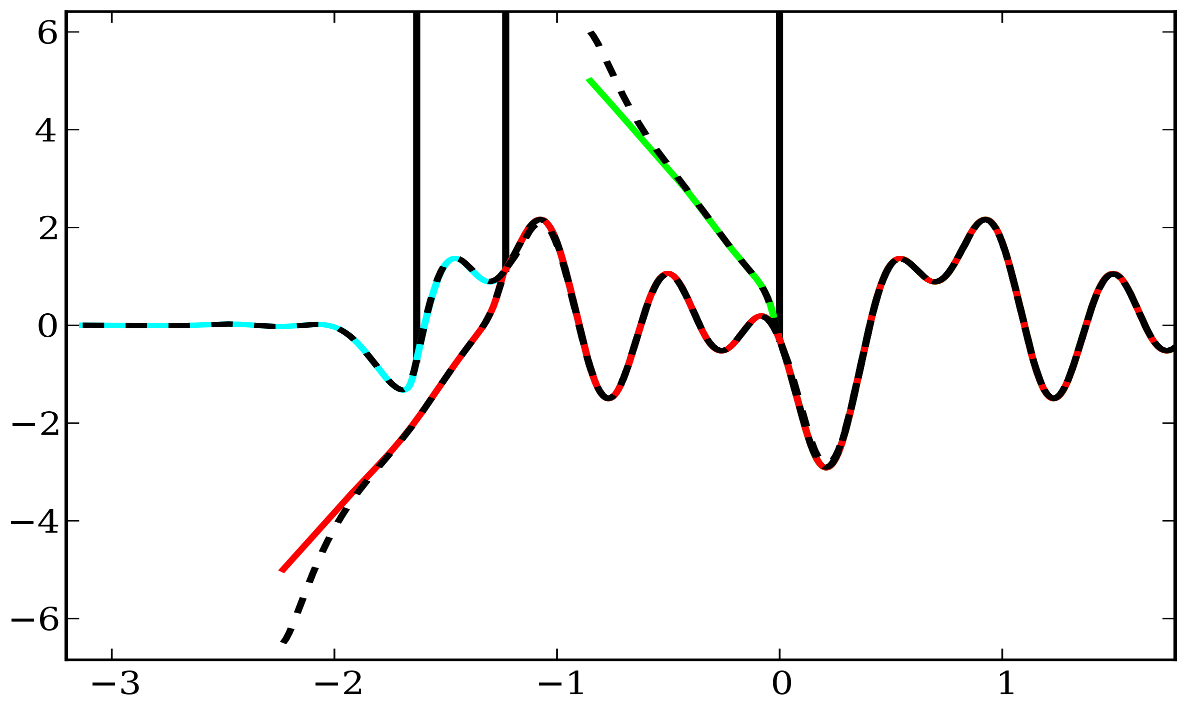


Evaluation - Complex-shaped transient

$$\text{transient}(t) = 3 - 4t - 2\sin(20t) - 0.5\cos(60t)$$



Encoding multiple Transients

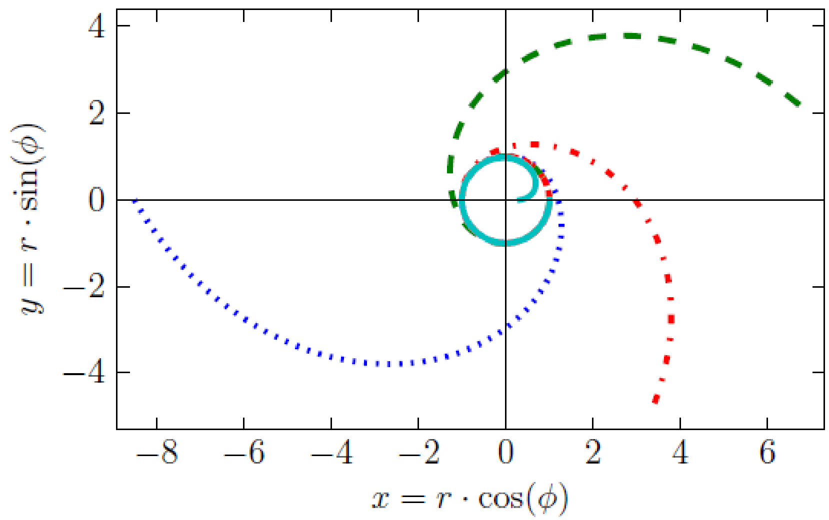


- Three different transients encoded into one DMP
- Generalization to new initial conditions

Summary: Extension of periodic DMP
New canonical system

- Current canonical systems
  - Discrete DMP: distance to end point
  - Periodic DMP: period phase state

- Idea: combine both canonical systems
  - r: distance from the periodic pattern
  - ϕ: phase of the periodic pattern



