Humanoid Robots: Design Issues and Control

International UJI Robotics School on Humanoid Robots
Benicasim, Castelló de la Plana, Spain

18-22 September 2006

Tamim Asfour
Universität Karlsruhe

Institut für Technische Informatik (ITEC)
Lehrstuhl für Industrielle Anwendungen der Informatik und Mikrosystemtechnik (IAIM)

http://wwwiaim.ira.uka.de/

Forschungszentrum Informatik Karlsruhe (FZI)
Interaktive Diagnose und Servicesysteme (IDS)

http://www.fzi.de/ids

Humanoid Robots

- Complex Challenge
  - mechatronics,
  - Motor Control,
  - Perception,
  - human-machine interaction
  - Learning theory,
  - computational neuroscience
  - biomechanics

- Design of humanoid robots requires coordinated and integrated research efforts of several disciplines
- The integration of several disciplines for the building of humanoid robots requires enormous collaborative resources

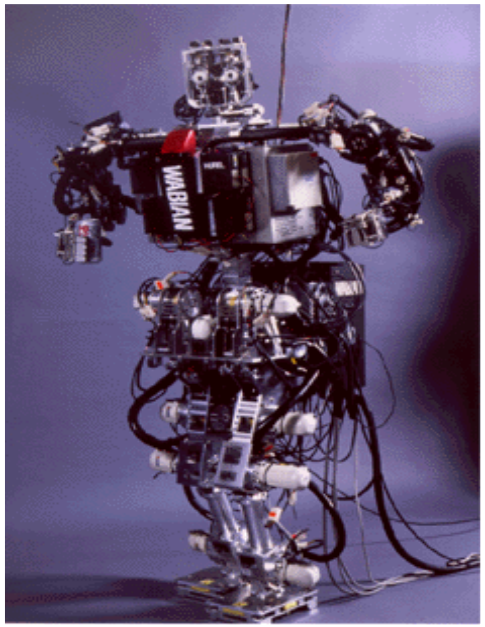
Humanoid robots and humanoid projects

Honda Robots, Japan

http://world.honda.com/ASIMO/history/history.html

Humanoid robots and humanoid projects

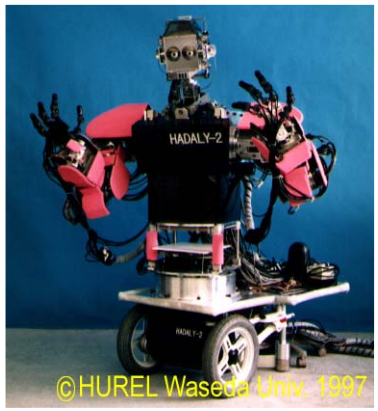
Waseda University, Tokyo, Japan



WABIAN

Wabian-2

Hadaly-2



Humanoid robots and humanoid projects

HRP-2, Kawada Industries, Japan

Walking

Assemblying a panel

Dancing

Standing up

Humanoid robots and humanoid projects

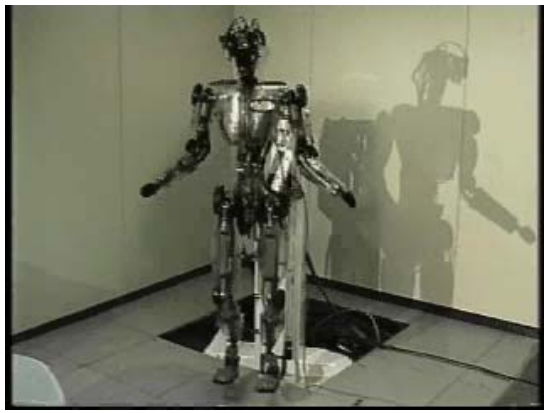
Sony's Humanoid Robots, Japan

SDR-3X

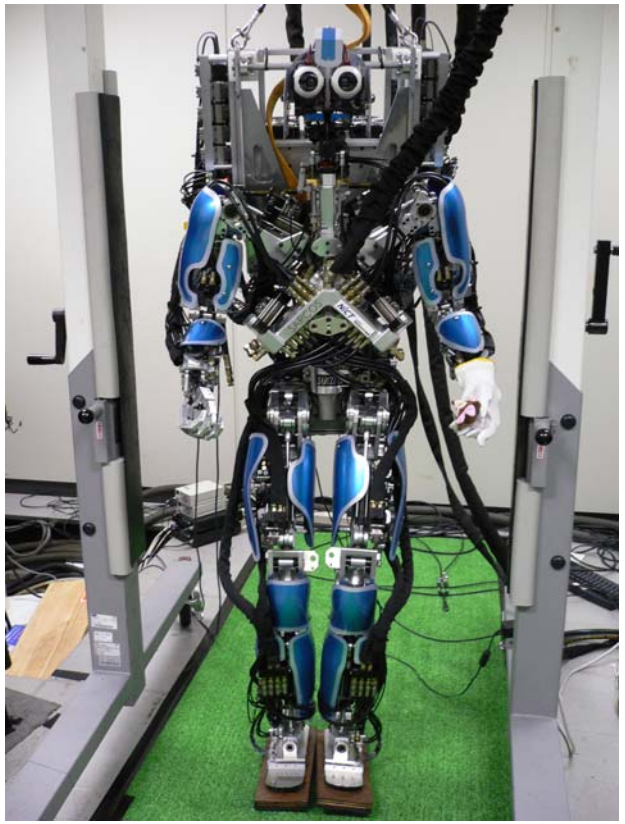
QRIO

Humanoid robots and humanoid projects

Sarcos Robots at ATR



DB (Dynamic Brain)

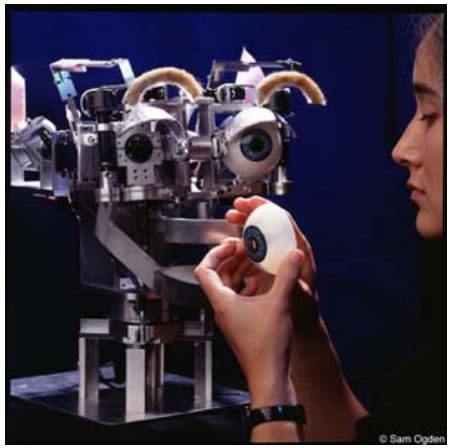


CB (Computational Brain)

Humanoid robots and humanoid projects

USA

Cog, MIT

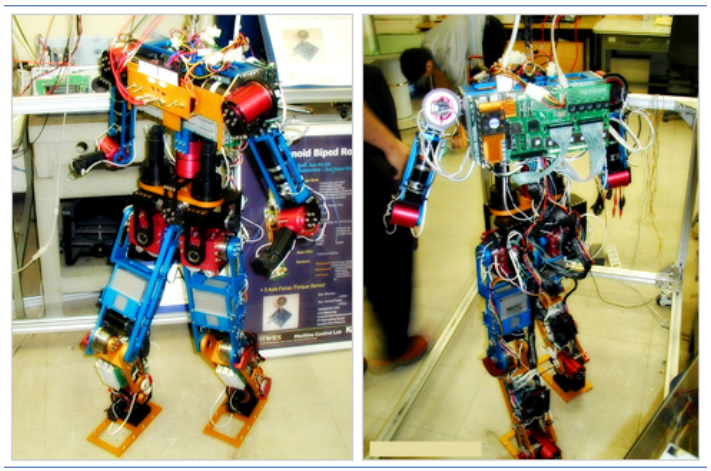


Kismet, MIT

Robonaut,
NASA's Johnson
Space Center

Humanoid robots and humanoid projects

Korea Institute of Science and Technology

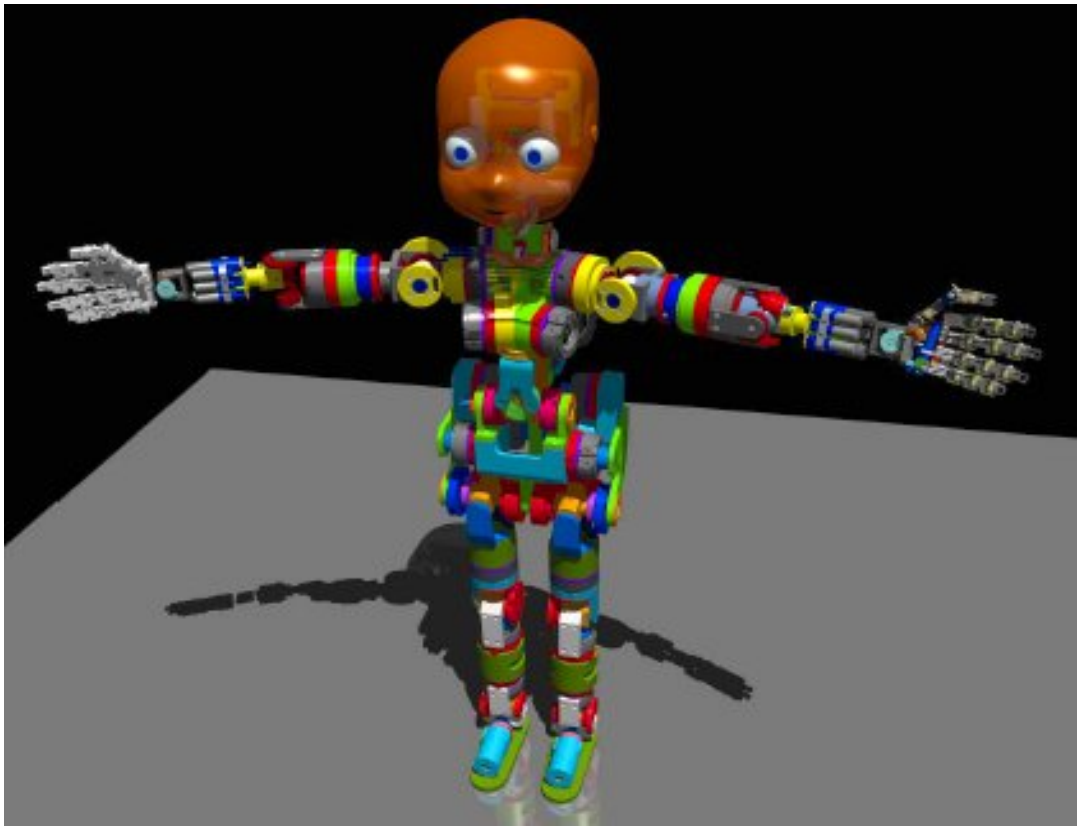


KHR-1
KAIST Humanoid Robot platform – 1

KHR-2
KAIST Humanoid Robot platform – 2

Humanoid robots and humanoid projects

Rh-1
UC3M Humanoid Universidad
Carlos III, Madrid, Spain

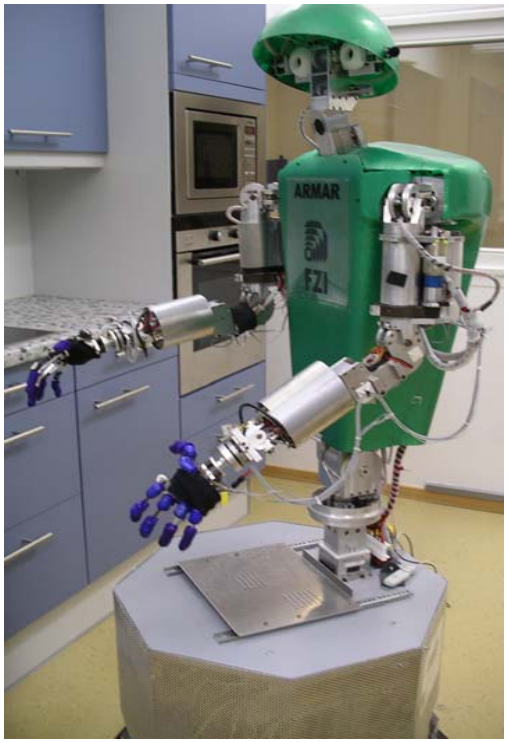


iCUB
EU-project RobotCub, Genoa, Italy

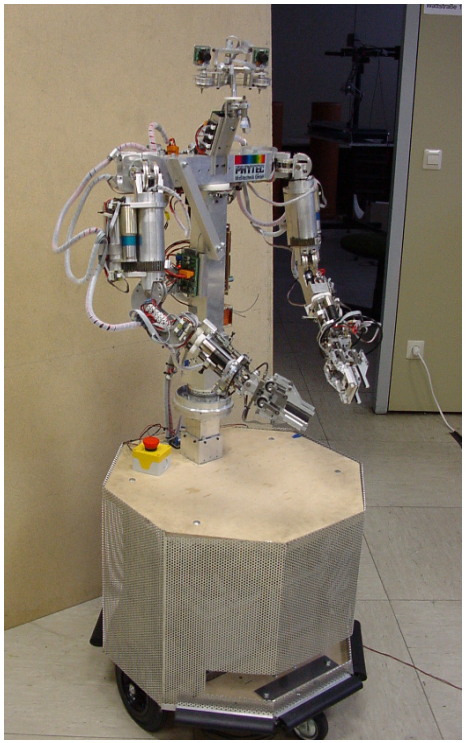
Humanoid robots and humanoid projects

Karlsruhe, Germany, SFB 588

ARMAR-III, 2006



ARMAR-II, 2002



ARMAR, 2000

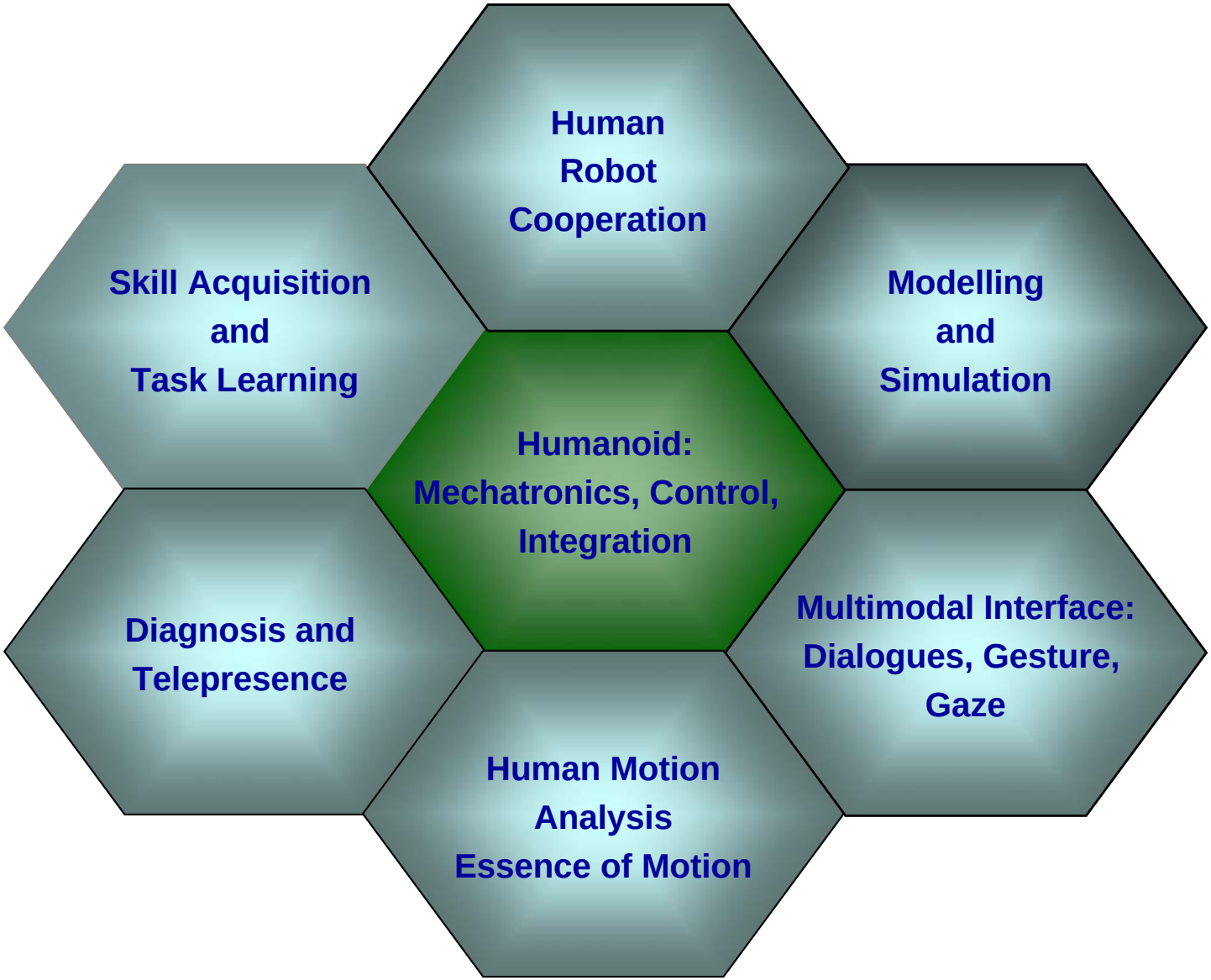
KAWAI-II, 2005

Humanoid Robots in Karlsruhe

Collaborative Research Center on Learning and Cooperating Multimodal Humanoid Robots (SFB 588)

- Goal: Methods for building humanoid robots for household environments (kitchen)
- Established by the German Research Foundation (DFG) in 2001; intended to run for 12 years
- Interdisciplinary research project
  - Universität Karlsruhe
  - Research Center Computer Science (FZI)
  - Research Center Karlsruhe
  - Fraunhofer Institute Karlsruhe (IITB)
- http://www.sfb588.uni-karlsruhe.de/

Project Research Topics



Outline

- Machatronics and control of the humanoid robots ARMAR I-III
  - Kinematics modelling
  - Mechanical design (arm and head)
- Control architecture
- Hardware and software architecture
- Programming of manipulation tasks
  - Inverse kinematics
  - Human-like motions
- Framework for task coordination

Building humanoids robots

- What needs to be done in building a humanoid robot?
- **Specification:**
  - Degrees of freedom for head, arms, hands, torso, legs ...
  - Sensors and actors?
  - Hardware components (PC's, microcontroller, DSP, ...)
  - Operating system (real-time)
  - Software development tools
- **Models:**
  - kinematics and dynamics models
  - models of the environment
  - ...
- **Control Architecture**

Kinematics criteria for humanoid arms

- Anthropomorphic Design
- Workspace
  - comparable to that of the human arm
  - Should enhance the ability to avoid obstacles
- Kinematics simplicity
  - easy to solve the inverse kinematics problem
  - reduces the complexity of path planning
  - singularity-free kinematics
- Machnical constructability
  - feasible joint reaslisation
  - placement of sensors, electronics
  - wiring and cabling
  - ...

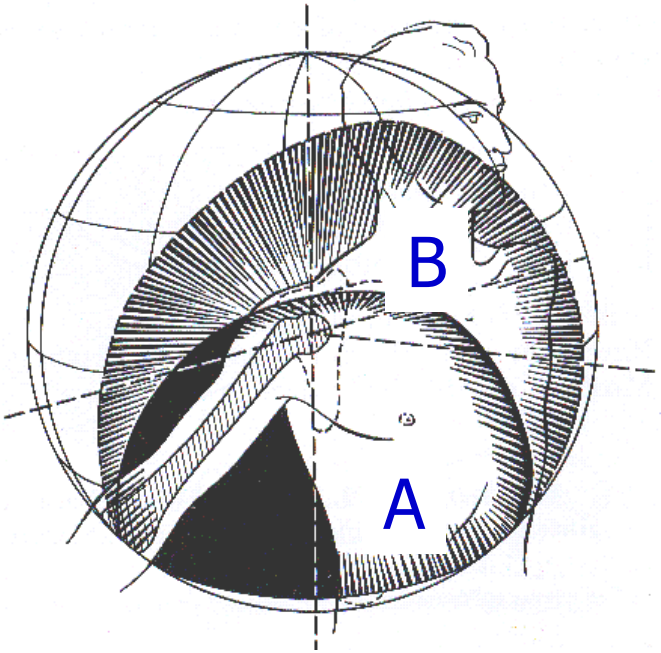
Building humanoids robots: Arms

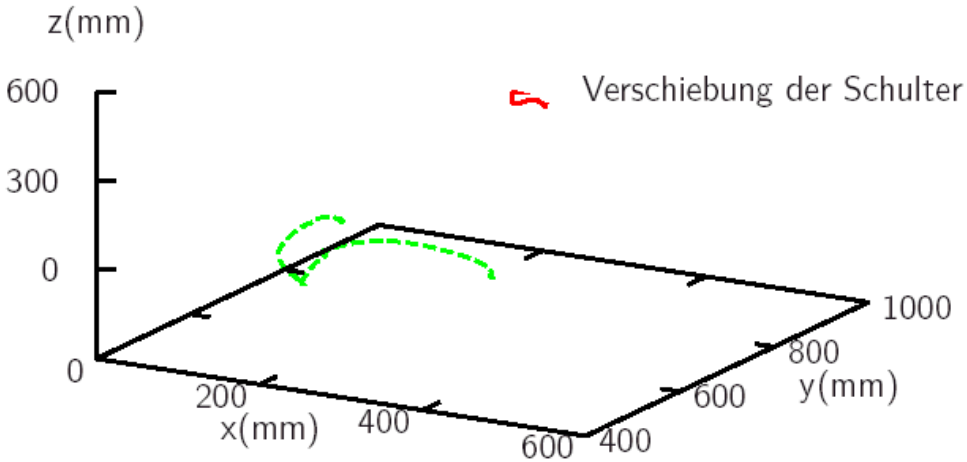
- Biomechanics und Medicine:
  [Inman 44], [Högfers 87], [Engin 89], [van der Helm 95], [Youm 79], [Gonzalez 96], [An 81], [Andrew 79]
- Computer graphics:
  [Thalmann 94], [Schmitt 95], [Maurel 98]

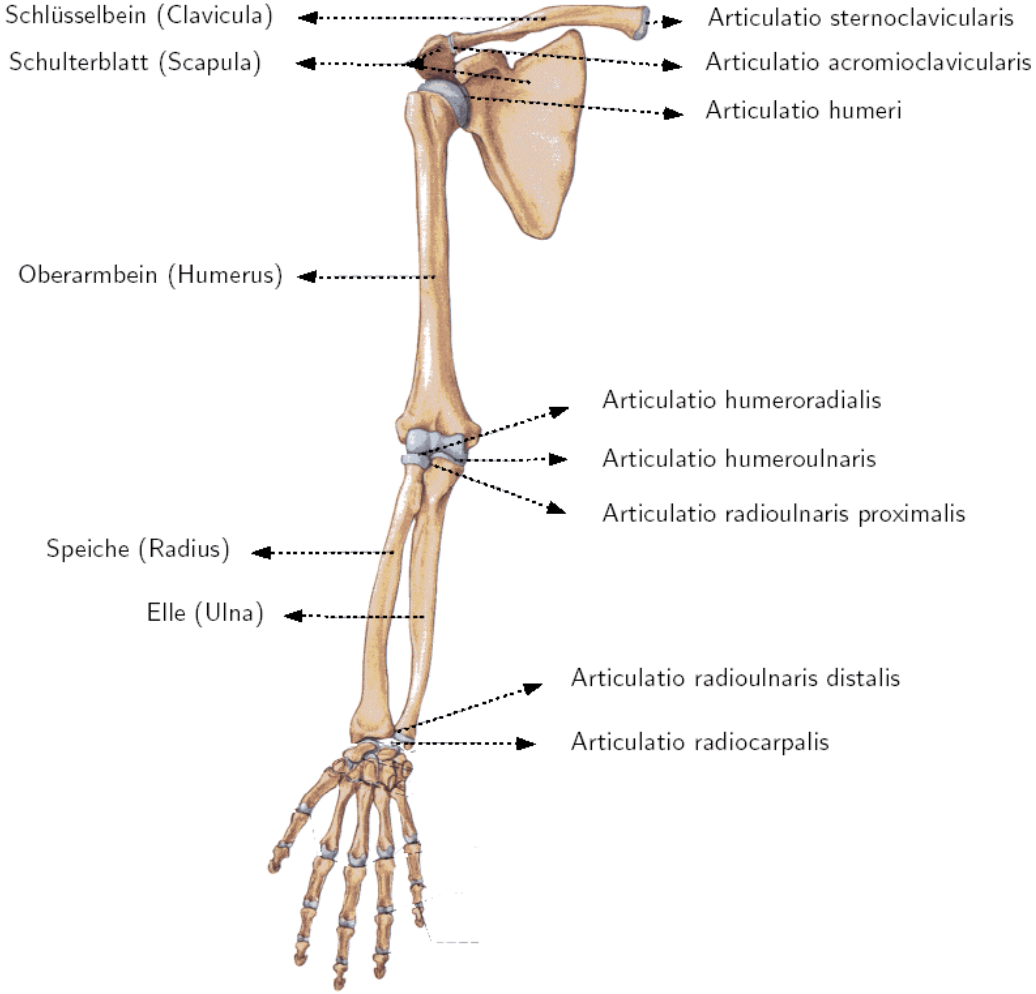
  Technical Realisation of the joint complexes in the human shoulder is impossible

- Robotic:
  [Lenarcic 94], [Lenarcic 00]