- **New canonical system**:
  - State $s \coloneqq (\phi, r)^T$ for ϕ, r solution of

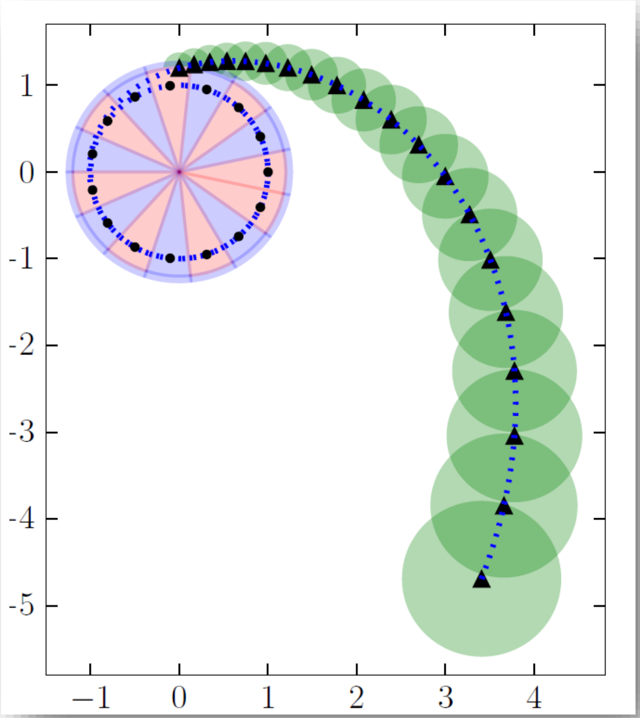
$$\begin{cases} \dot{\phi} = \Omega \\ \dot{r} = \eta(\mu^\alpha - r^\alpha) r^\beta \end{cases},$$

  - With $\mu, \eta, \alpha, \beta > 0$ constant, $\Omega > 0$ phase frequency

Summary: Extension of periodic DMP: perturbation representation

- The perturbations functions needs f to be extended as well

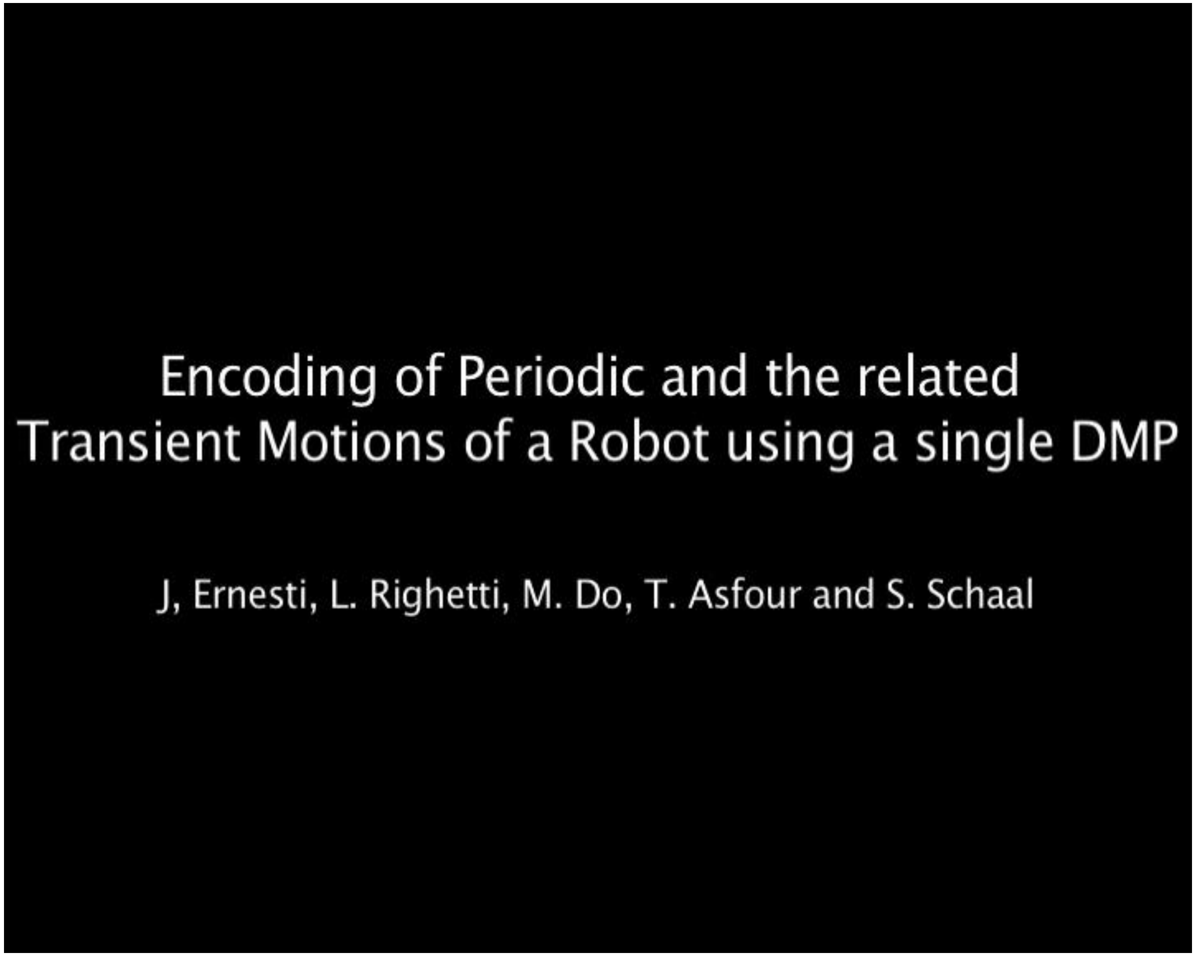
$$f(\phi, r) = \frac{\sum_{i=1}^{N} \tilde{w}_i \psi_i(\phi,r) + \sum_{i=1}^{N} w_i \varphi_i(\phi,r)}{\sum_{i=1}^{N} \psi_i(\phi,r) + \sum_{i=1}^{N} \varphi_i(\phi,r)}$$