Anatomy of the human arm system

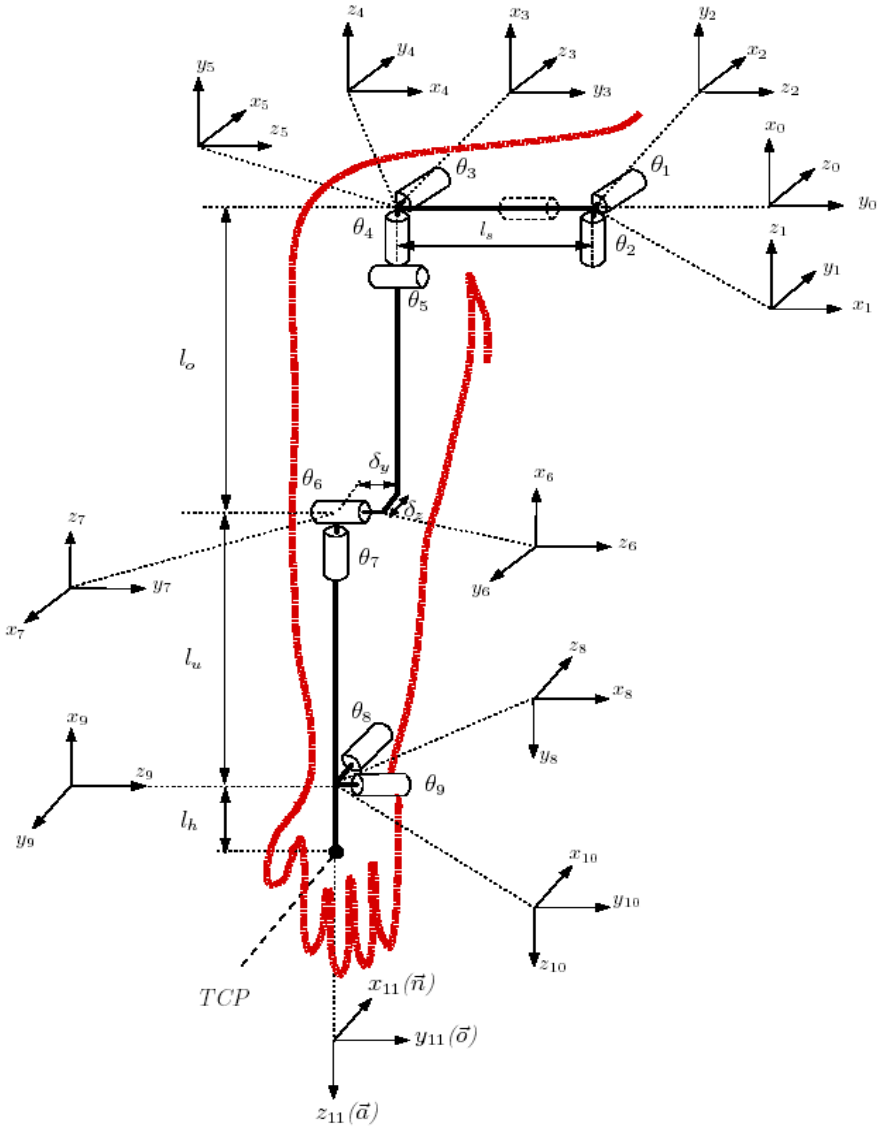




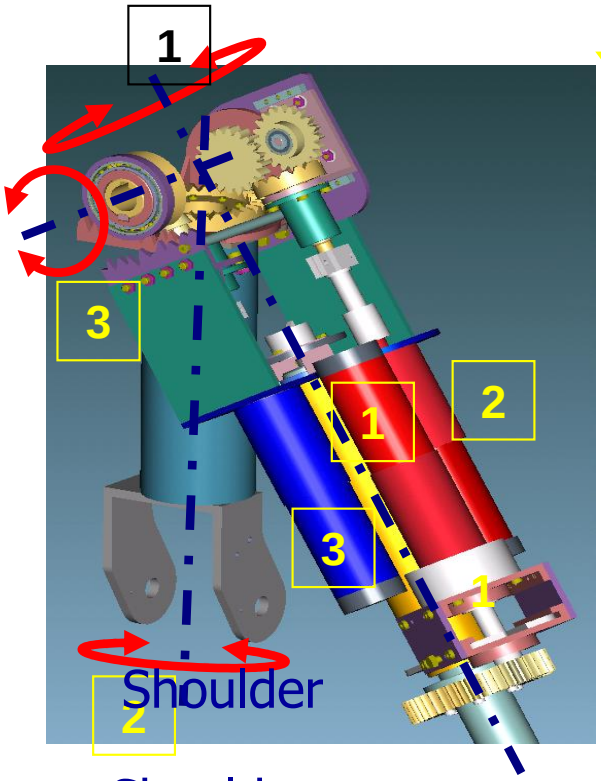


Simple model of the human arm system

- To achieve realistic shoulder motion
  - inner shoulder joint: 2 DOF
  - outer shoulder joint: 3 DOF
- Elbow: 2 DOF
- Wrist: 2 DOF
- ➔ **kinematics chain with 9 DOF**

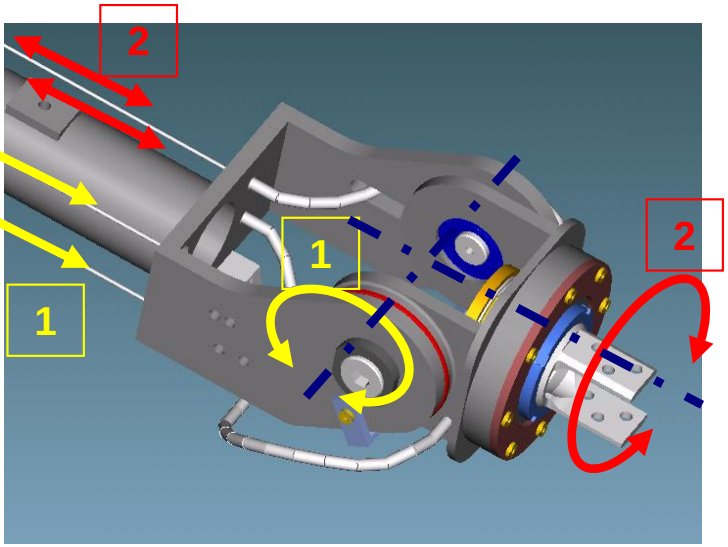


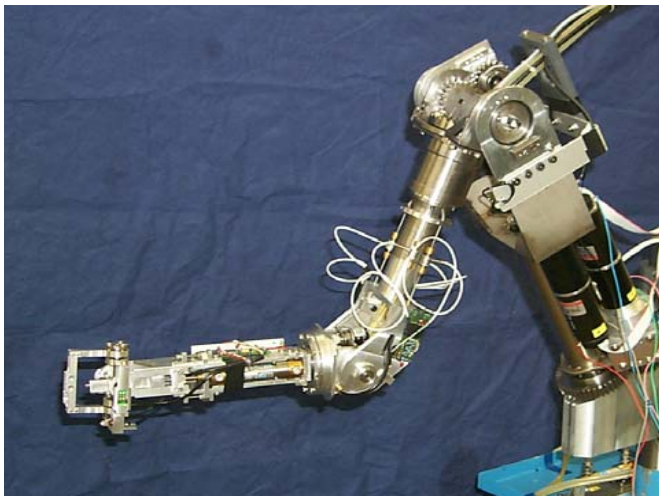
Realisation of a 7 DOF arm (ARMAR-III)

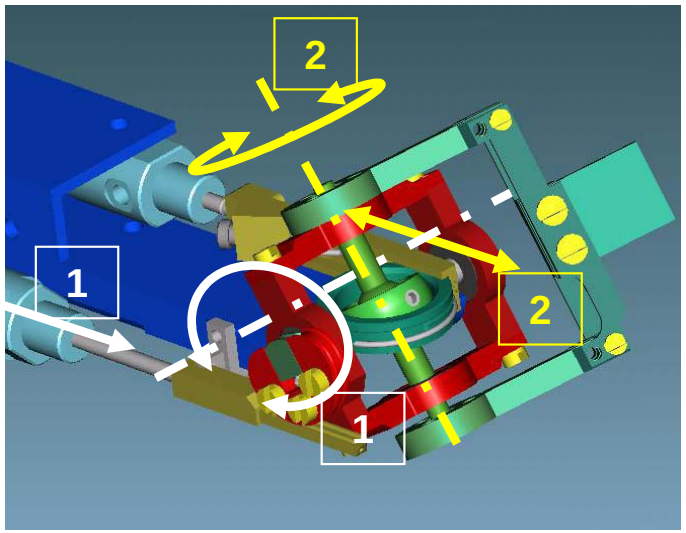


Shoulder

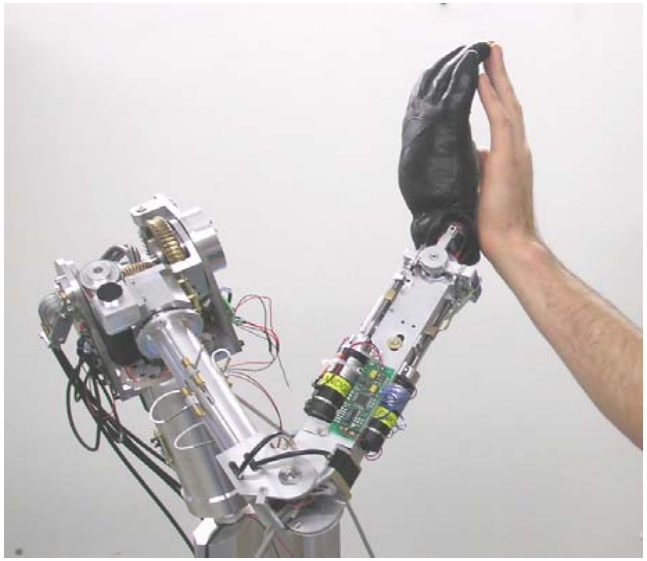
Elbow







Wrist



Sven Brudniok et al.
http://www.ipek.uni-karlsruhe.de/

Realisation of a 7 DOF arm (ARMAR-III)

- Weight: 5 kg, Max. Payload: 3 kg
- Motors/Gear units:
  - Shoulder: Harmonic Drives units
  - Elbow/Forearm: linkage system
  - Wrist: Faulhaber motor/gear units
- Sensors: each joint
  - Encoder attached to the axis of the motor
  - Optical encoder in the axis position
  - Joint torque sensor
    (Novatech Measurements Ltd., www.novatechuk.demon.co.uk)
- Wrist sensor: 6 DOF Force/Torque sensor (ATI)
- Artificial skin for advanced human-robot cooperation developed at IPR, Karlsruhe (wwwipr.ira.uka.de) and Weiss robotics

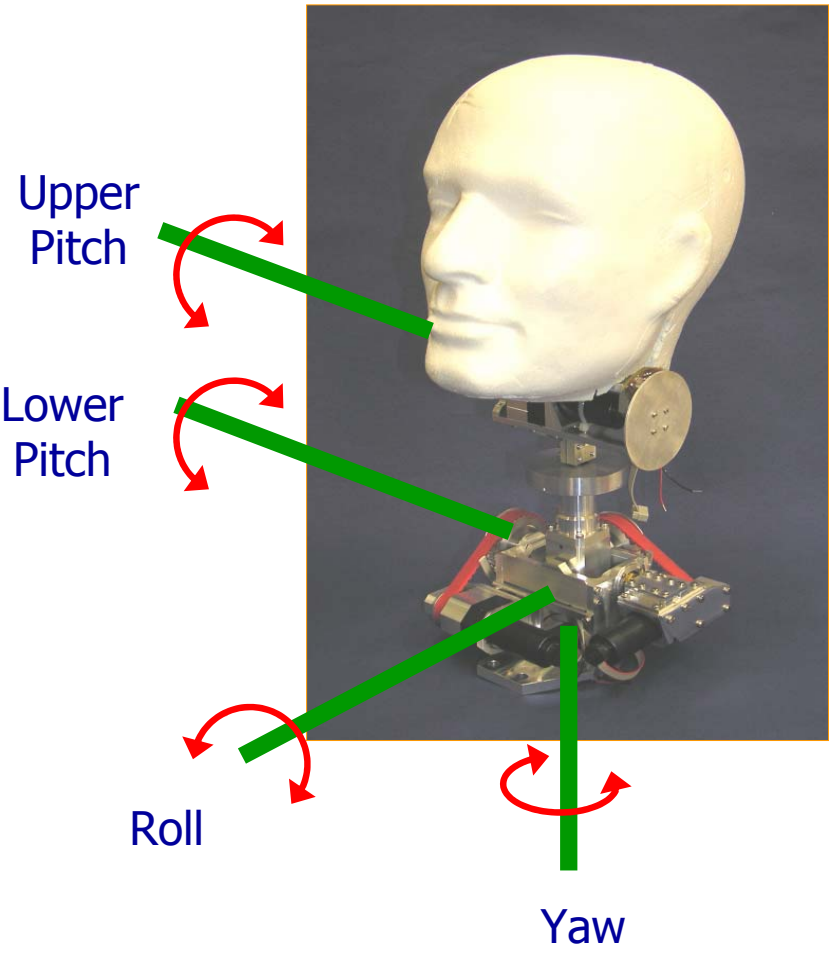
Mechanical Design: Humanoid Head

- The head should provide rich perceptual input necessary to realize various visuo-motor behaviours, e.g.
  - smooth pursuit and saccadic movements towards salient regions,
  - sensory-motor tasks such as hand-eye coordination, gesture identification, human motion perception, ...

- Major head design criteria:
  - Realistic human size and shape while modelling the major degrees of freedom (DoFs) in the human neck/eye system
  - Feature humanlike characteristics in motion and response
  - Saccadic motions and smooth pursuit over a wide range of velocities.
  - The optics should mimic the structure of the human eye, which has a higher resolution in the fovea.
  - Easy to construct, easy to maintain and easy to control.
  - The acoustic system should allow for sound source 3D-localisation

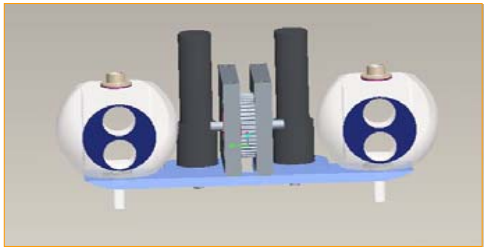
Mechanical Design: Humanoid Head

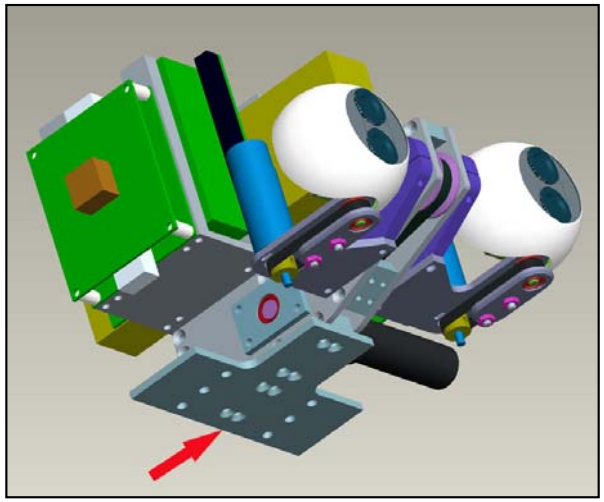
- Neck mechanism with 4 DOFs
  - Pitch-Roll-Yaw-Pitch
  - Harmonic Drive Motor/Gear units
  - www.ipek.uni-karlsruhe.de/
- Realisation of the visual system
  - Current technology does not allow to exactly mimic the features of the human visual system
  - Camera systems that provide both peripheral and foveal vision from a single camera are still experimental



Mechanical Design: Humanoid Head

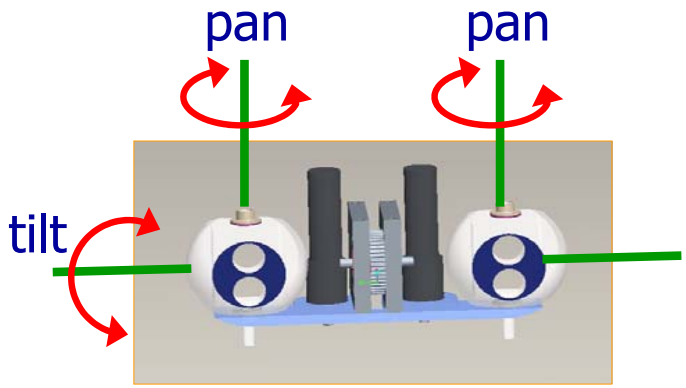
- Alternative which allows to use commercially available camera systems that are less expensive and more reliable
  - two cameras per eye, one with wide-angle lens for peripheral vision and one with narrow-angle lens for foveal vision
  - Point Grey Research Dragonfly IEEE-1394 camera in the extended version (www.ptgrey.com).
  - Extended version allows the CCD to be up to 6 inches away from the camera interface board

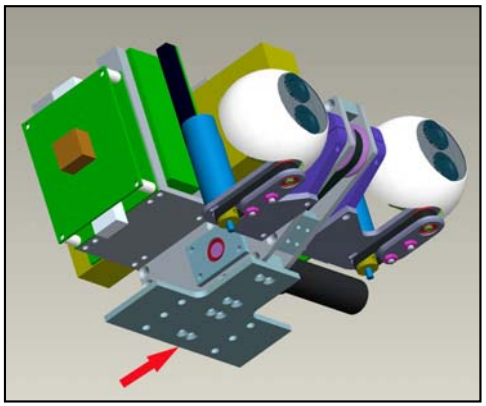


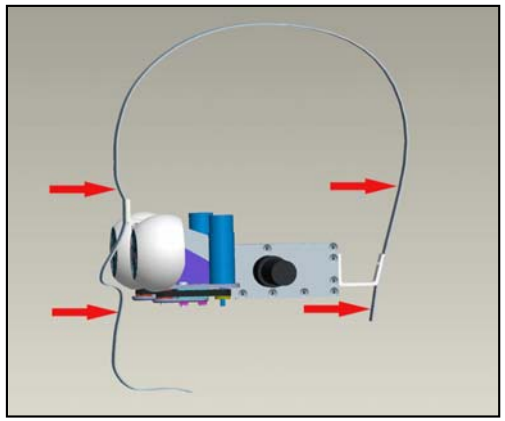


Mechanical Design: Humanoid Head

- Two cameras per eye
- Point Grey Research Dragonfly IEEE-1394 camera in the extended version
- Human like eye movements → Each eye can independently rotate about a vertical axis (pan), and the two eyes share a horizontal axis (tilt)
- 3D acoustic localisation → 6 microphones (SONY ECMC115.CE7)
  - two in the ears, two in the front and two in the rear of the head.





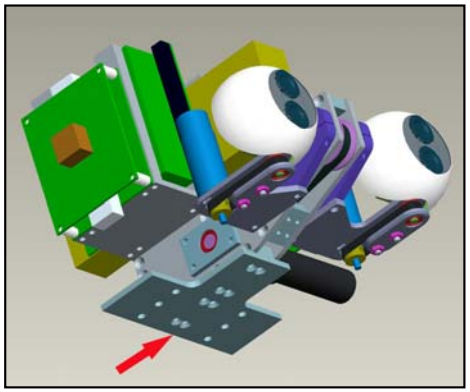
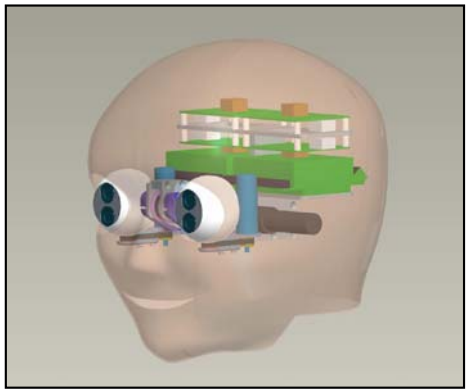
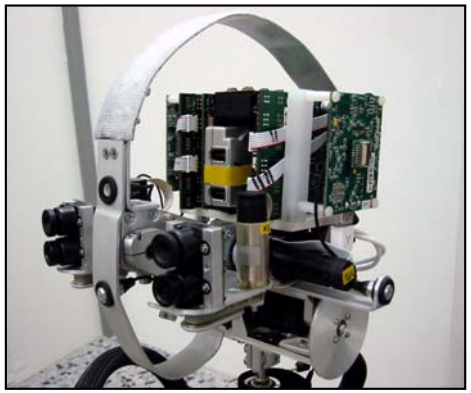


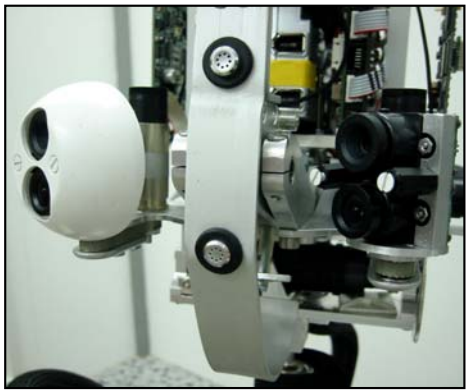
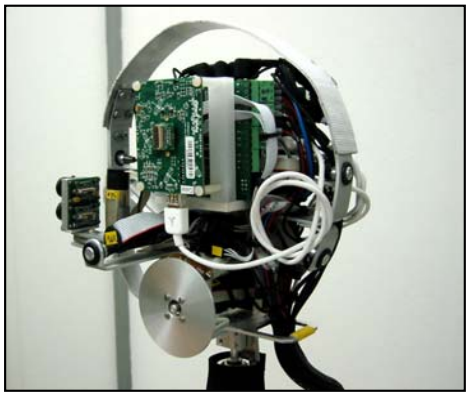
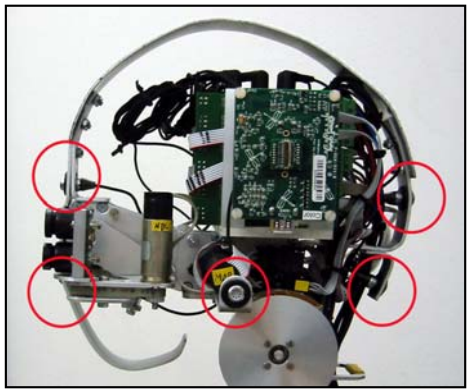
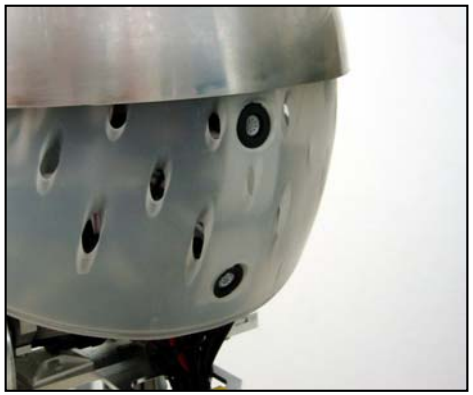
Mechanical Design: Head

- Pan joints
  - Motor/gear units (Faulhaber)
  - Motor 1524 024 SR (2,5 mNm, 24V)
  - gear ratio=76:1
  - Encoder IE2-512

- Tilt joint:
  - Harmonic-Drive unit PMA-5A-50
  - 50:1 gear ratio
  - torque 0,47 Nm

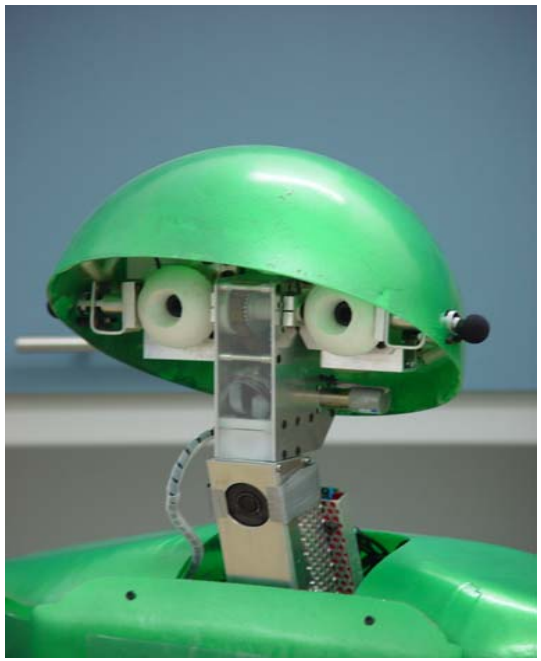
Mechanical Design: Head

Tamim Asfour
Universität Karlsruhe

Heads: ARMAR-II and ARMAR-III

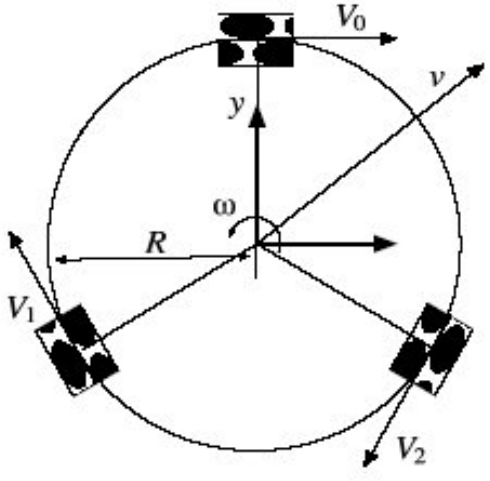


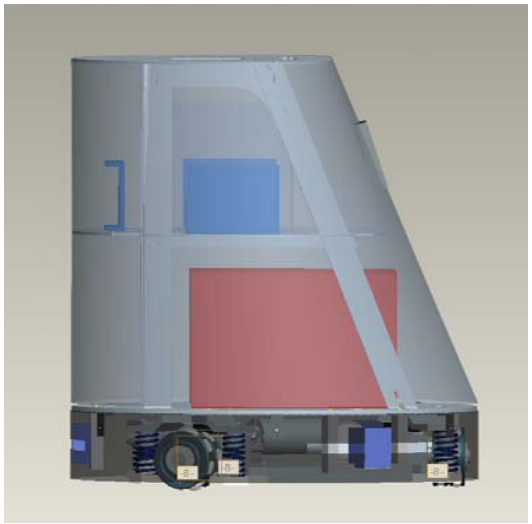
- **ARMAR II**
  - 2 Dragonfly color cameras, 640x480 (30 Hz), Firewire, 6 mm lenses
  - Pan/Tilt (2 DOF)
  - Fixed Stereo calibration
  - 2 microphones

- **ARMAR III**
  - 4 Dragonfly Color cameras, 640x480 (30 Hz), Firewire, 2 mm und 8 mm micro lenses
  - Pan/Pitch (3 DOF)
  - 4 DOF Neck
  - Variable stereo calibration
  - 6 microphones

Mechanical Design: Holonomic Platform

- Application area: Kitchen
- holonomous flexibility using wheels with passive rolls at the circumference (Mecanum wheels or Omniwheels)
- Kinematics [Borenstein 96]
  - J. Borenstein, H. R. Everett, and L. Feng, *Where am I? Sensors and Methods for Mobile Robot Positioning*. Ann Arbor, Michigan, USA, University of Michigan, Department of Mechanical Engineering and Applied Mechanics, 1996
- Platform specification:
  - Maximum height: 700 mm
  - Weight of the upper body: 30 kg
  - Speed minimum: 1 m/s
  - Holonomous drive
  - Power supply (whole system): 8 h with 25% drive
  - Spring-damper combination to reduce vibrations



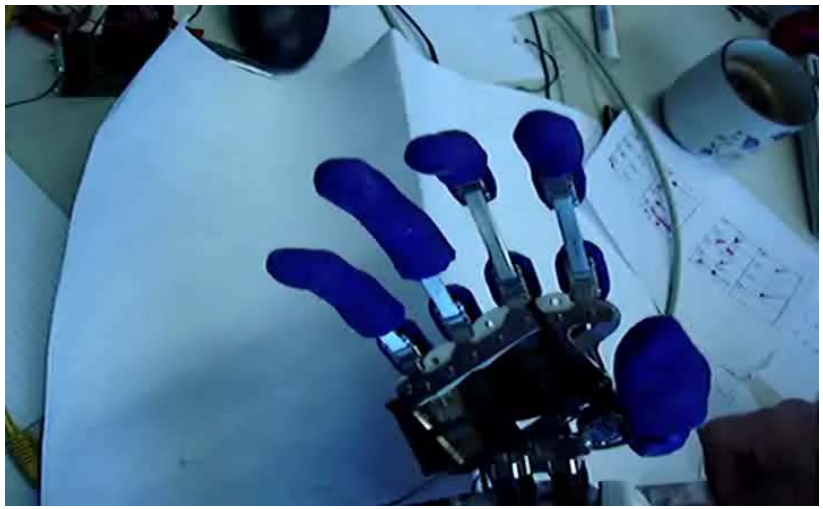


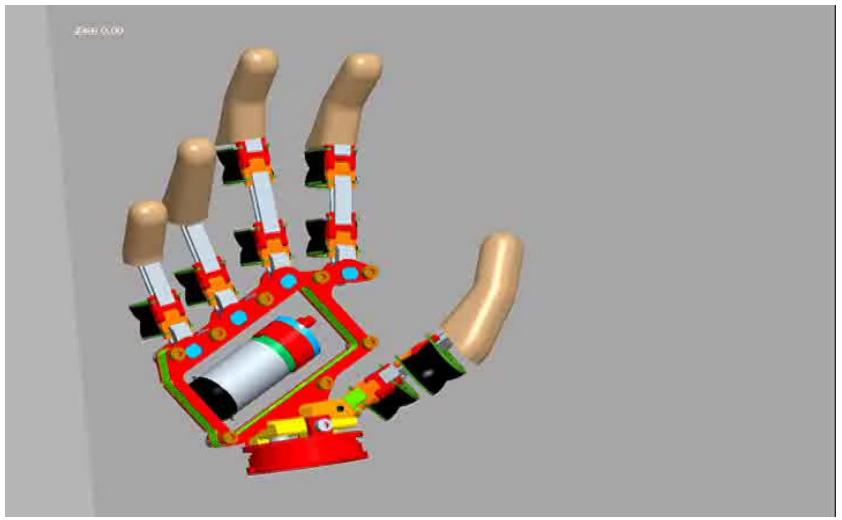
Mechatronics: 5-Finger Hands

Antropomorphic

5-finger-lightweight hand

- Controller „On-board“
- Fluidic/pneumatic actuators
- Anthropomorphic shape and motion characteristics
- Precision and power grasps
- Modular tactile sensor concept with finger tip sensors and joint sensors
- Developed by Dr. Stefan Schulz at FZK (www.fzk.de)

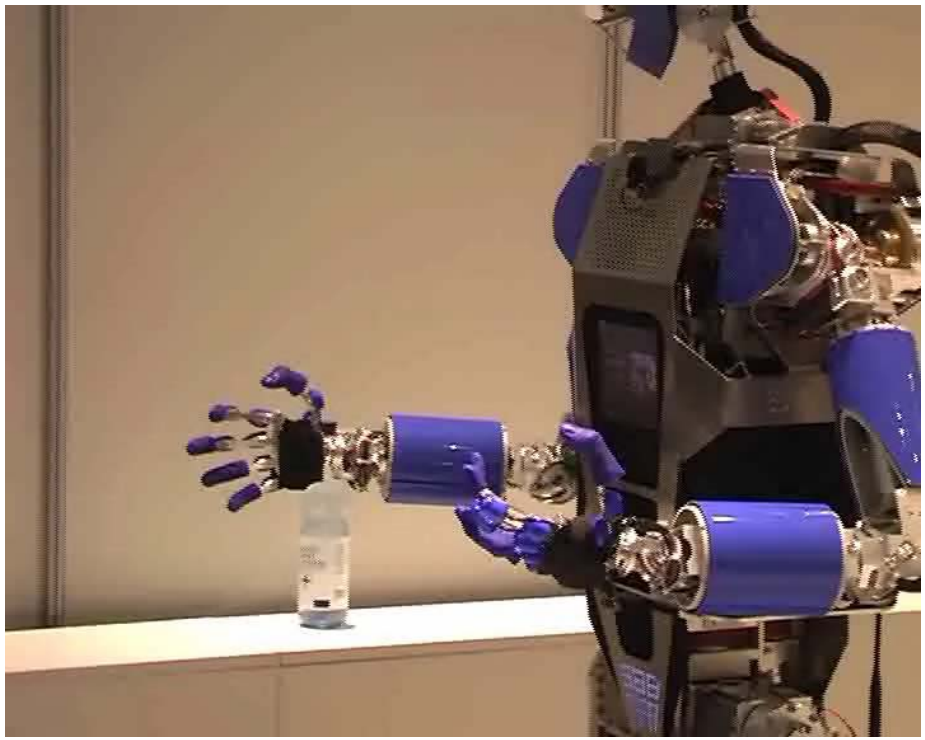




Mechatronics: 5-Finger Hands

Antropomorphic

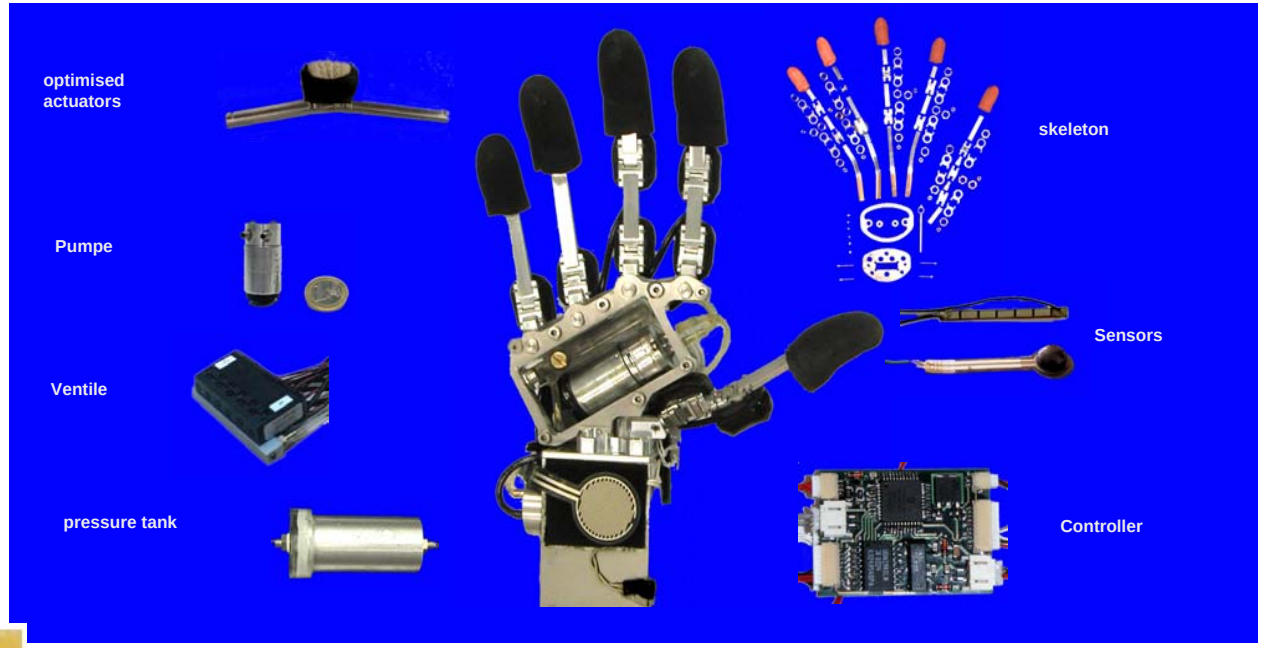
5-finger-lightweight hand



- Controller „On-board“
- Fluidic/pneumatic actuators
- Anthropomorphic shape and motion characteristics
- Precision and power grasps
- Modular tactile sensor concept with finger tip sensors and joint sensors
- Developed by Dr. Stefan Schulz at FZK (www.fzk.de)

Hand with artificial skin (data glove)

- 8 DOF (up to 18)
- Weight: 395 g
- Force:
  - Finger tip: 7 N
  - Holding force: 35 N
- RS 232/Bluetooth



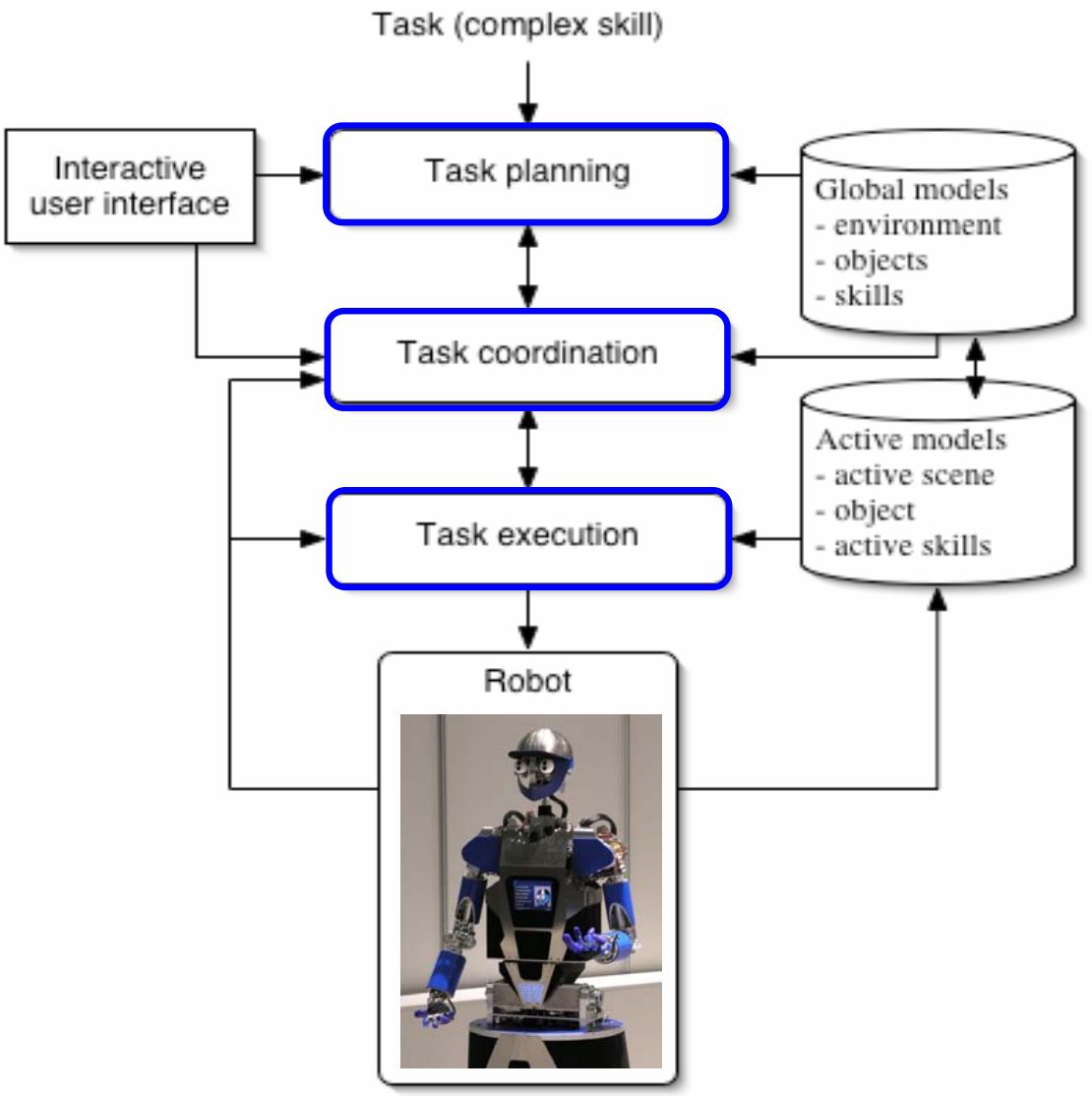
- Dataglove with 8 or 14 tactile sensors
- Bluetooth or serial Interface

Control Architecture

3-Level-Architecture:

- Task planning:
  - task organisation
  - synchronization information
- Task coordination:
  - primitive coordinated actions
  - Condition/Event Petri Nets
- Task execution:
  - sensory motor control

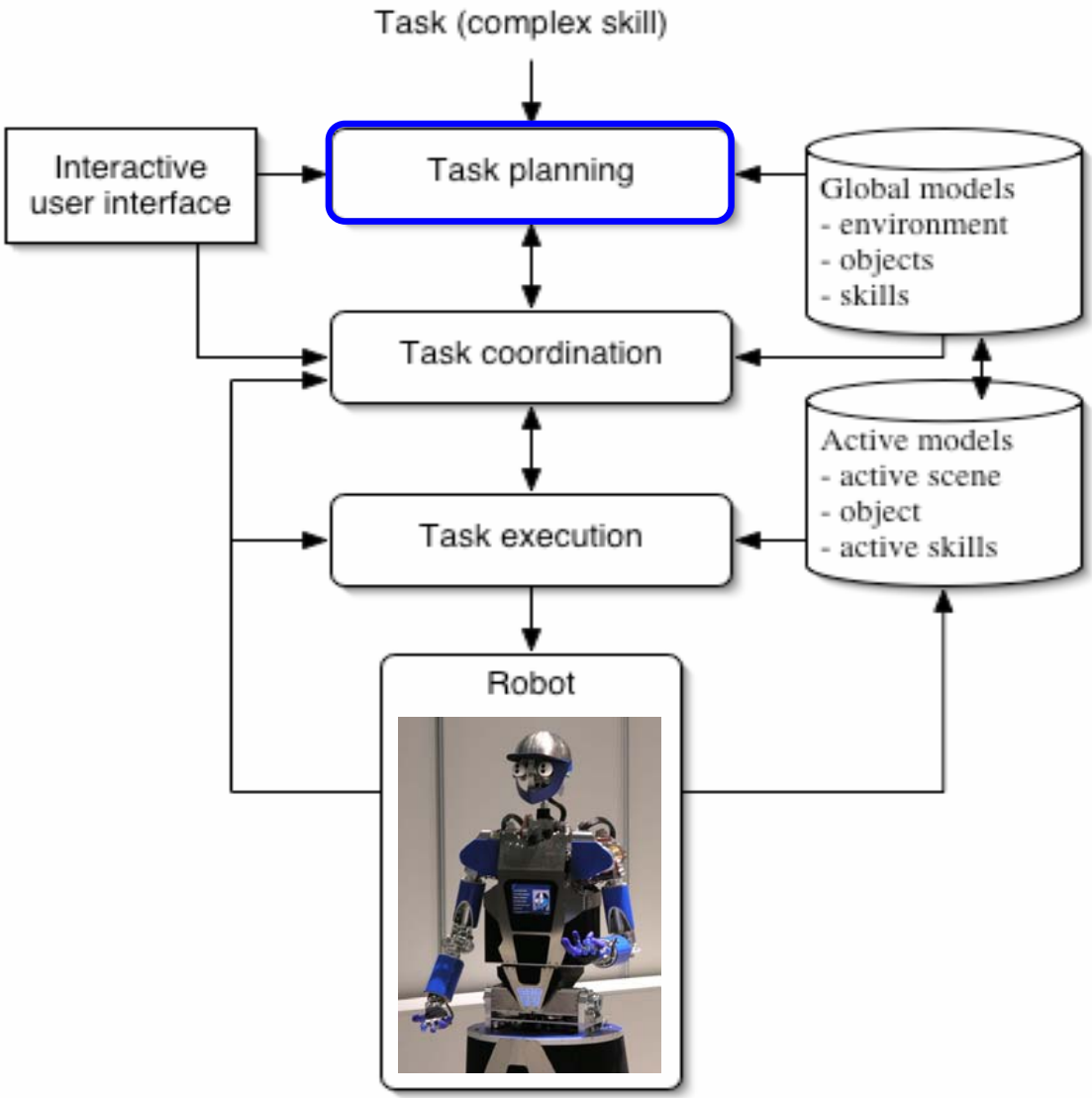
- Reference: [ISER 2004]



Control Architecture

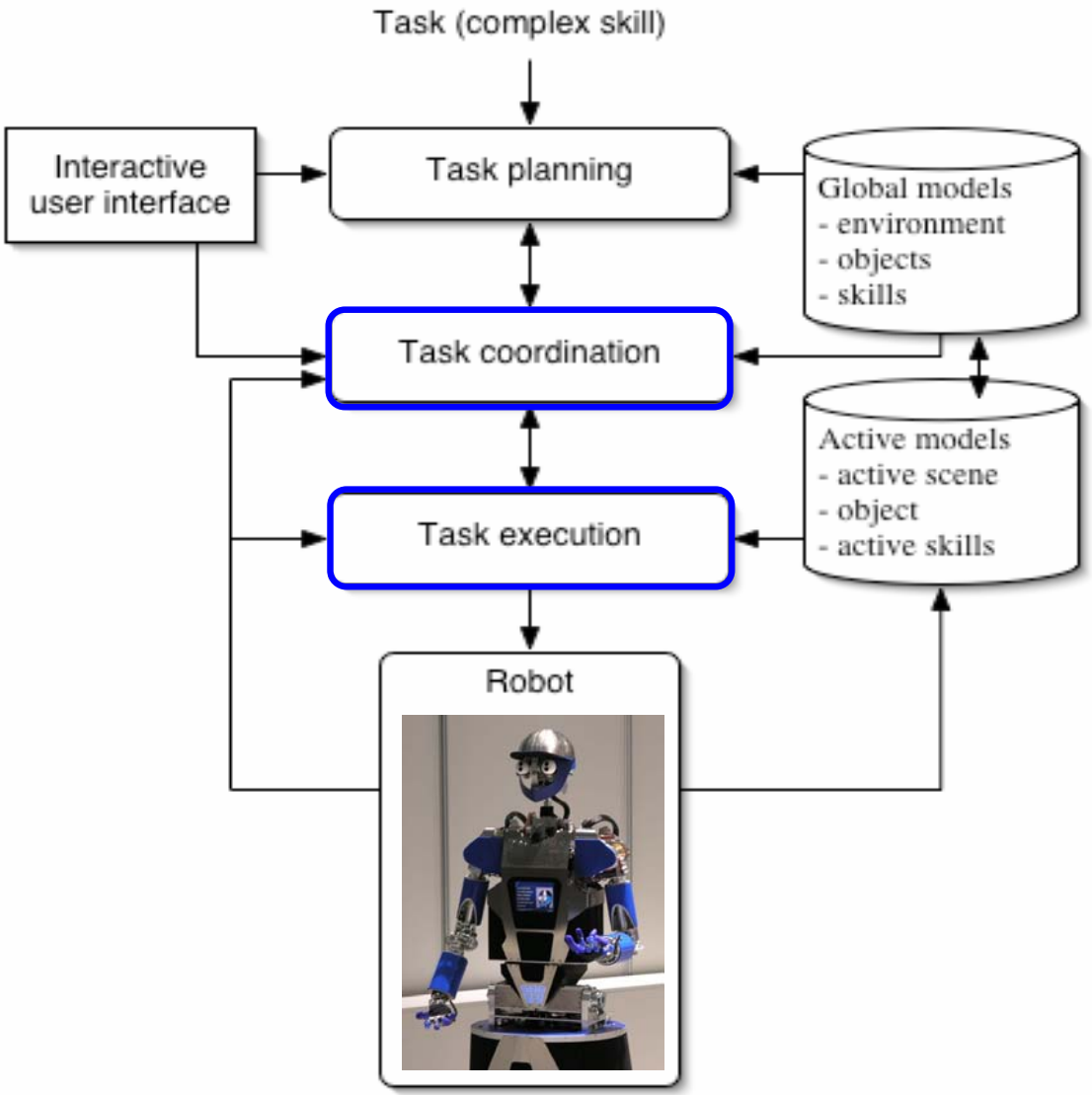
Task planning level

- Generates task plan
- Specifies subtasks for all subsystems of the robot
- scheduling of tasks and management of resources
- Includes synchronization information
- Subtasks hold all necessary parameters:
  - External (objects etc.)
  - Internal (joint velocities etc.)
  - Trajectories



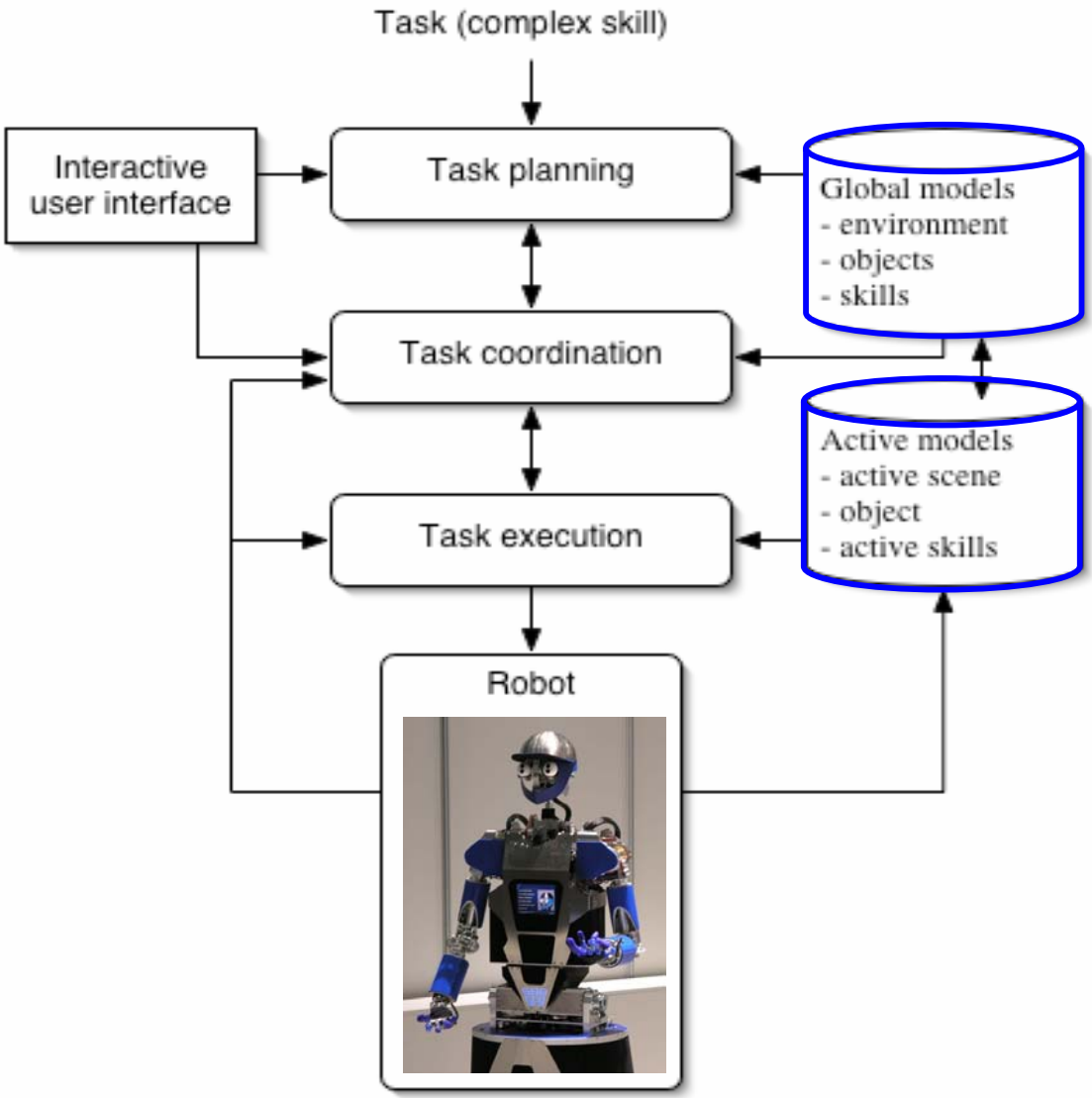
Control Architecture

- Task coordination level
  - Generation of sequential or parallel primitive actions
  - Subtask synchronization by including sync. constraints
  - Representation: Condition/Event Petri Nets
- Task execution level
  - execute specified sensory motor control commands



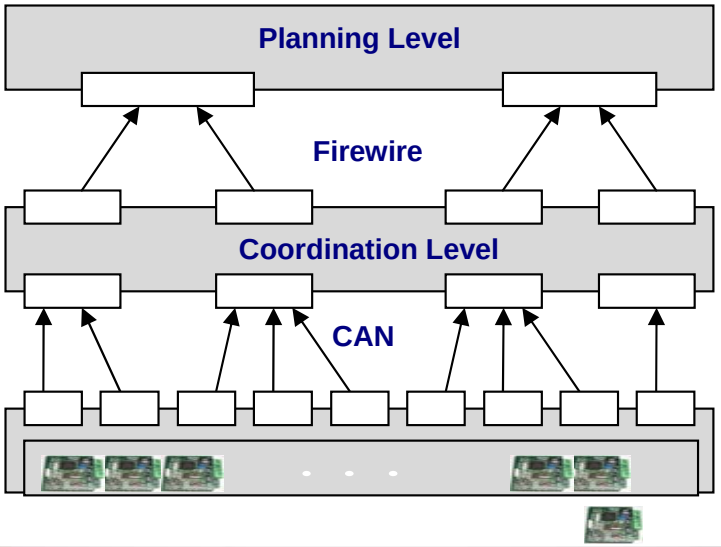
Control Architecture

- Global models:
  - Object models include all necessary features
  - Skill models
  - Maps, 3D models of the environment
  - Robot model (kinematics, dynamics, sensor/actor models)
- Active "local" models
  - "Cache" for runtime mode
  - Update of global models
  - can be updated by the perception system