- $\psi_i(\phi, r)$ controls the transient part of the motion
- $\varphi_i(\phi, r)$ controls the periodic part of the motion

J. Ernesti, L. Righetti, M. Do, T. Asfour, and S. Schaal (2012) , ***Encoding of Periodic and their Transient Motions by a Single Dynamic Movement Primitive.*** International Conf. on Humanoid Robots (Humanoids 2012), Osaka, Japan,
Best paper award Finalist

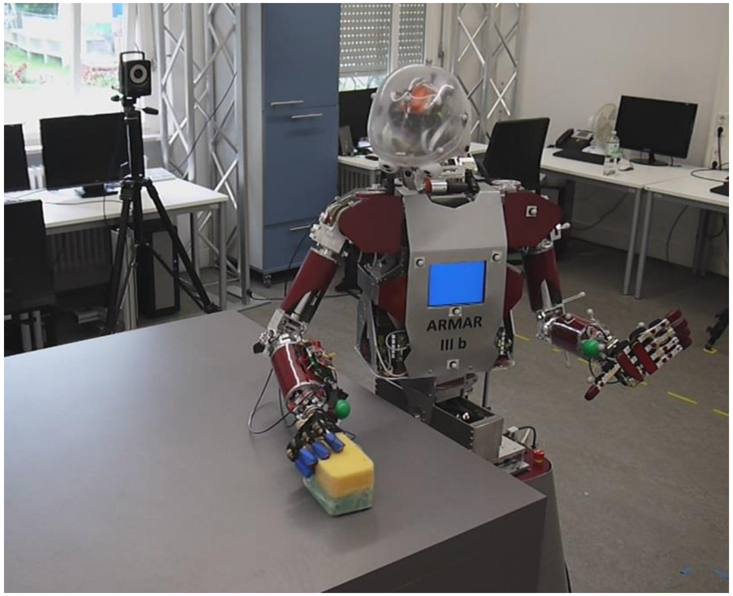
Extending DMPs Periodic DMPs with transients