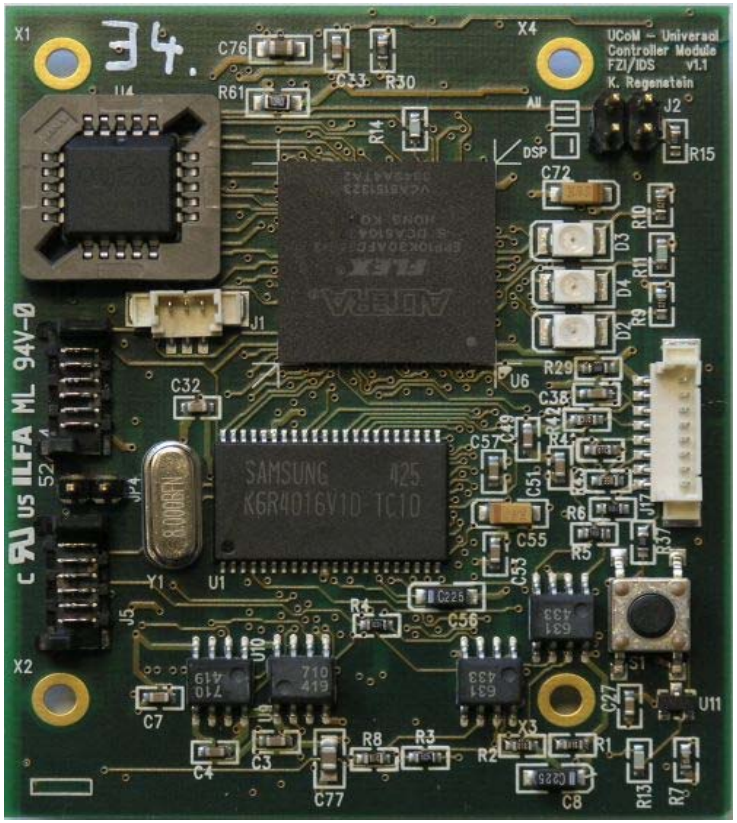
Computer Architecture

- Distributed computing architecture:
  - Planning level (Embedded PCs)
  - Coordination level (Embedded PCs)
  - Execution level (DSP/FPGA-Boards, UCoM)
- Connectivity: Ethernet, Firewire, CAN-Bus
- Properties
  - Scalability
  - Parallel computing
  - Modularity



Universal Controller Module (UCoM)

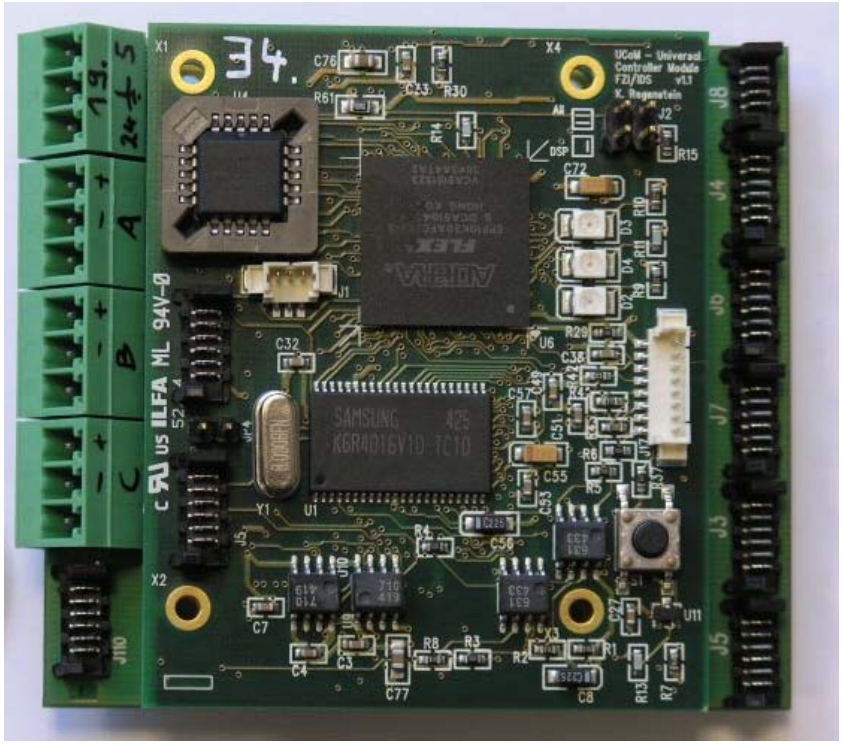
- Size: 57mm * 70mm * 15mm
- 80MHz DSP: Motorola Hybrid Controller DSP56F803
  - Interfaces: CAN, SCI, SPI, JTAG
  - Analog Ports: 2 * 4 channels ADC
  - Motorcontrol: 6 PWM
- External RAM
- FPGA: Altera EPF10k30
- FPGA-Configuration: Altera EPC2-ROM
- All Components programmable via JTAG
- DSP programmable via CAN-Bus



Work done by K. Regenstein
[Humanoids 2006]

Universal Controller Module (UCoM)

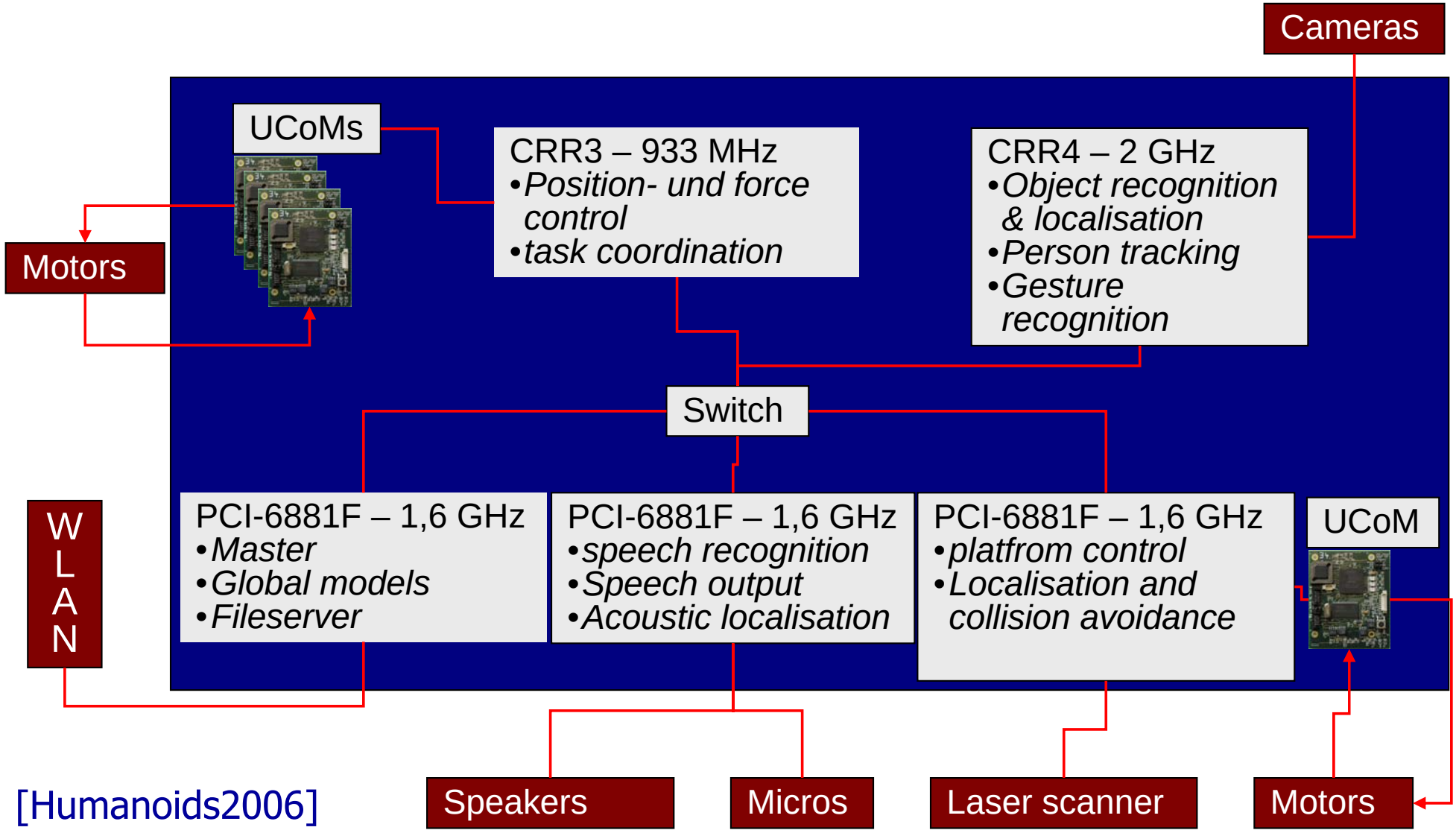
- Size: 80mm * 70mm * 20mm
- Up to 3 Motors (24V up to 5A)
- 6 Encoder-Ports (6-pin; 2 power supply, 4 IO)
- Serial Peripheral Interface (SPI)
- 3 General Purpose IOs



Work done by K. Regenstein
[Humanoids 2006]

Tamim Asfour
Universität Karlsruhe

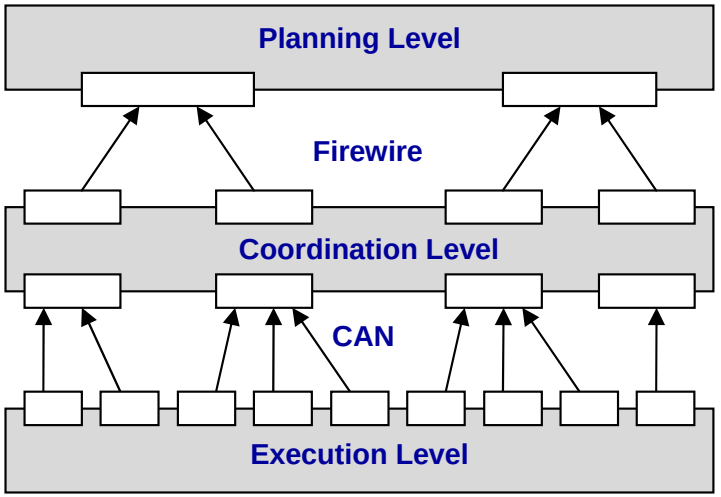
Computer architecture: ARMAR-III

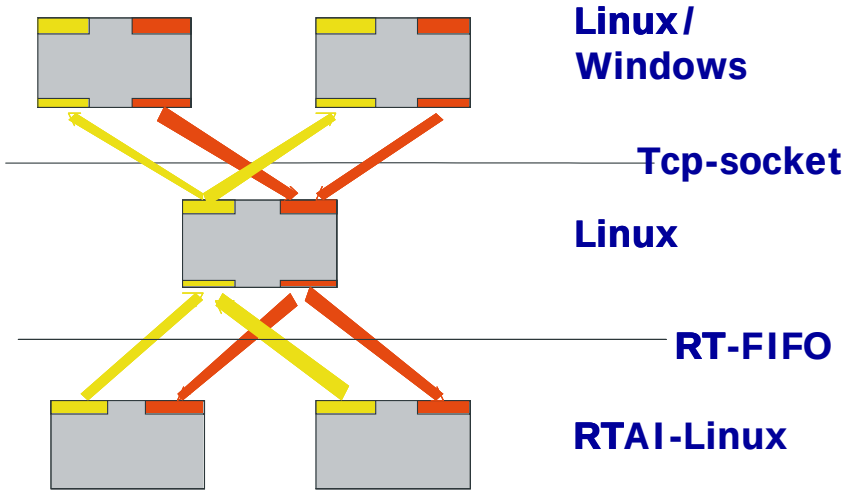


[Humanoids2006]

Software Framework

- **Modular Controller Architecture (http://mca2.sourceforge.net)**
  - Designed for real-time capabilities
  - Supports distributed computing
  - RTAI-Linux
  - Easy to parameterize
  - Easy to administrate
  - Reusability of modules
  - Defined interfaces
  - Module groups form complex functional units





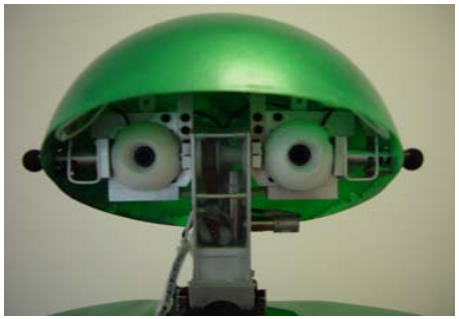
Programming of manipulation tasks

- Inverse Kinematics of a humanoid redundant robot arm
- A kinematic approach for the generation of human-like manipulation motions
- Framework for Motion Coordination
- ARMAR-II performing various tasks

Next Generation of Humanoid Robots

- Object manipulation in working environments that
  - are populated by humans
  - are made for humans
  - should not be modified in any way for the benefit of the robot
- Natural interaction with humans
  - Natural communication channels
  - Human-like motion

ARMAR-II

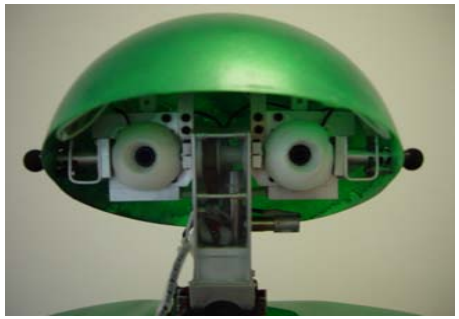


- Mobile platform
- 3-DOF torso
- 7-DOF arms
- 2 laser scanner
- 6D force/torque sensors
- Sensor-head
  - Stereo vision
  - 2 microphones

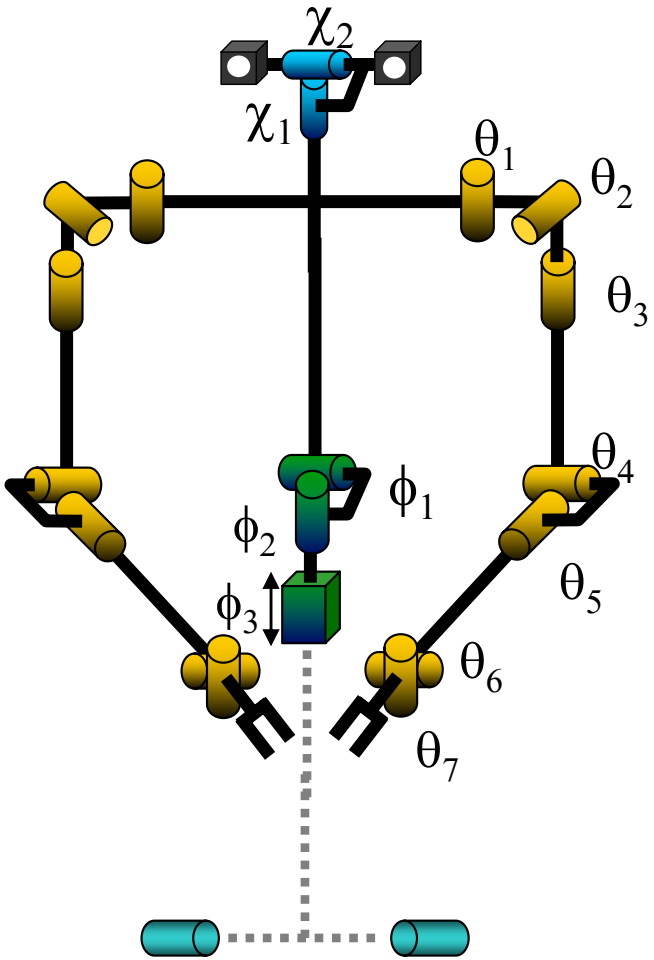
- Speech input and output
- Collision-free navigation
- Acoustic localization
- Visual user tracking
- Visual color-based object tracking
- Redundancy resolving
- Hybrid force position control
- Coordination of task execution

Tamim Asfour
Universität Karlsruhe

ARMAR-II



- Mobile platform
- 3-DOF torso
- 7-DOF arms
- 2 laser scanner
- 6D force/torque sensors
- Sensor-head
  - Stereo vision
  - 2 microphones



- Speech input and output
- Collision-free navigation
- Acoustic localization
- Visual user tracking
- Visual color-based object tracking
- Redundancy resolving
- Hybrid force position control
- Coordination of task execution

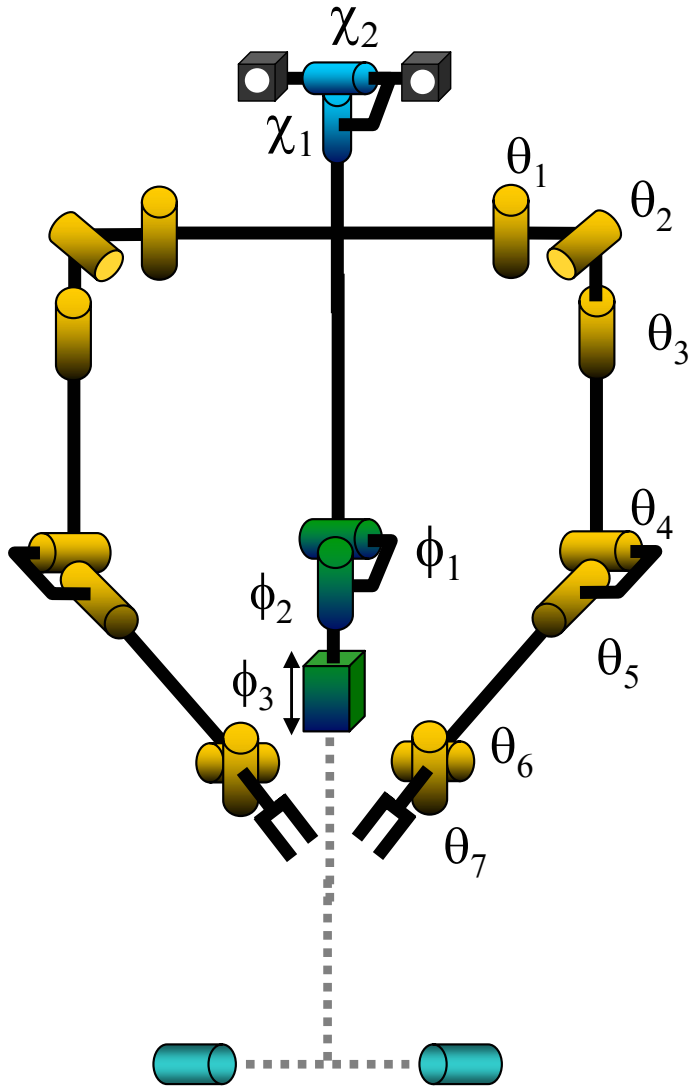
Kinematics Model of the Robot Arm

Anthropomorphic redundant arm with 7 DOFs

- Infinite number of joint angle trajectories leads to the same end-effector trajectory.

Closed-Form Solution of the IK-Problem:

- Mathematical description of the arm redundancy is needed.



Tamim Asfour
Universität Karlsruhe

Control of Redundant Arms

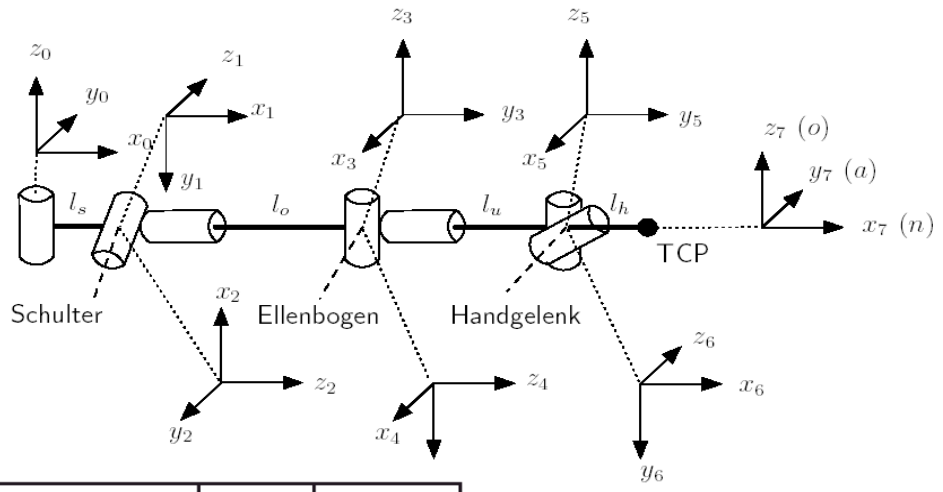
- "Classical" use of the kinematics redundancy to
  - avoid joint limits [Ligéois]
  - avoid obstacles [Maciejewski]
  - avoid singular configurations [Nakmura & Hanafusa]
  - provide a fault tolerant operation [Gramm]
  - optimize the robot arm dynamics [Hollerbach & Suh]
- Use of the kinematics redundancy for the generation of human like motions:
  - Hypothetical cost functions suggested to explain the principles of human arm movements [Potkonjak, ...]
  - Motion capture techniques [Ude, Hodgins, ...]

Our Approach

- Generation of human-like manipulation motions from the kinematics point of view:
  - Closed-form solution of the inverse kinematics Problem (IK-Problem)
  - Application of a hypothesis from the neurophysiology to resolve the redundancy resulting in human-like motions

Closed-Form Solution of the IK-Problem

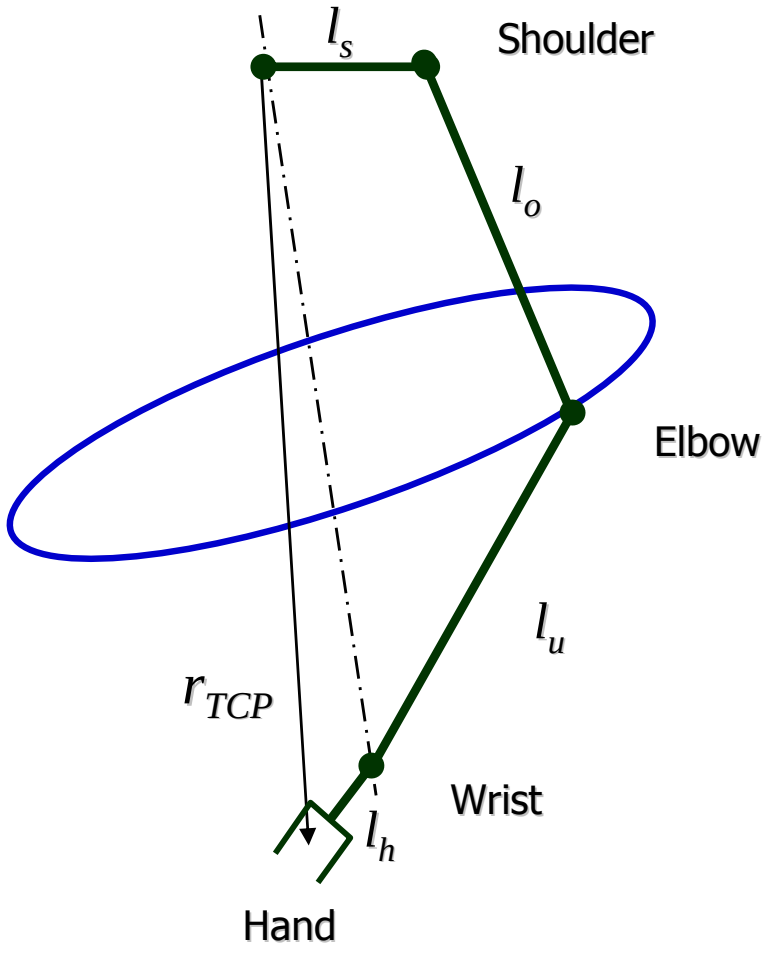
DH-Parameter des Arms



Gelenk	θ_i	α_i	a_i	d_i	θ_{max}	θ_{min}
1	θ_1	$-90°$	$l_s = 30\ mm$	0	$90°$	$-90°$
2	$-90° + \theta_2$	$-90°$	0	0	$90°$	$-90°$
3	$90° + \theta_3$	$90°$	0	$l_o = 223,5\ mm$	$90°$	$-230°$
4	θ_4	$-90°$	0	0	$145°$	$0°$
5	θ_5	$90°$	0	$l_u = 270\ mm$	$350°$	$0°$
6	$90° + \theta_6$	$-90°$	0	0	$45°$	$-45°$
7	θ_7	$90°$	$l_h = 106\ mm$	0	$45°$	$-45°$

Closed-Form Solution of the IK-Problem

- The arm redundancy can be described by a curve which results from the intersection of the workspaces of the upperarm and forearm.
- Determination of the position and orientation of the Elbow for a given position and orientation of the TCP.
- Determination of the joint angles of the arm in closed-form.

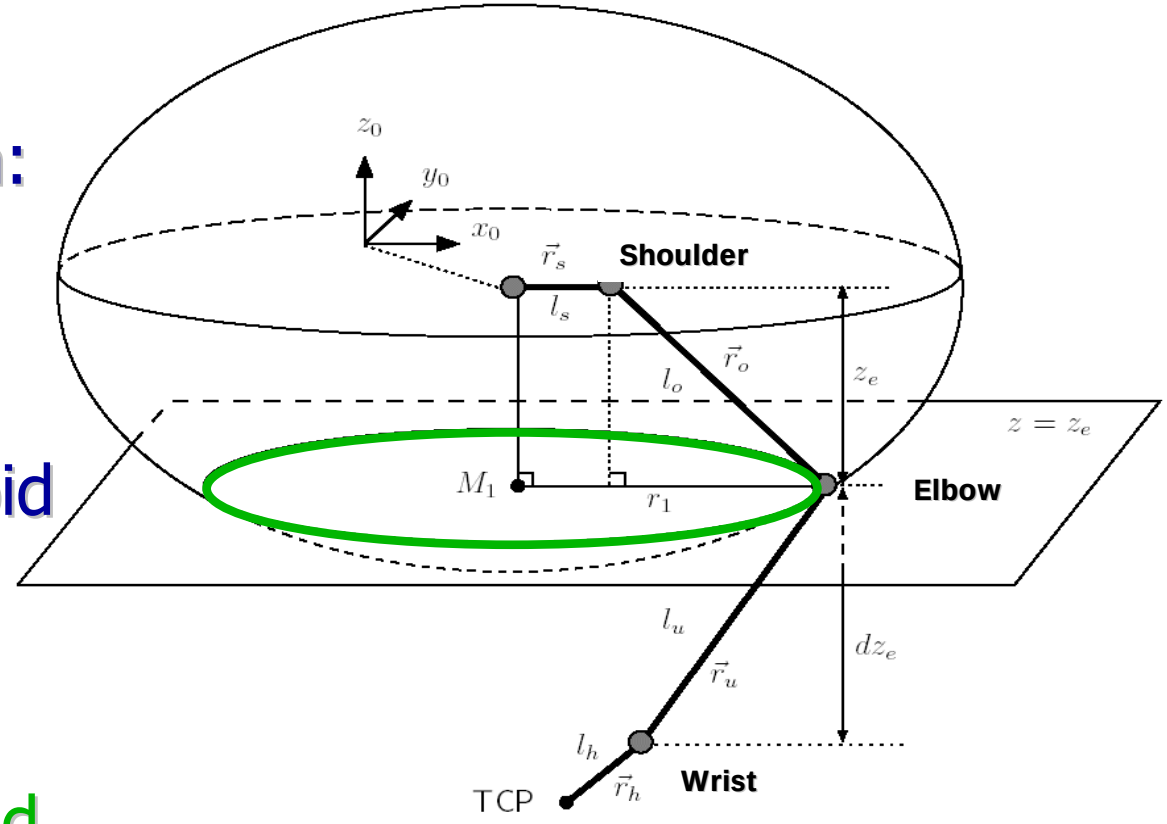


Closed-Form Solution of the IK-Problem

Workspace of the upperarm:

- Ellipsoid centered in $(x_0, y_0, z_0)^T$
- Intersection of this ellipsoid with the plane
  $z = z_{elbow}$

➔ Circle with radius r_1 and center in M_1



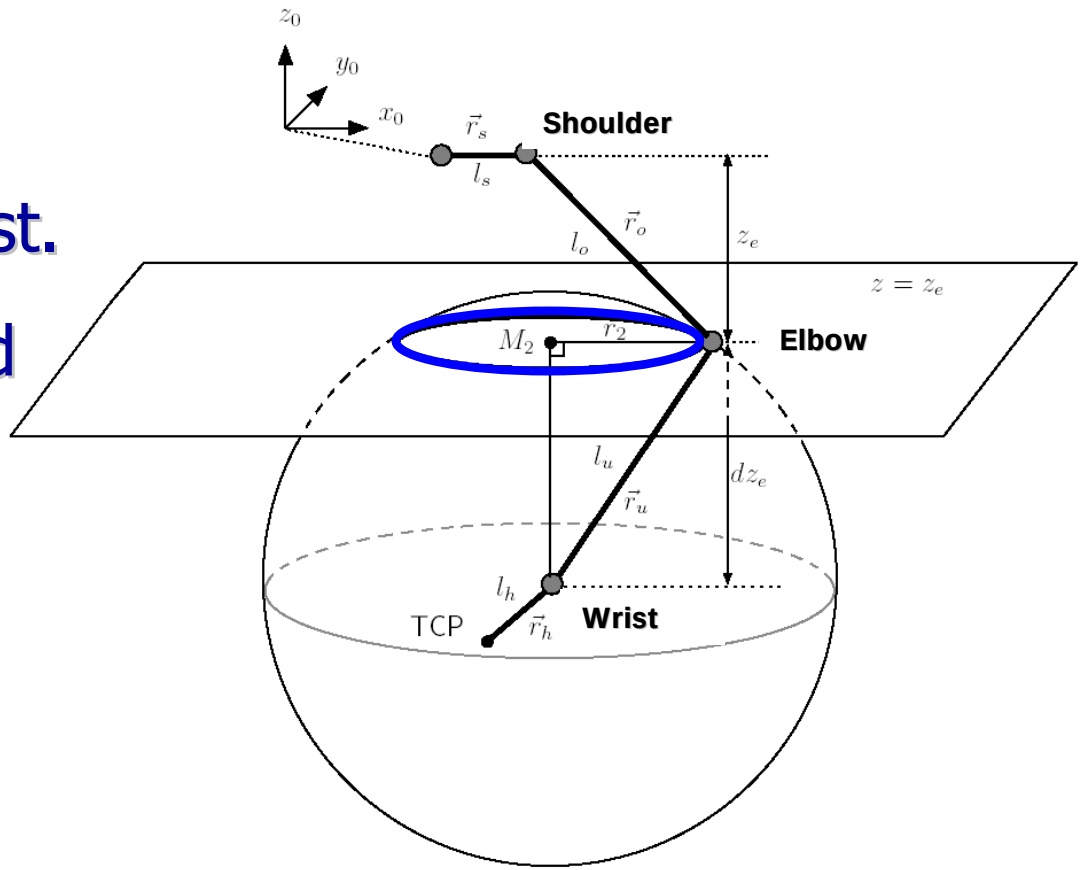
$$r_1 = l_s + \sqrt{l_o^2 - z_e^2}$$
$$\vec{M}_1 = (0,\ 0,\ z_e)^T$$

Closed-Form Solution of the IK-Problem

Workspace of the forearm:

- Sphere centered in the wrist.
- Intersection of this ellipsoid with the plane
  $z = z_{elbow}$

➔ Circle with radius r_2 and center in M_2



$$r_2 = \sqrt{l_u^2 - dz_e^2}$$

$$\vec{M}_2 = (hg_x, hg_y, z_e)^T$$

Closed-Form Solution of the IK-Problem

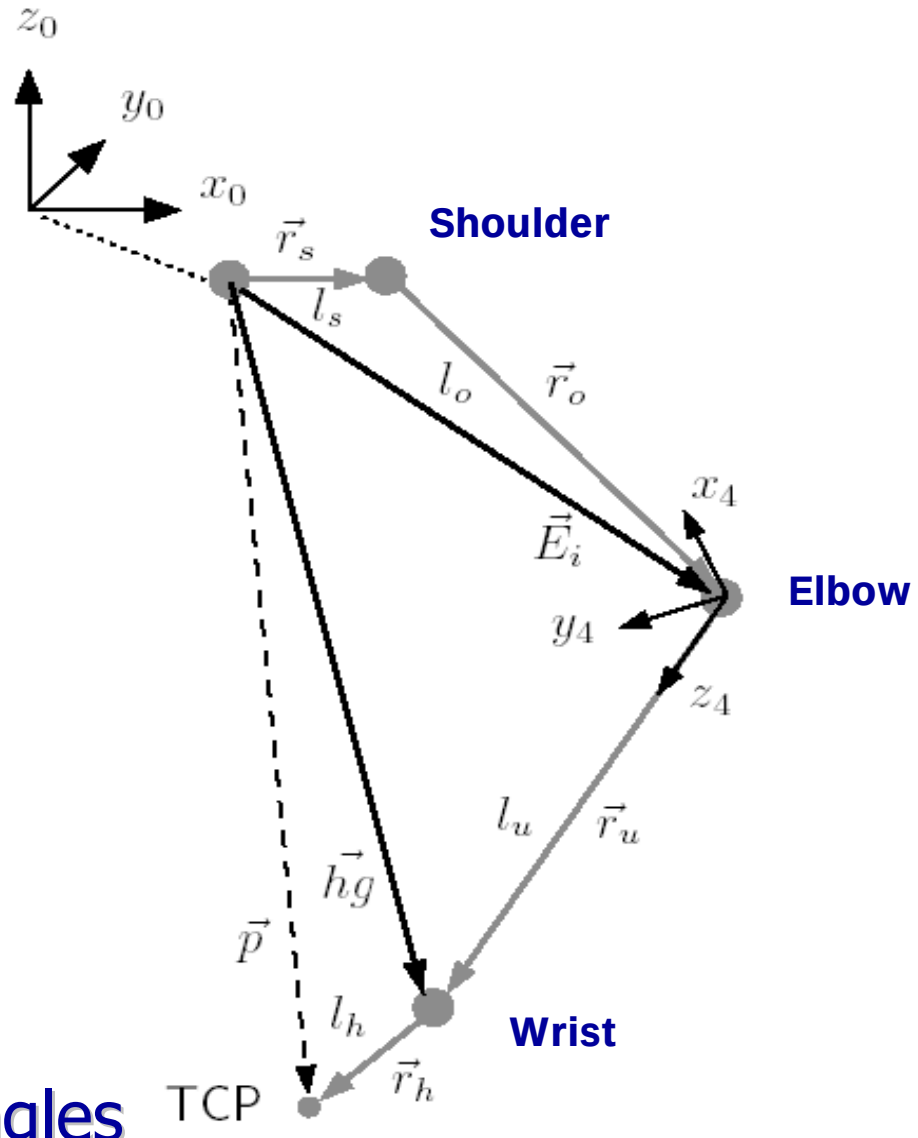
- Feasible positions of the elbow are given by the intersection points E_i (i =1,2) of the circles M_1 and M_2
- The position and orientation of the elbow is given by:

$$\vec{z}_4 = \frac{\vec{hg} - \vec{E}_i}{\|\vec{hg} - \vec{E}_i\|}$$

$$\vec{y}_4 = \frac{\vec{z}_4 \times (\vec{r}_s - \vec{E}_i)}{\|\vec{z}_4 \times (\vec{r}_s - \vec{E}_i)\|} \qquad i = 1, 2$$

$$\vec{x}_4 = \vec{y}_4 \times \vec{z}_4$$

➔ Analytical solution for the upperarm and forearm joint angles



Determination of the joint angles

TCP:

$$\boldsymbol{T}_{TCP} = {}^{0}\boldsymbol{T}_{7} = \prod_{i=1}^{7} \boldsymbol{B}_i = \begin{pmatrix} \vec{n} & \vec{o} & \vec{a} & \vec{p} \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Wrist:

$$\vec{hg} = \vec{p} - l_h \cdot \vec{n}$$

Elbow:

$$\boldsymbol{T}_{Ellenbogen} = \begin{pmatrix} \vec{x}_4 & \vec{y}_4 & \vec{z}_4 & \vec{E}_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad i = 1, 2$$

$$= \boldsymbol{B}_1 \cdot \boldsymbol{B}_2 \cdot \boldsymbol{B}_3 \cdot \boldsymbol{B}_4$$

Tamim Asfour
Universität Karlsruhe

Determination of the joint angles

Joint angles of the upperarm:

$$\underbrace{B_1^{-1} \cdot T_{Ellenbogen}}_{L_1} = \underbrace{B_2 \cdot B_3 \cdot B_4}_{R_1}$$

$$\underbrace{B_2^{-1} \cdot B_1^{-1} \cdot T_{Ellenbogen}}_{L_2} = \underbrace{B_3 \cdot B_4}_{R_2}$$

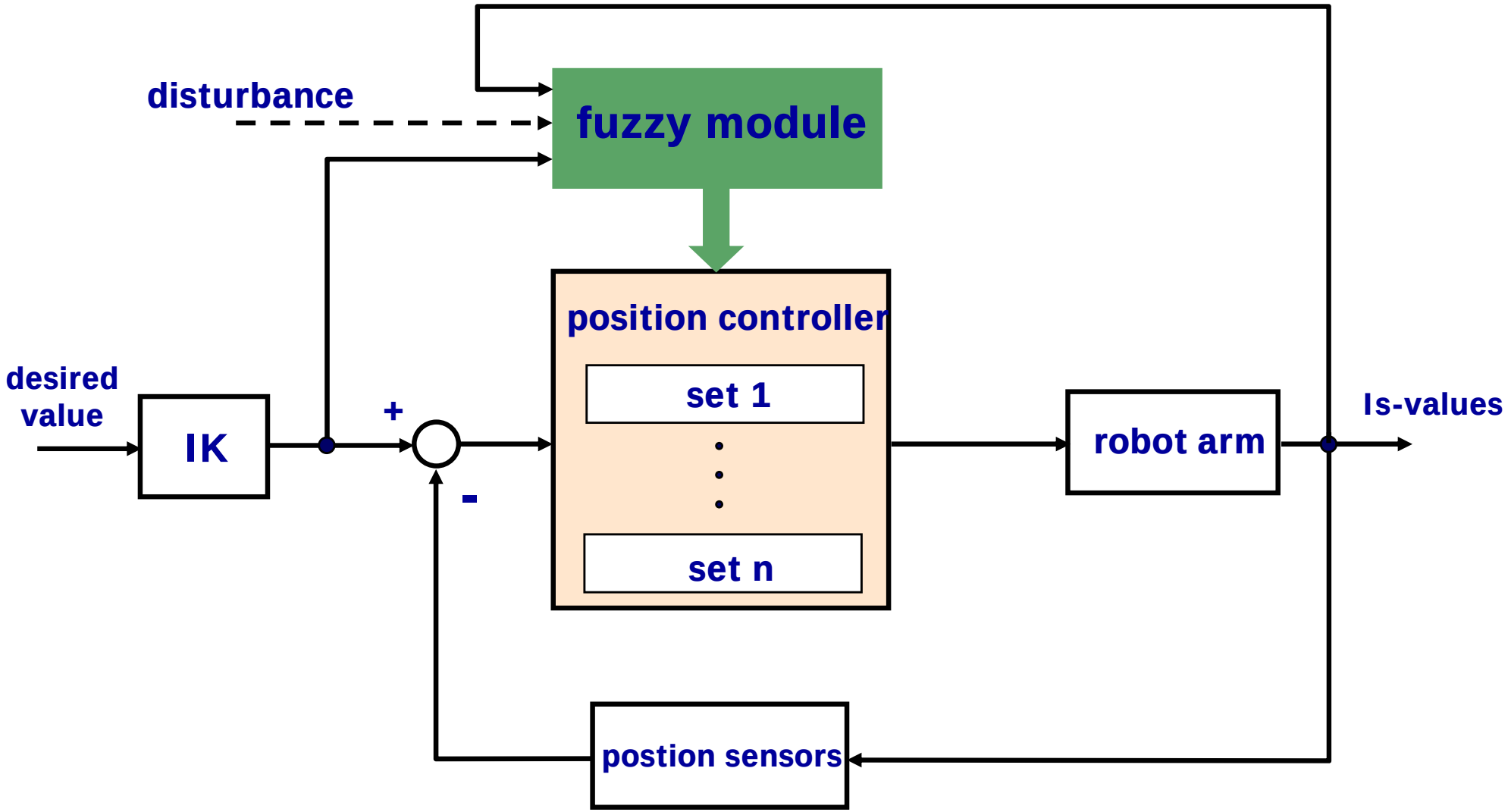
Joint angles of the forearm:

$$T'' = B_4^{-1} \cdot B_3^{-1} \cdot B_2^{-1} \cdot B_1^{-1} \cdot T_{Ellenbogen}$$

$$= B_5 \cdot B_6 \cdot B_7$$

Tamim Asfour
Universität Karlsruhe

Position control System



Human-like Robot Arm Motions

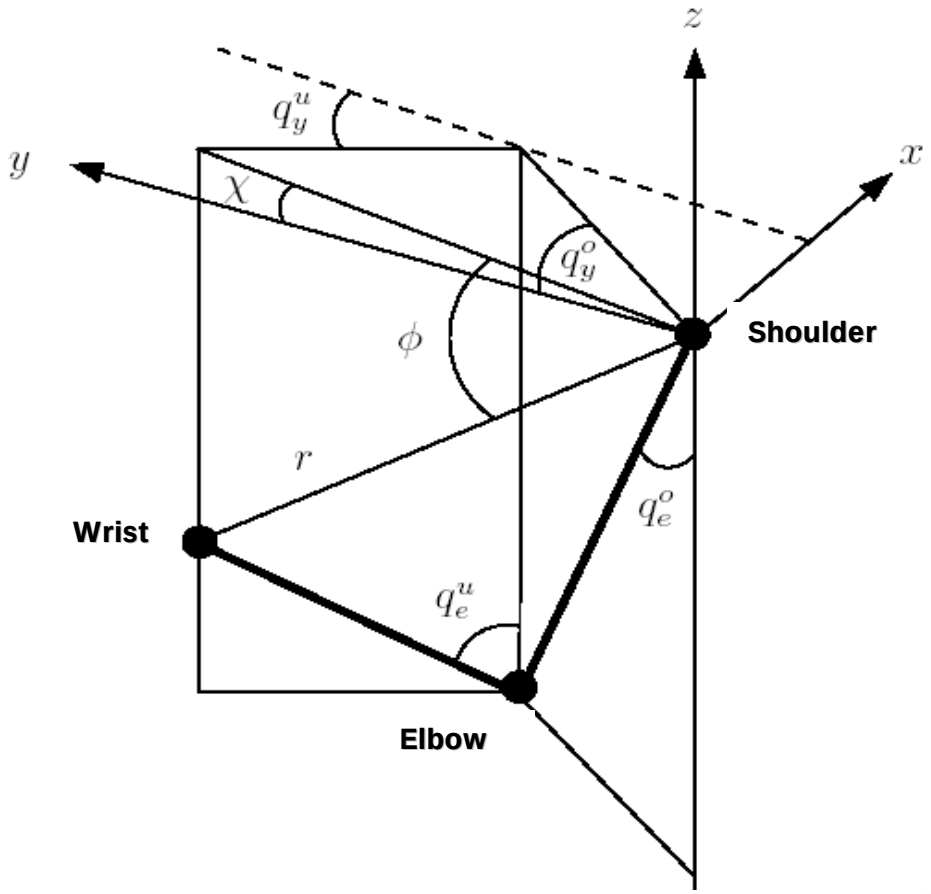
- How does the central nerve system (CNS) solve the IK-Problem?
- How does the CNS resolve the redundancy of the human arm?
- Soechting and Flanders:
  - Human arm movements are planned in a shoulder-centered spherical coordinates.
  - The wrist position is mapped into a natural posture described by four parameters (elevation and yaw angles of the upperarm and forearm)

Human-like Robot Arm Motions

Representation of the arm posture through four Parameters:

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

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



Human-like Robot Arm Motions

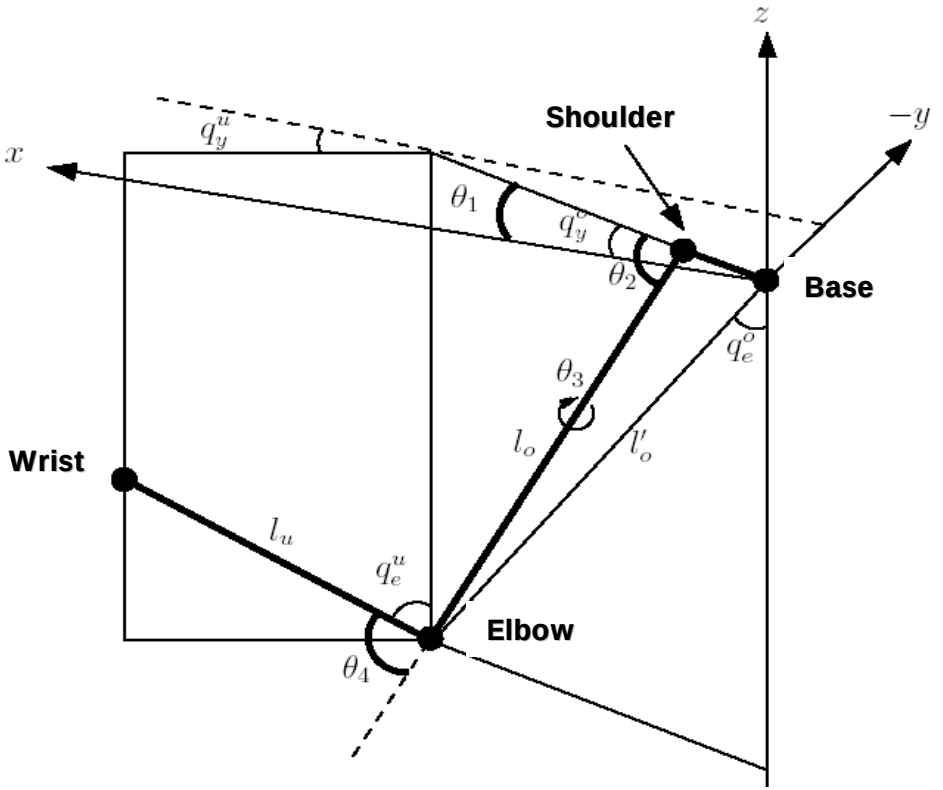
Mapping of the posture representation parameters into the joint angles of the arm, which are responsible for the position of the wrist:

$$\theta_1 = -q_y^o$$

$$\theta_2 = \pi - q_e^o - \arccos\left(\frac{l_s \cdot \cos(q_e^o)}{l_o}\right)$$

$$\theta_4 = \pm \arccos\left(\frac{< \vec{l_o}, \vec{l_u} >}{||\vec{l_o}|| \cdot ||\vec{l_u}||}\right)$$

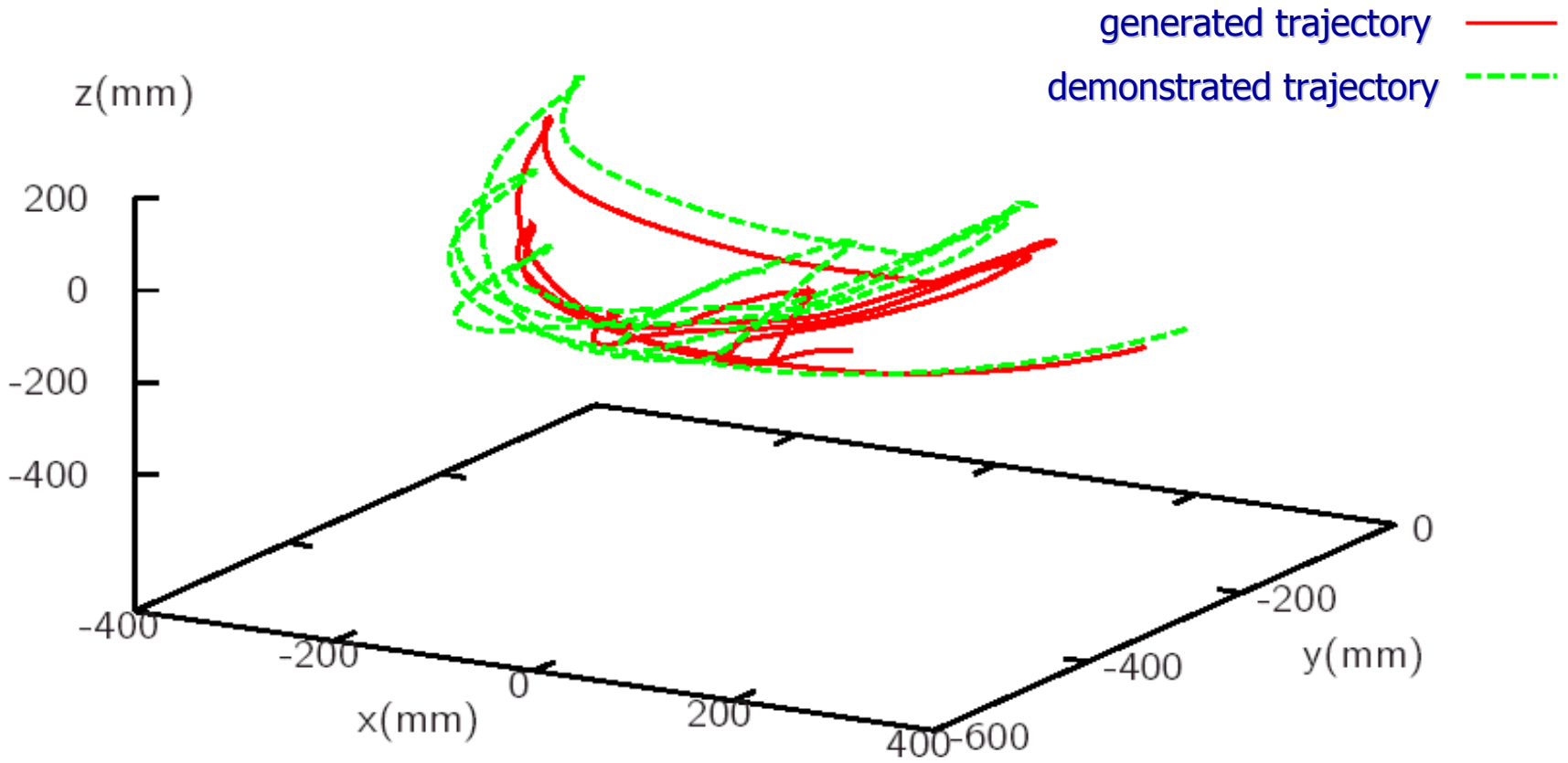
$$\theta_3 = \arcsin\left(\frac{\sin(\theta_2) \cdot \cos(\theta_4) + \cos(q_e^u)}{\cos(\theta_2) \cdot \sin(\theta_4)}\right)$$



The exact position und orientation of the TCP can be achieved by a final adjustment step