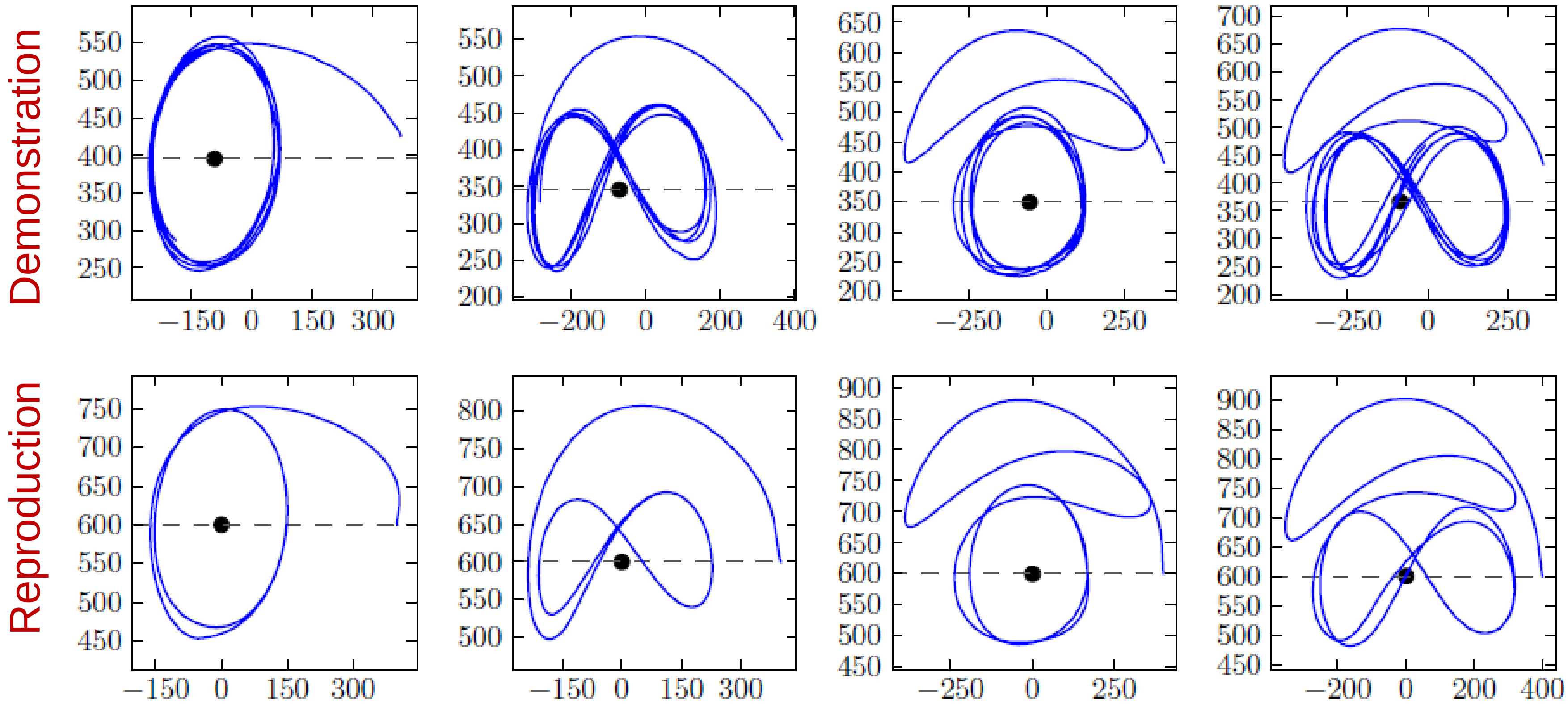
Ernesti, J. and Righetti, L. and Do, M. and Asfour, T. and Schaal, S. (2012). Encoding of Periodic and their Transient Motions by a Single Dynamic Movement Primitive, Humanoids 2012 **- Best paper award Finalist -**

Evaluation (I)

- We implemented a wiping task on the humanoid robot ARMAR
- Four different wiping styles with simple and complex transients
- Motion capturing using Vicon

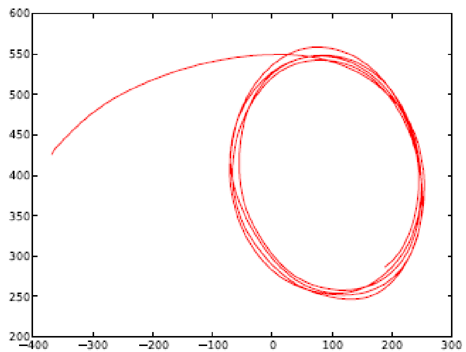


Evaluation (II)

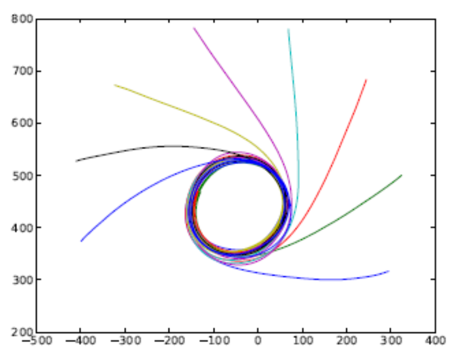


Generalization with multiple transients

- Training with one transient



- Training with multiple transients



Literature

- Auke Jan Ijspeert, Jun Nakanishi, Heiko Hoffmann, Peter Pastor, and Stefan Schaal. *Dynamical Movement Primitives: Learning Attractor Models for Motor Behaviors,* Neural Computation, February 2013, Vol. 25, No. 2 , Pages 328-373

- J. Ernesti, L. Righetti, M. Do, T. Asfour and S. Schaal, *Encoding of Periodic and their Transient Motions by a Single Dynamic Movement Primitive*, IEEE/RAS International Conference on Humanoid Robots (Humanoids), pp. 57 - 64, December, 2012

- P. Pastor, H. Hoffmann, T. Asfour and S. Schaal, *Learning and Generalization of Motor Skills by Learning from Demonstration*, Proceedings of the IEEE International Conference on Robotics and Automation, 2009