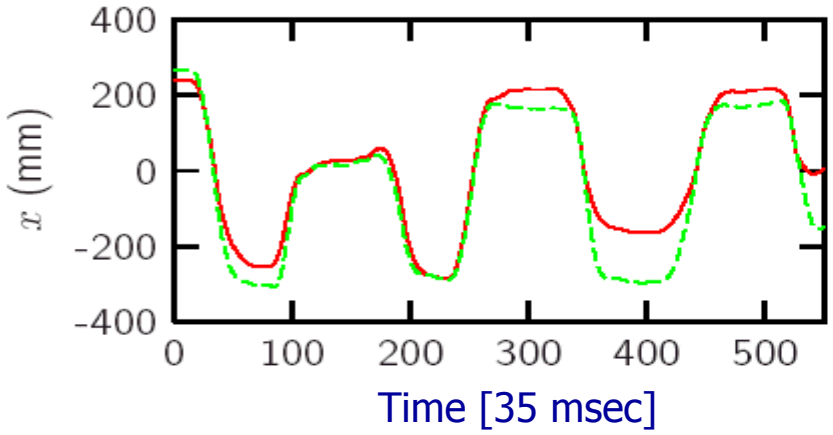
Manipulation Task I

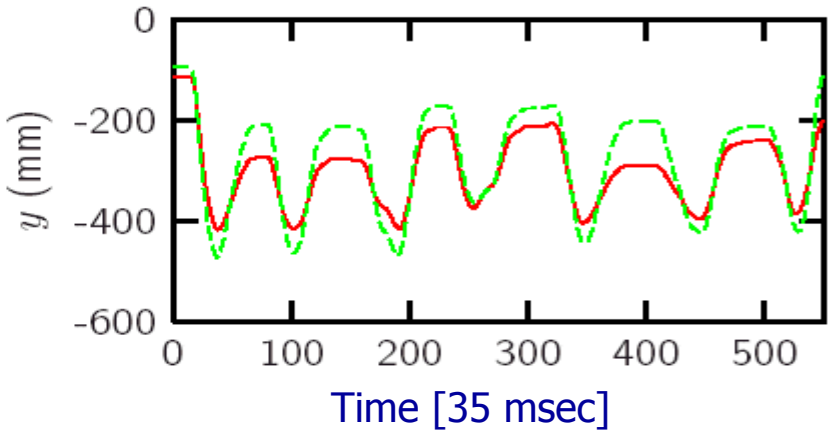
Task: „Lay a table“

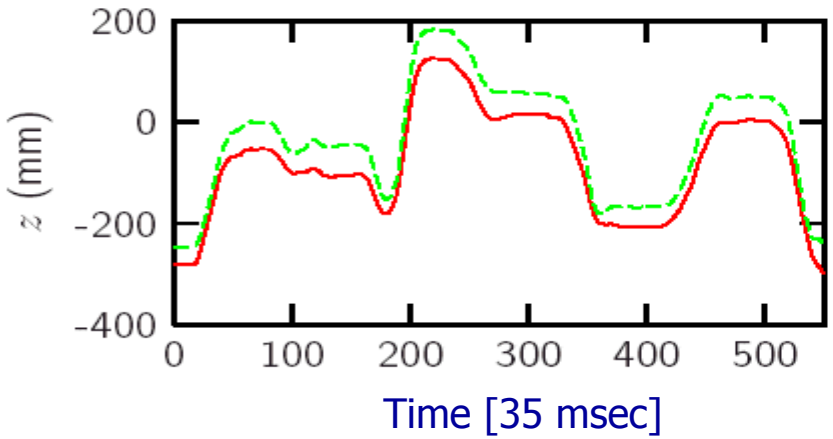


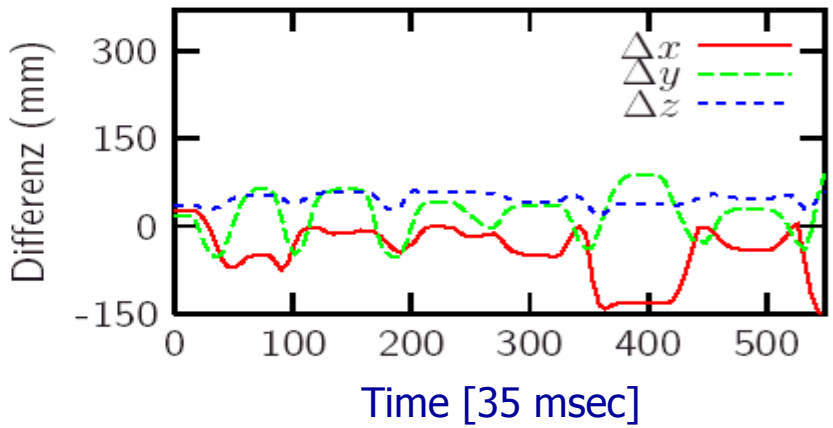
Tamim Asfour
Universität Karlsruhe

Manipulation Task I









Differences can be eliminated using the remaining joint angles θ_5, θ_6, θ_7 and small modifications of θ_1, θ_2, θ_3, θ_4

Figure eight

$$
\begin{aligned}
x(t) &= x_0 \\
y(t) &= y_0 + 200 \cdot \sin(t) \\
z(t) &= z_0 + 200 \cdot \sin(t)\cos(t)
\end{aligned}
$$

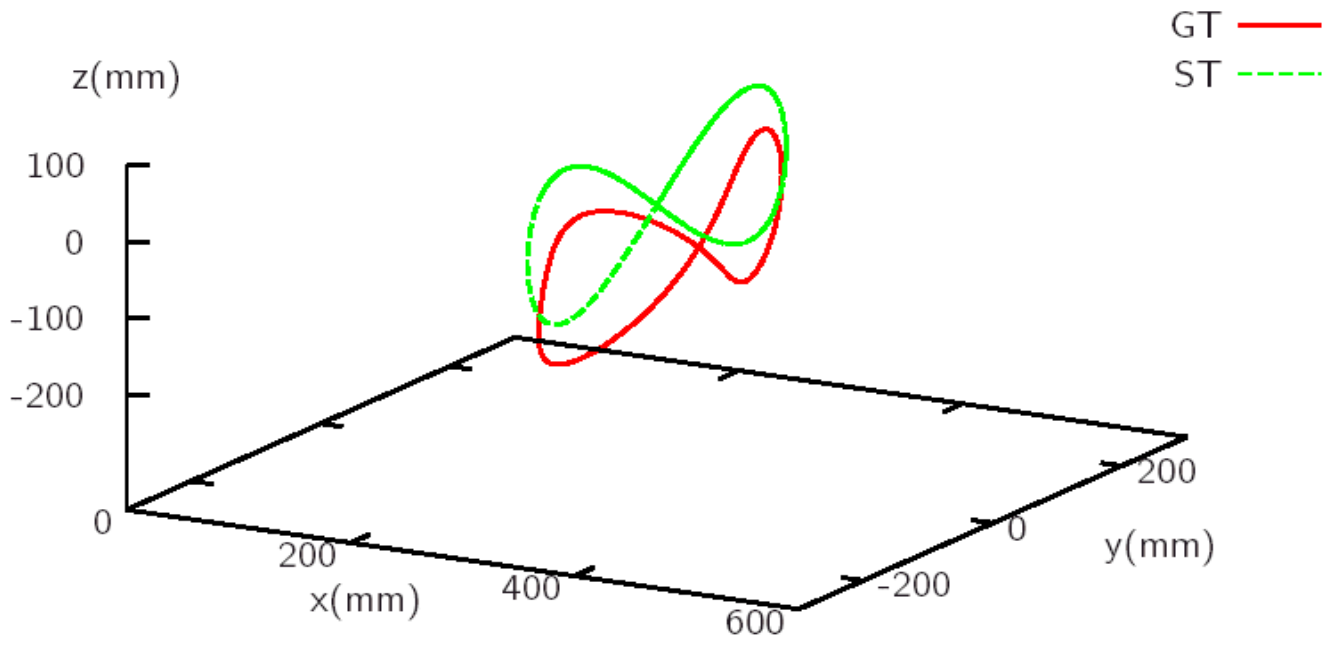
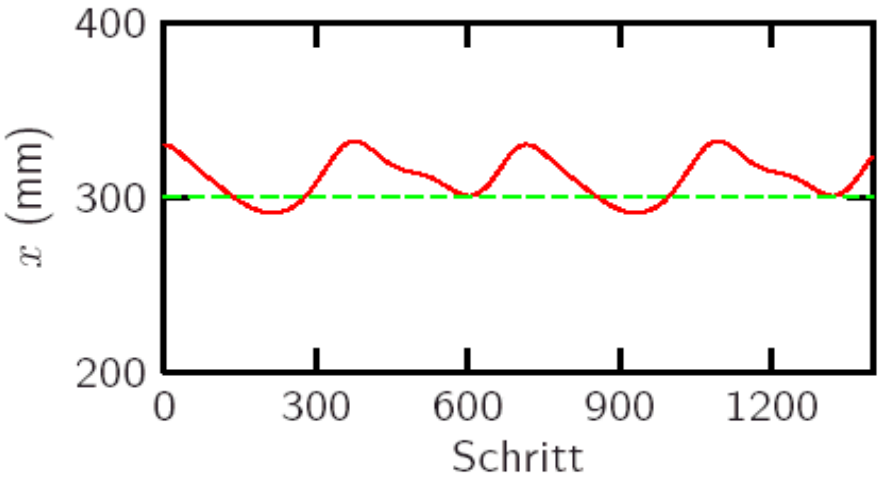
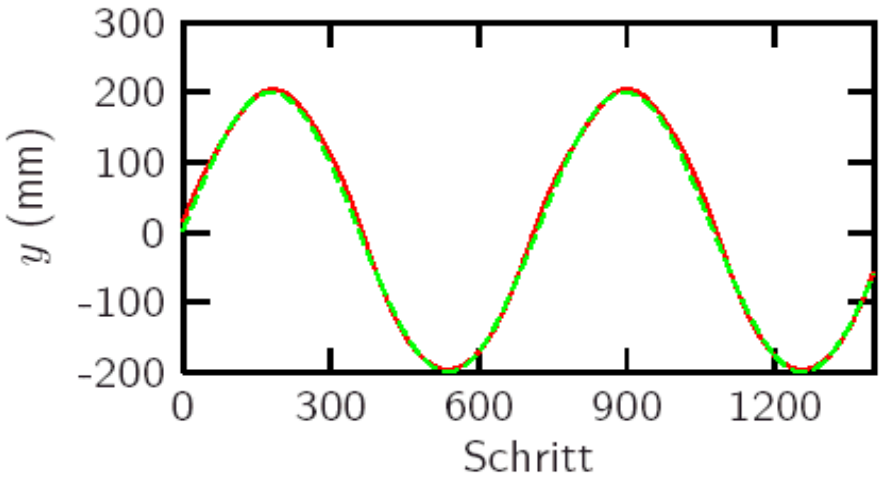
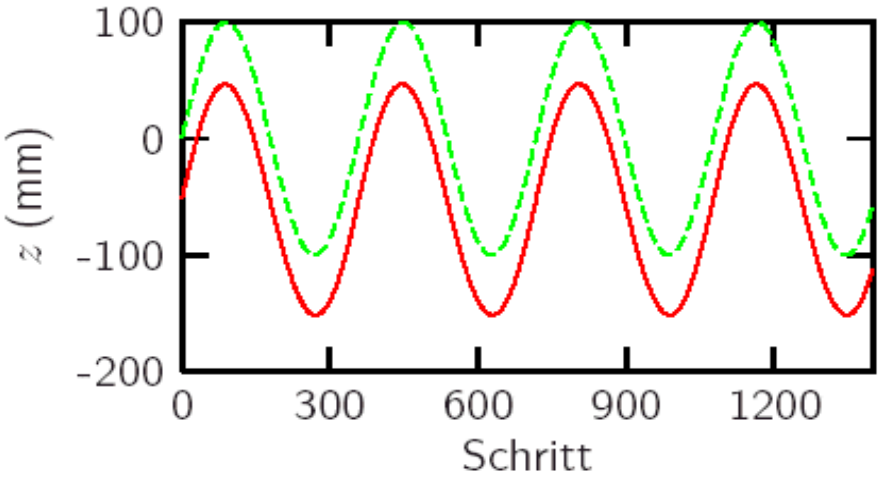


Figure eight (sampling time = 8 ms)







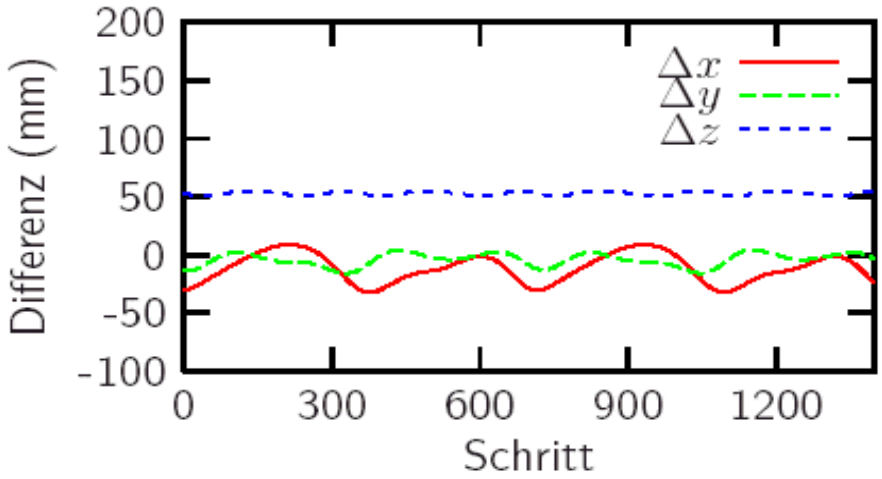
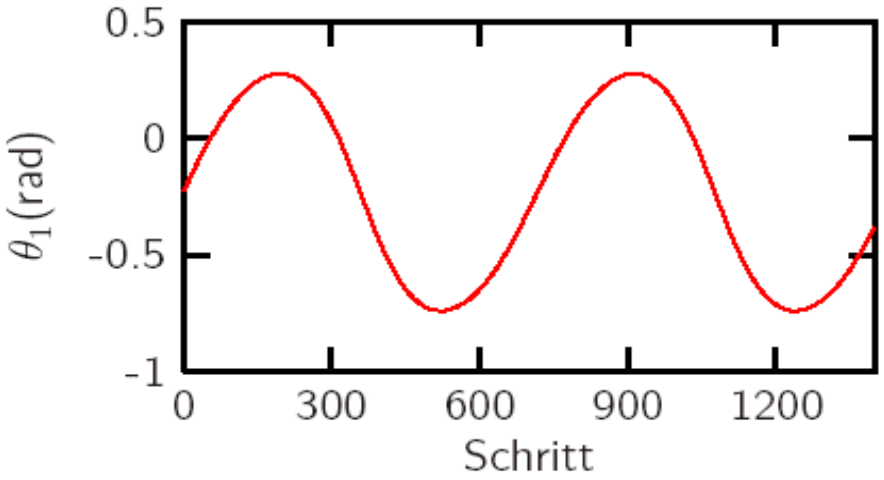
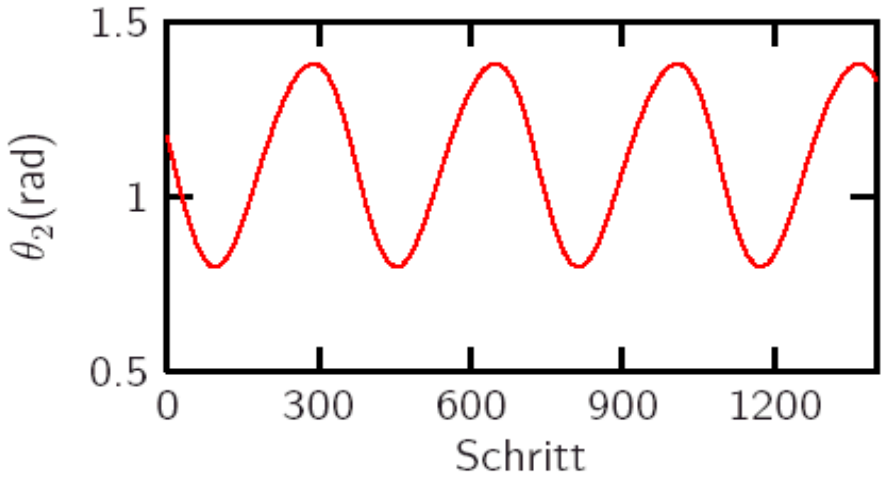
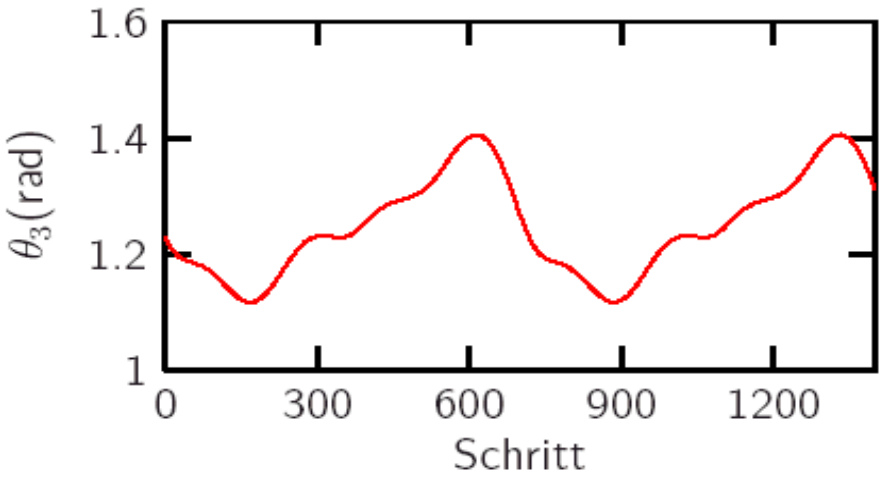
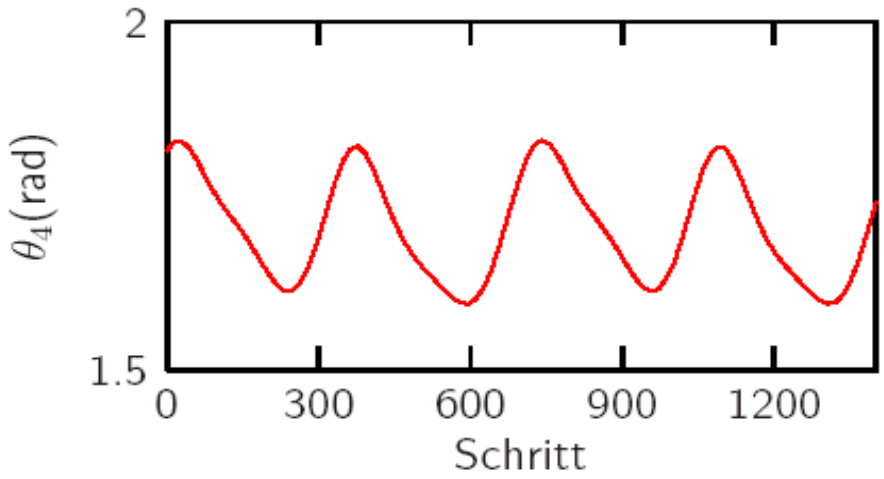


Figure eight (sampling time = 8 ms)

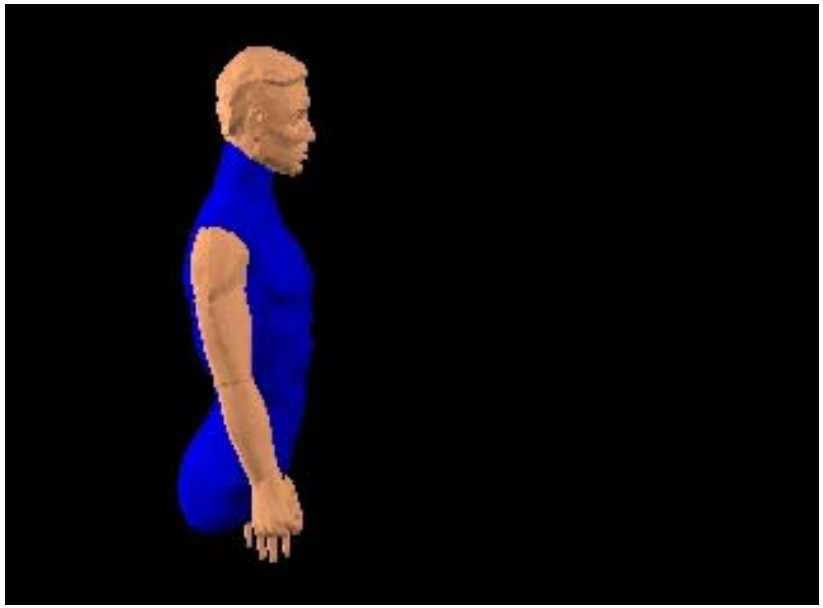








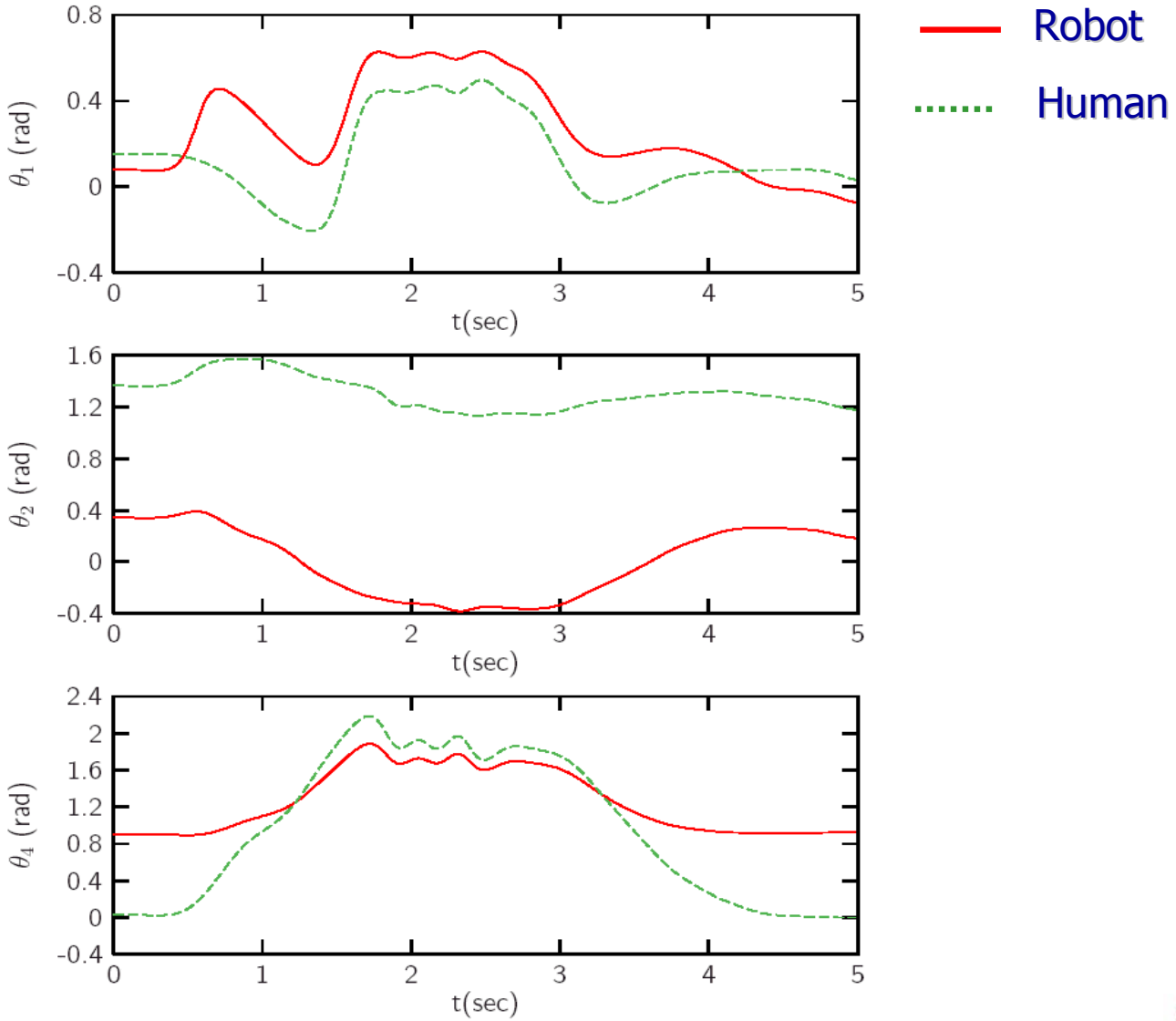
Comparison with Human Motions (Knocking)



Motion generated
according to [Ales Ude]

Comparison with Human Motions

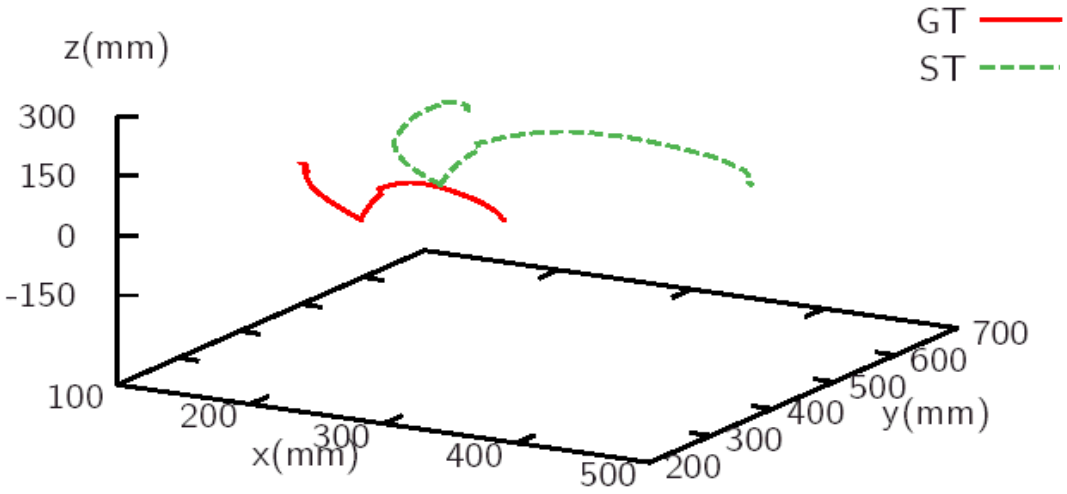
- Similar behavior in the main phase of the motion
- Difference in θ_2 due to the different motion range of this joint

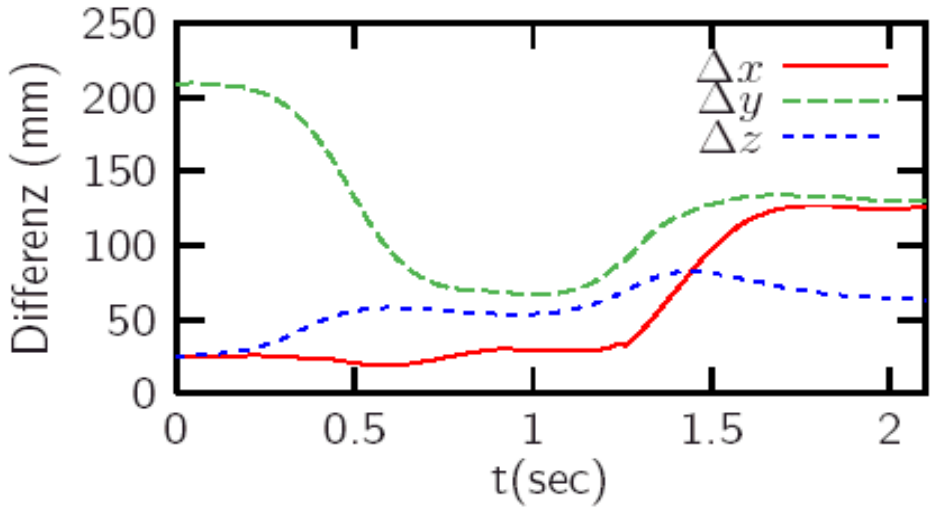


Tamim Asfour
Universität Karlsruhe

Manipulation Task II

Complex manipulation task





Differences can be only be eliminated through significantly large modifications of θ_1, θ_2, θ_3, θ_4 ➔ The resulting arm posture is not guaranteed to be human-like.

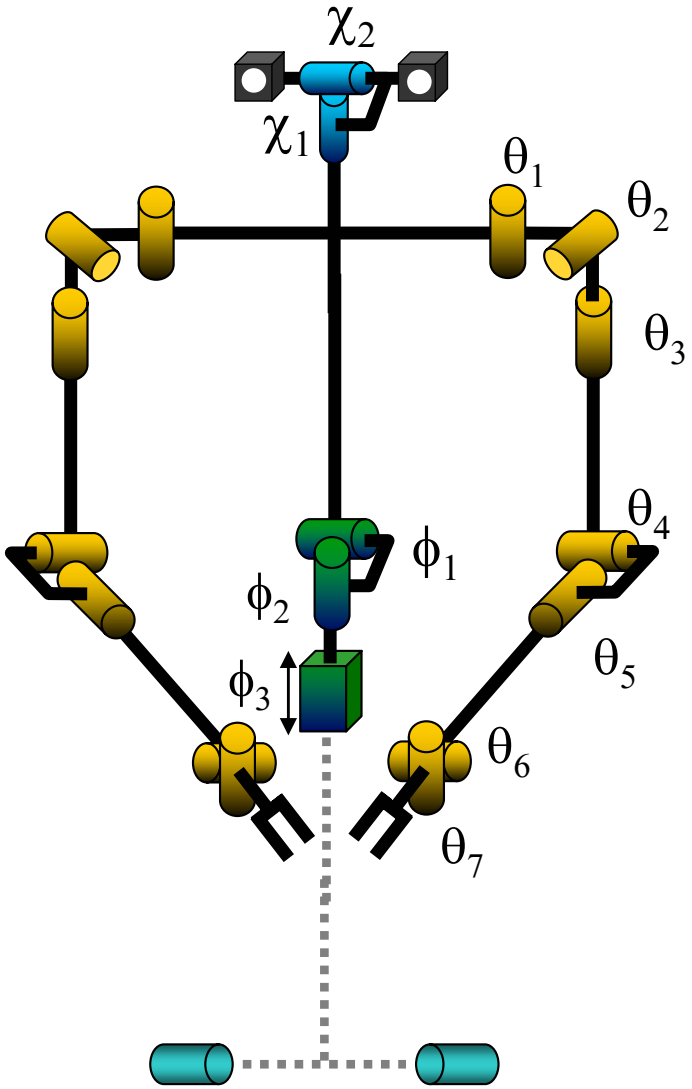
Tamim Asfour
Universität Karlsruhe

Human-like Robot Arm Motions

- Real-time control scheme of an anthropomorphic robot arm with a closed-form solution of the IK-Problem and human-like arm postures
- The approach does not consider the dynamics of manipulation tasks
- In case of complex manipulation tasks, the limited redundancy of the arm makes it impossible to achieve the exact position and orientation of the hand with small reconfiguration of the arm.
- References: [IROS03], [Humanoids06]

Framework for Motion Coordination

- The humanoid robot ARMAR has 21 mechanical degrees-of-freedom (without including the DOFs of the hand)
- seven subsystems: head, left arm, right arm, left hand, right hand, Torso and platform
- Coordinated execution of a task requires the scheduling of the subtasks and their synchronization with logical conditions, external and internal events
- Framework for coordinated execution of tasks using condition/event Petri nets
- Petri nets to efficiently represent both control and data flow within one formalism.



Tamim Asfour
Universität Karlsruhe

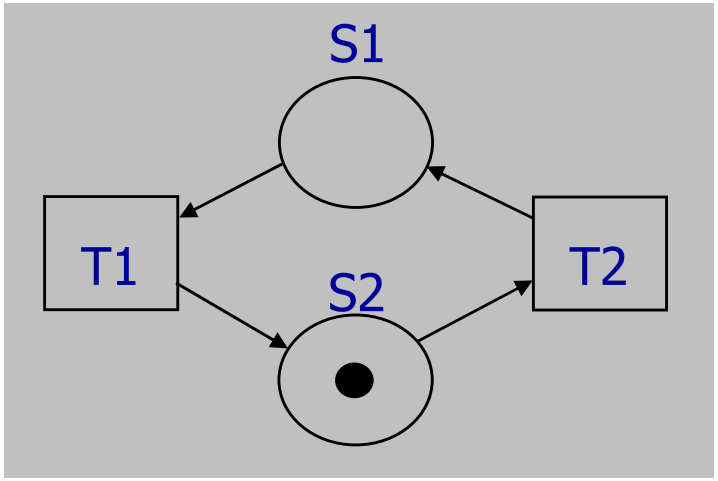
Condition/event Petri Net

Definition (Condition/event net) A condition/event Petri net is defined by the 4-tuple $N = (P, T, A, m_0)$, where

- $P = \{p_1, \ldots, p_{n_p}\}$ is a finite set of places,
- $T = \{t_1, \ldots, t_{n_t}\}$ is a finite set of transitions,
- $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$
- $A \subseteq (P \times T) \cup (T \times P)$ is a set of arcs,
- $m_0 : P \rightarrow \{0, 1\}$ is the initial marking. It defines the initial number of undistinguishable tokens on each place $p \in P$.
- **Enabling rule:** A transition $t_j \in T$ is enabled if all input places of t_j contain a token and all output places are empty.
- **Firing rule** An enabled transition may fire. On firing it removes the tokens from all its input places and places one token in each of its output places.

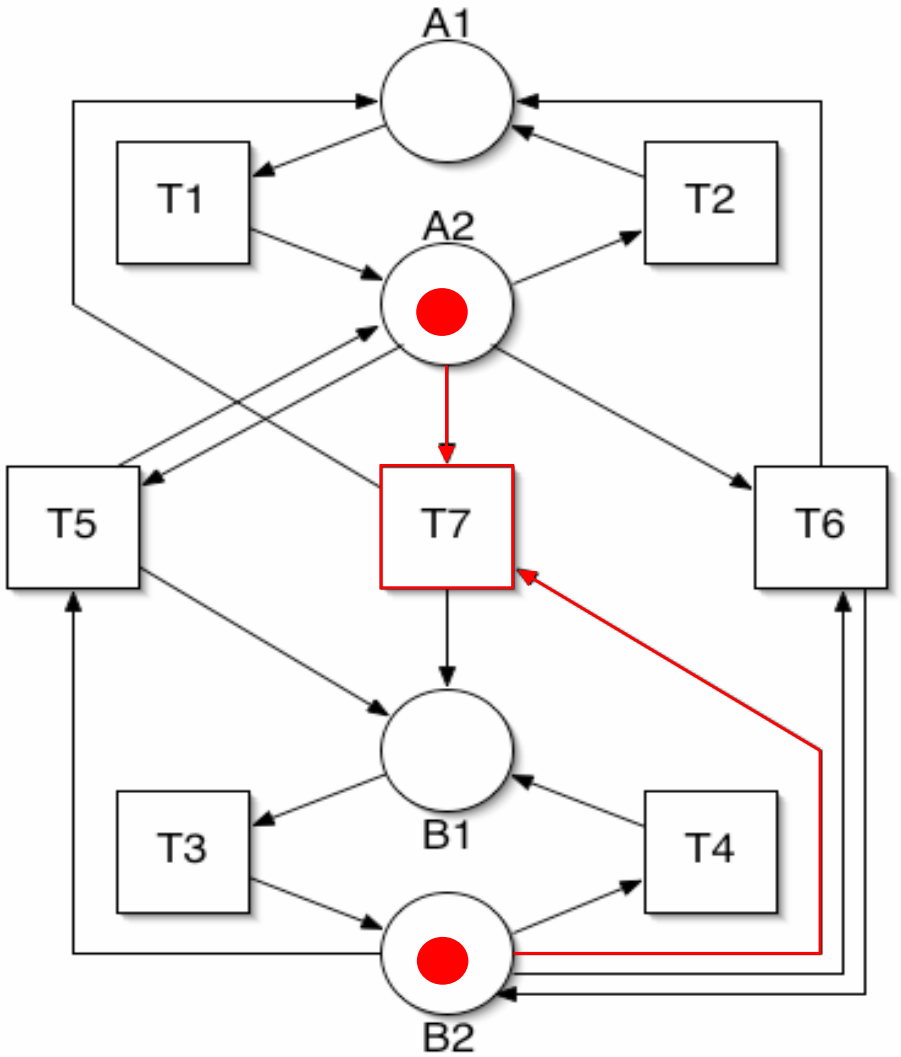
Framework for Motion Coordination

- C/E Petri Net for each subsystem
  - Head
  - Left hand
  - right hand
  - Left arm
  - Right arm
  - Torso
  - Mobile platform
- Conditions (places)
  - S1: Subsystem is active
  - S2: Subsystem is ready
- Events (Transitions)
  - T1: Task is completed
  - T2: Task triggered



- The shown initial marking indicates the state *ready*
- The task execution can be invoked by firing the transition T2 which leads to the state *active*

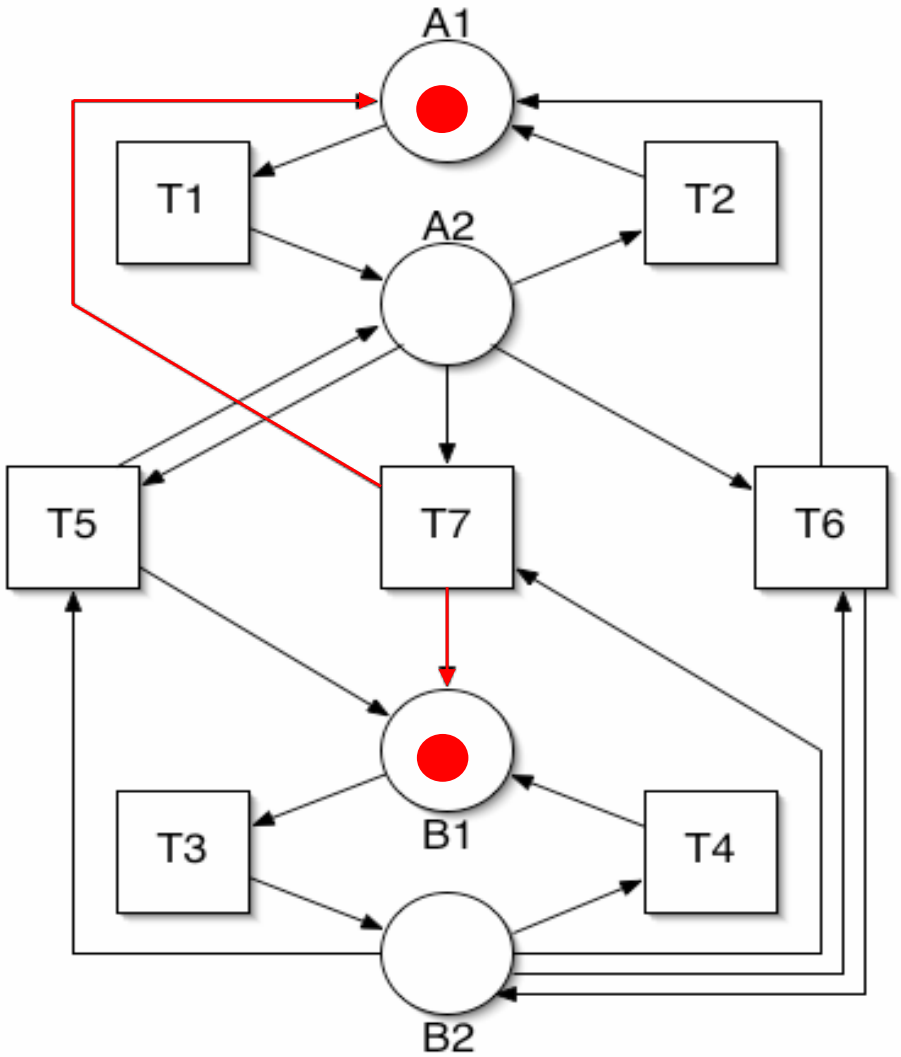
Task coordination for 2 Subsystems



Example for coordinated action:

- A2, B2 marked (both systems ready)
- Events associated with T7 trigger firing of T7
- A1, B1 marked (both systems active)
- Both systems are ready after finishing execution

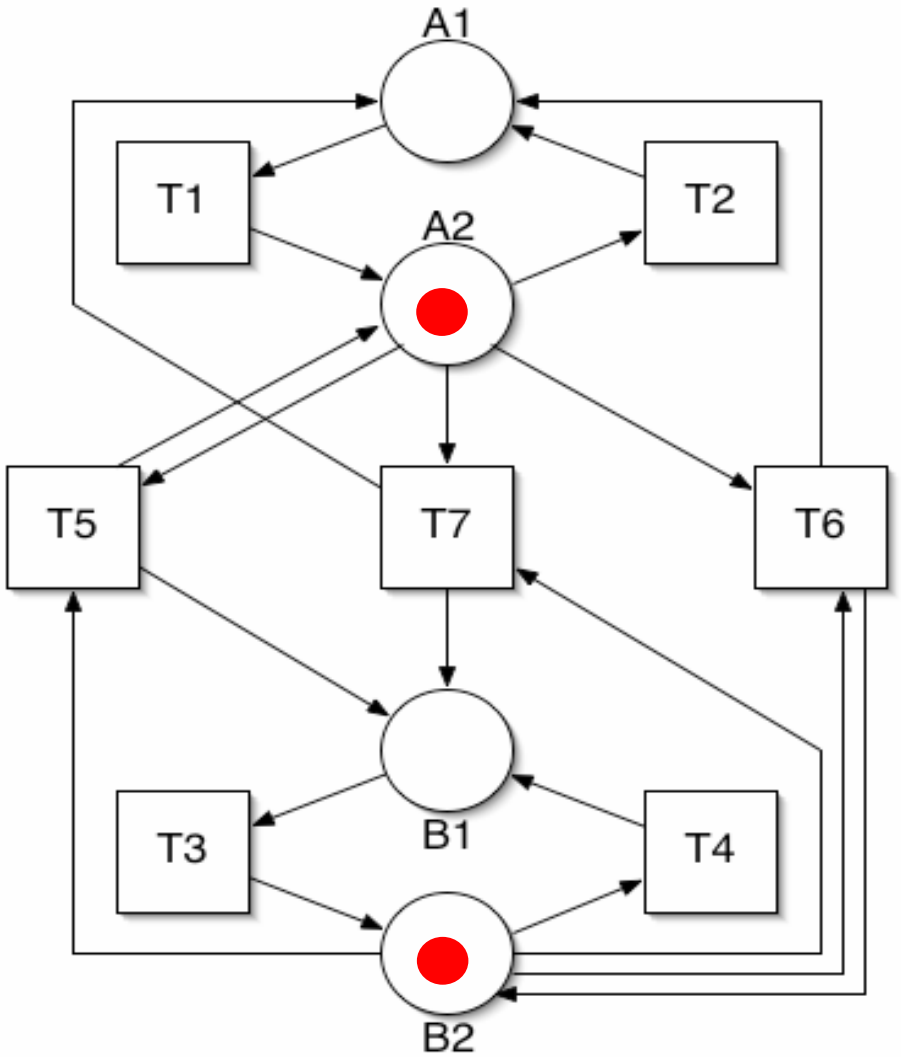
Task coordination for 2 Subsystems



Example for coordinated action:

- A2, B2 marked (both systems ready)
- Events associated with T7 trigger firing of T7
- A1, B1 marked (both systems active)
- Both systems are ready after finishing execution

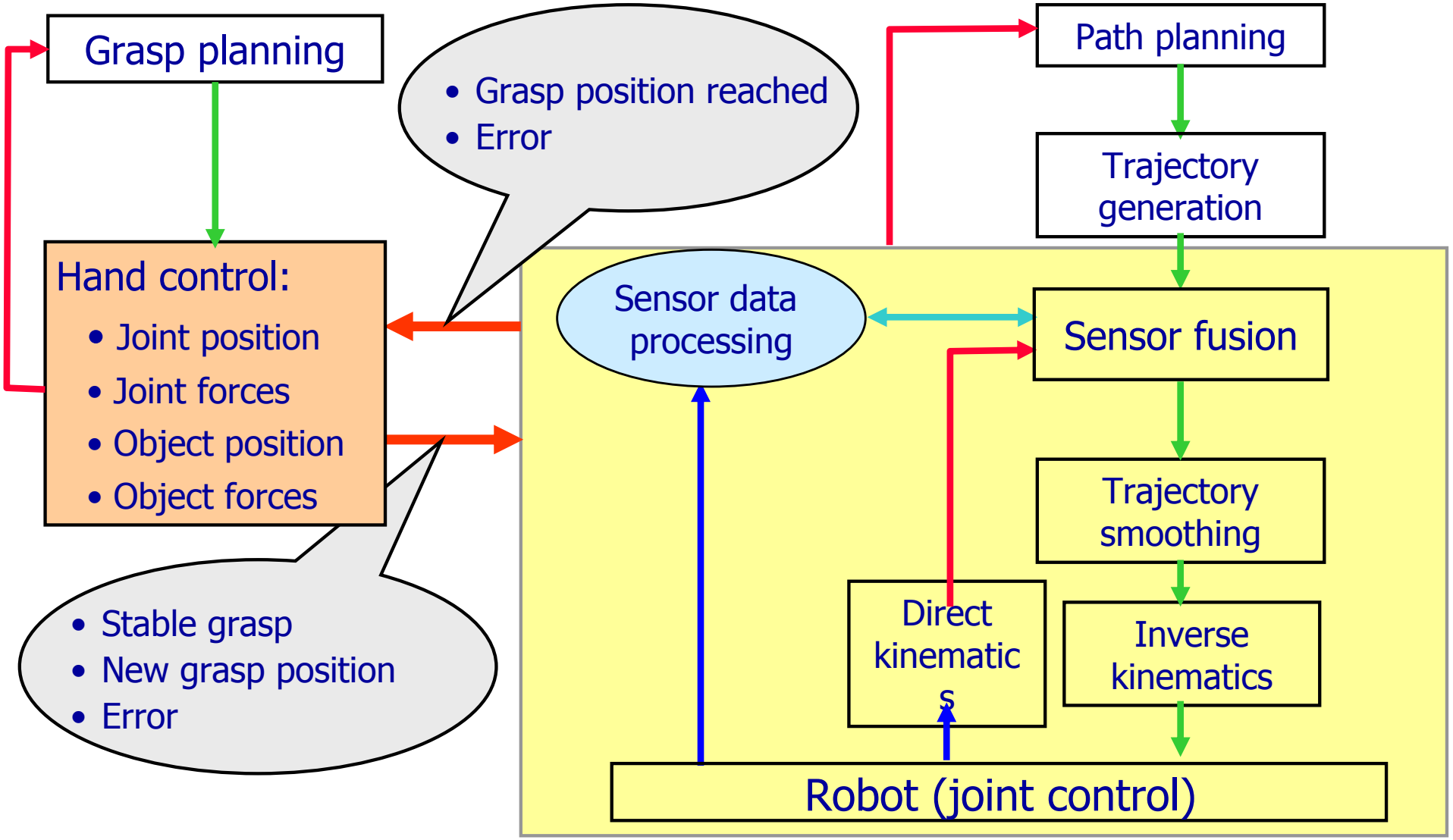
Task coordination for 2 Subsystems



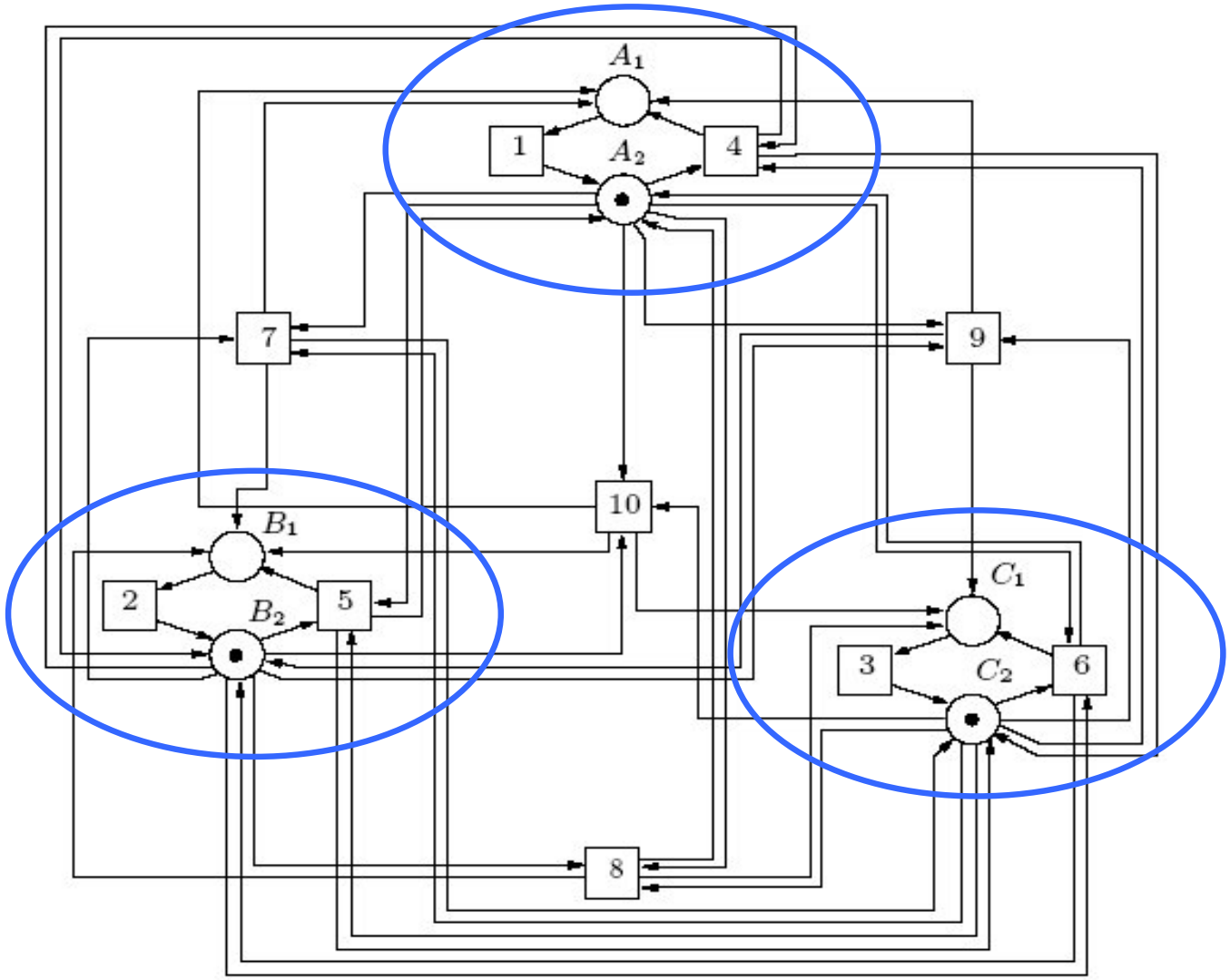
Example for coordinated action:

- A2, B2 marked (both systems ready)
- Events associated with T7 trigger firing of T7
- A1, B1 marked (both systems active)
- Both systems are ready after finishing execution

Example: Events Hand-Arm-Koordination

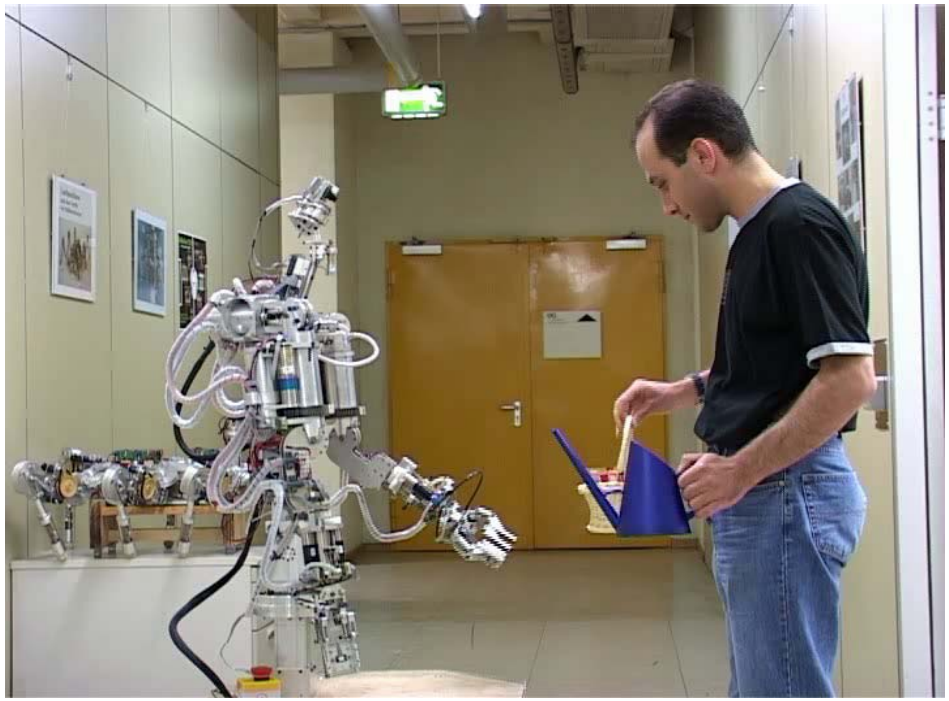


Task coordination for 3 Subsystems



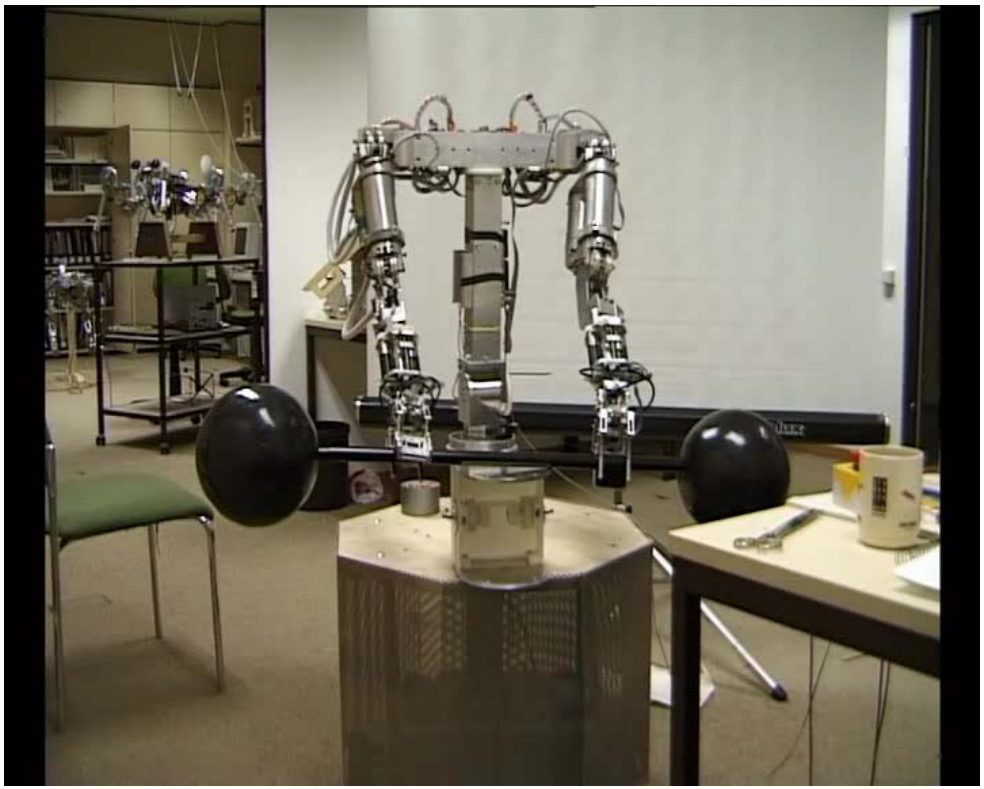
Execution of manipulation tasks

Coordination: head/arm/arm

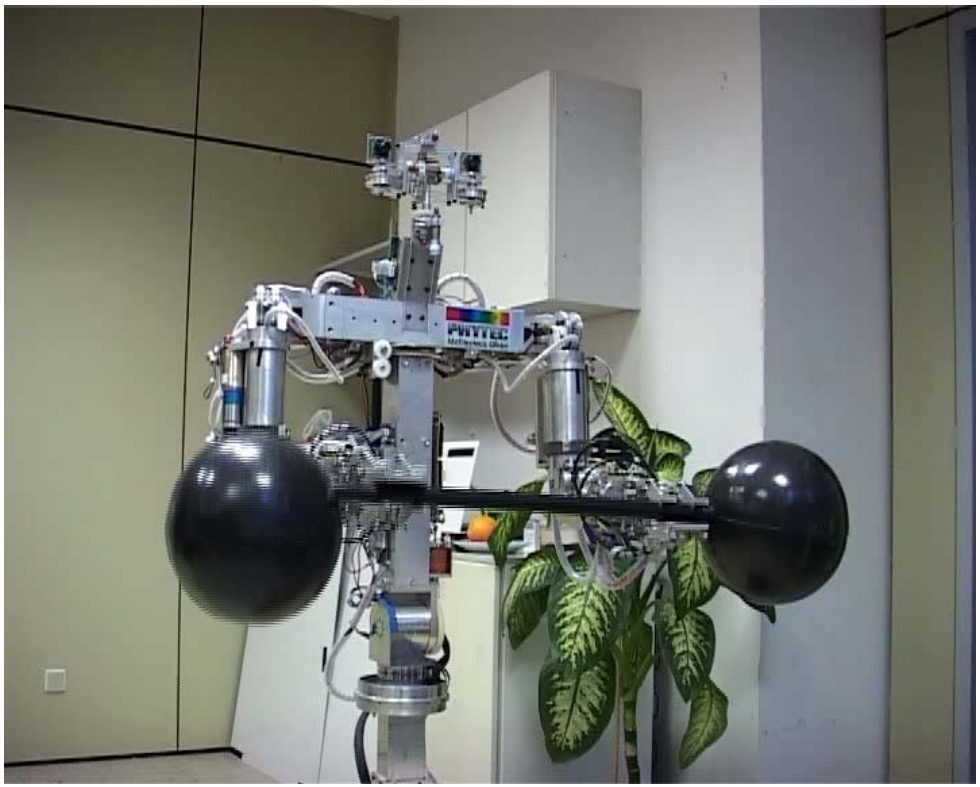


Coordination: head/arm/platform

Execution of manipulation tasks



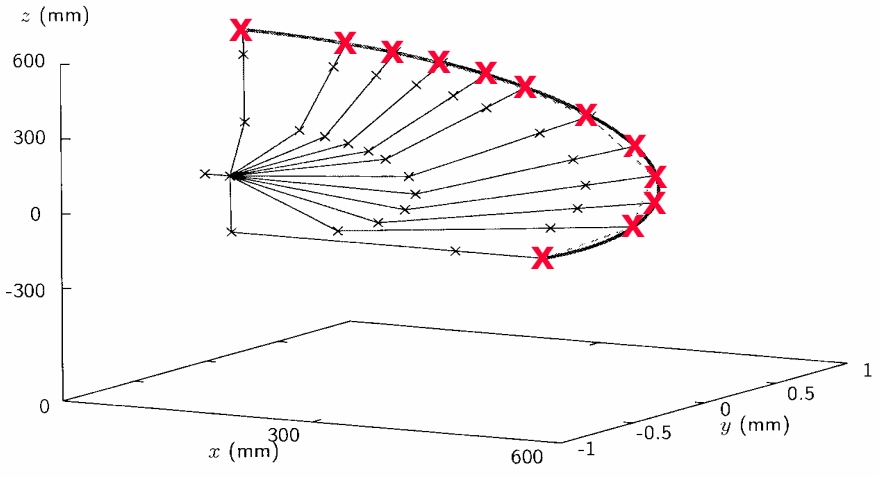
dual arm coordination



dual arm coordination

Tamim Asfour
Universität Karlsruhe

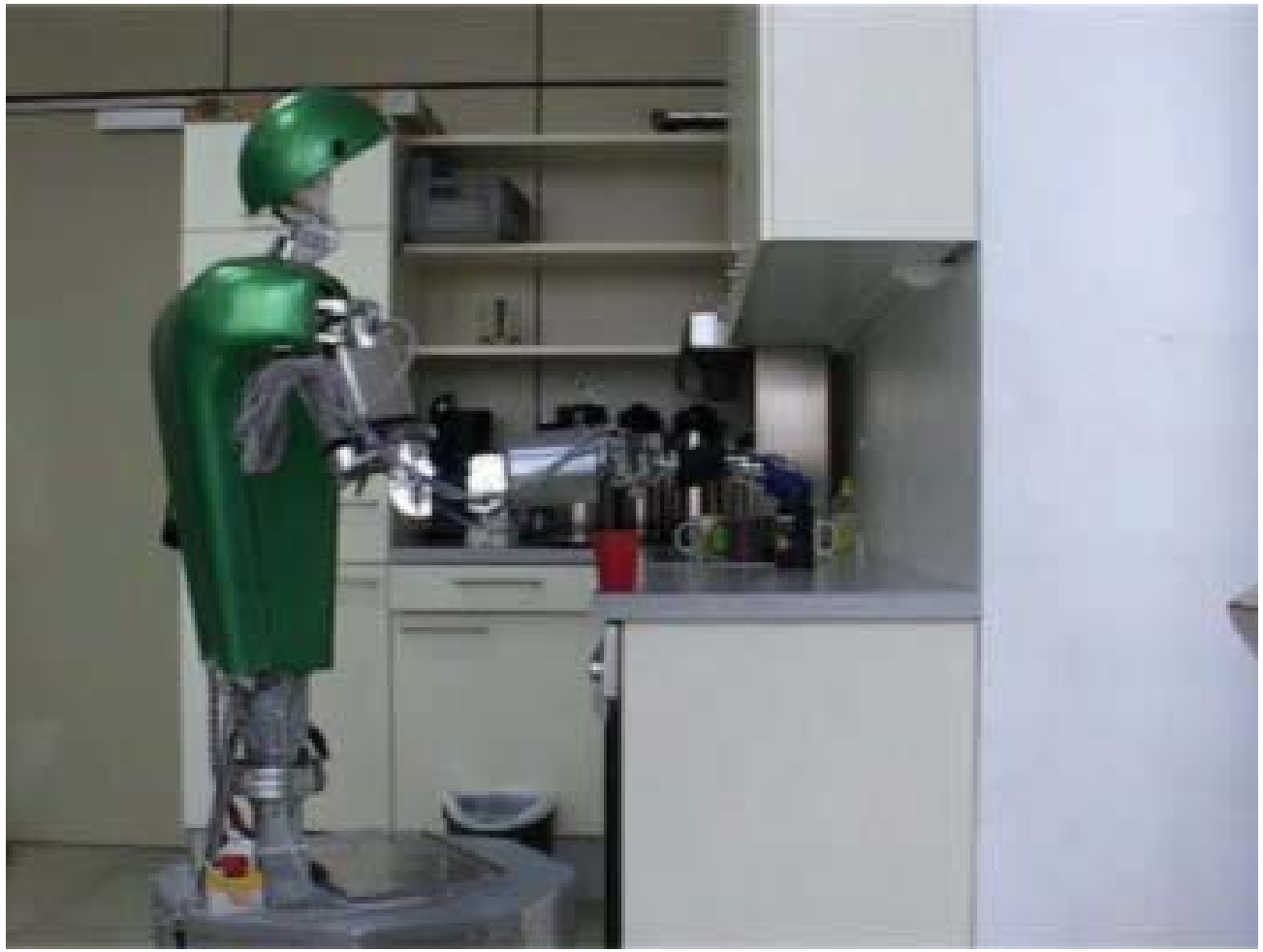
2-Arm Coordination



x: Synchronisationspunkt

First experiments in the kitchen

- Manipulation of coloured objects with 5-finger hand
- Cup loading into the dishwasher



Human-Robot-Dialogues

- Audio tracking of sound sources
  - work done by Prof. Kroschel
  - http://www.int.uni-karlsruhe.de/~kroschel/
- Visual tracking of faces
- Audio-Visual Person Tracking
- Speech recognition for continuous speech (german/ english)
  - work done in the group of Prof. A. Waibel, Karlsruhe/CMU http://isl.ira.uka.de/

Modular Controller Architecture MCA2

Dr.-Ing. Kay-Ulrich Scholl

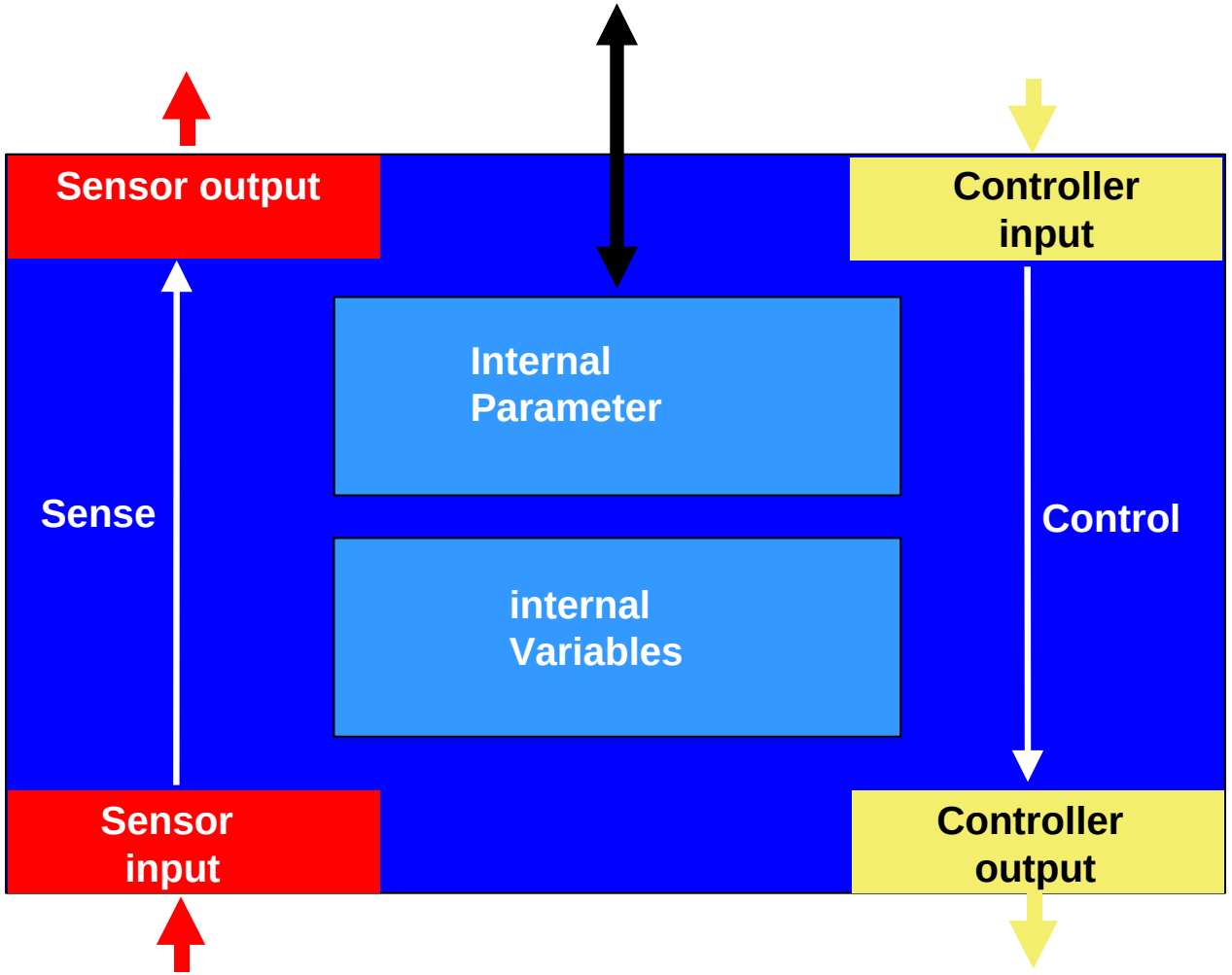
www.fzi.de/ids

www.mca2.org

Universität Karlsruhe

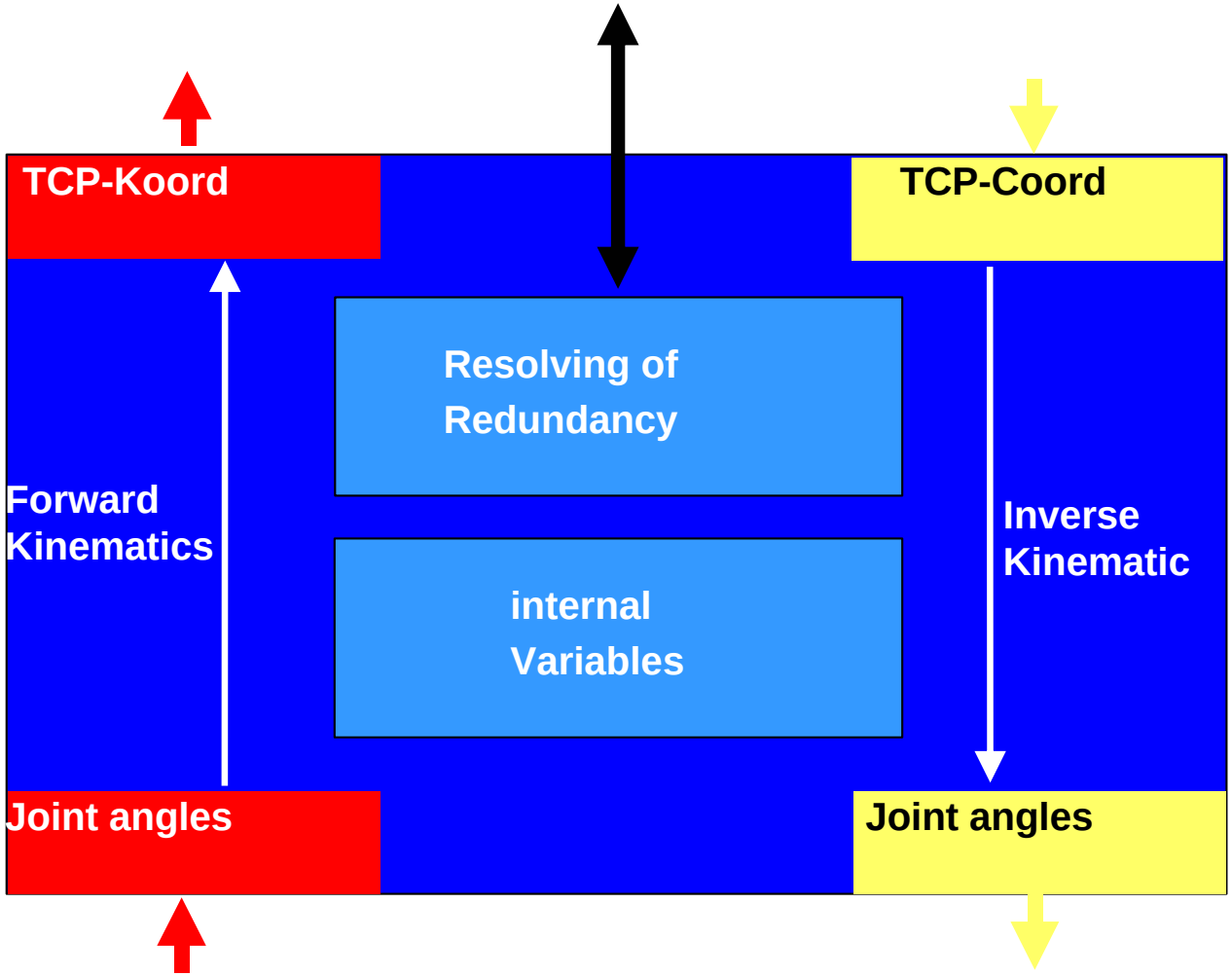
Tamim Asfour

MCA2: Module


Sensor output
Controller input
Internal Parameter
Sense
Control
internal Variables
Sensor input
Controller output

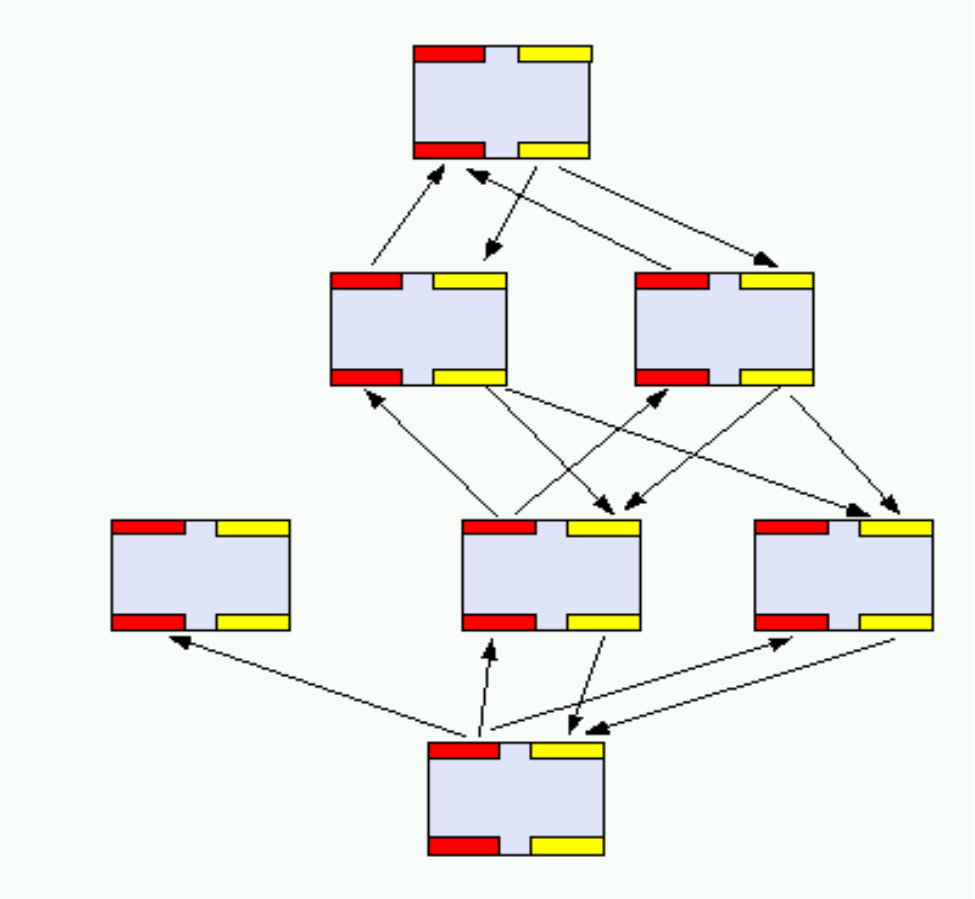
MCA2: Modules

Example: Kinematics



Edges: Communication

- Connect modules via edges:
  - Sense – Sense
  - Control – Control
- Copying of values
  - whole vector
  - part of a vector
  - permutate values



Modulegroups

Execution (by level):

Sense: from bottom to top

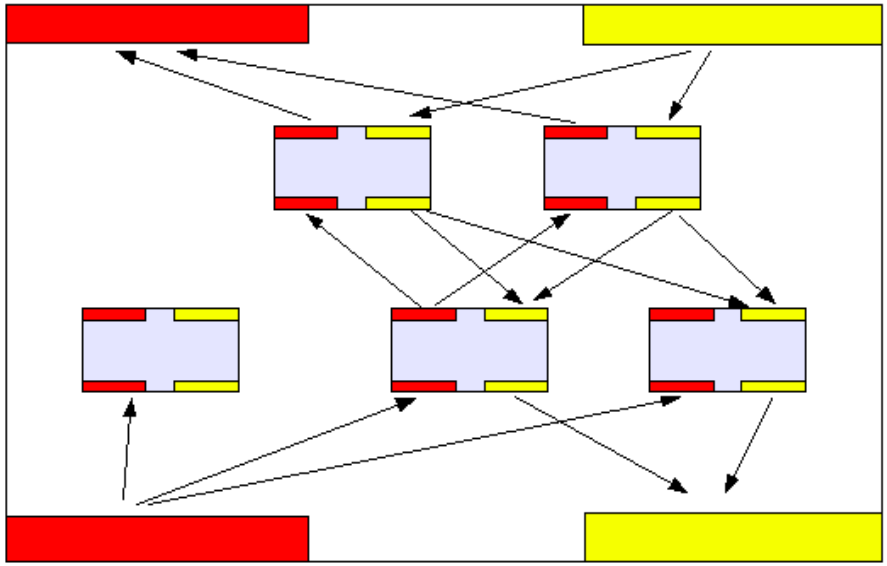
all edges

all modules: Sense

Control: from top to bottom

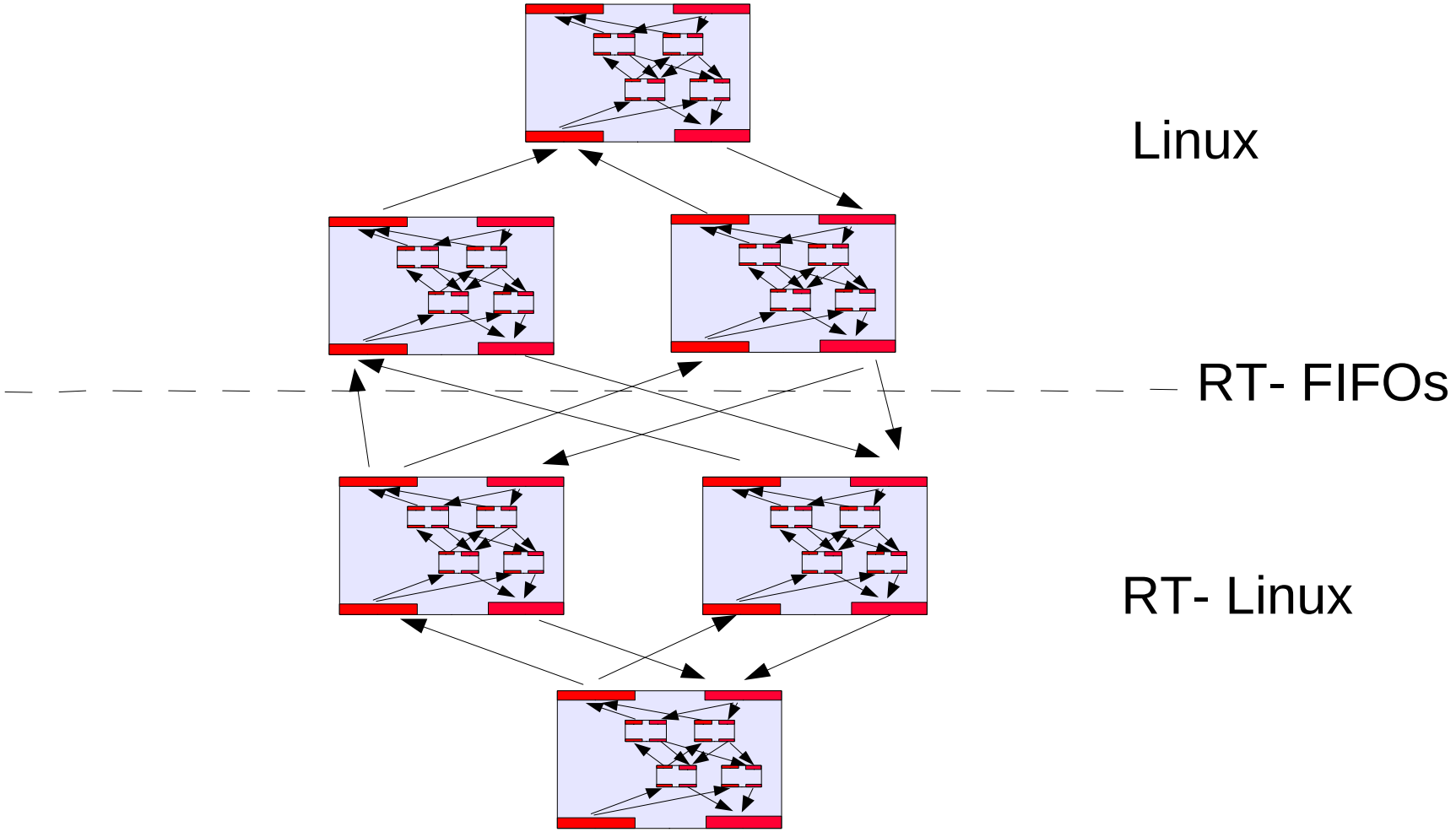
all edges

all modules: Control



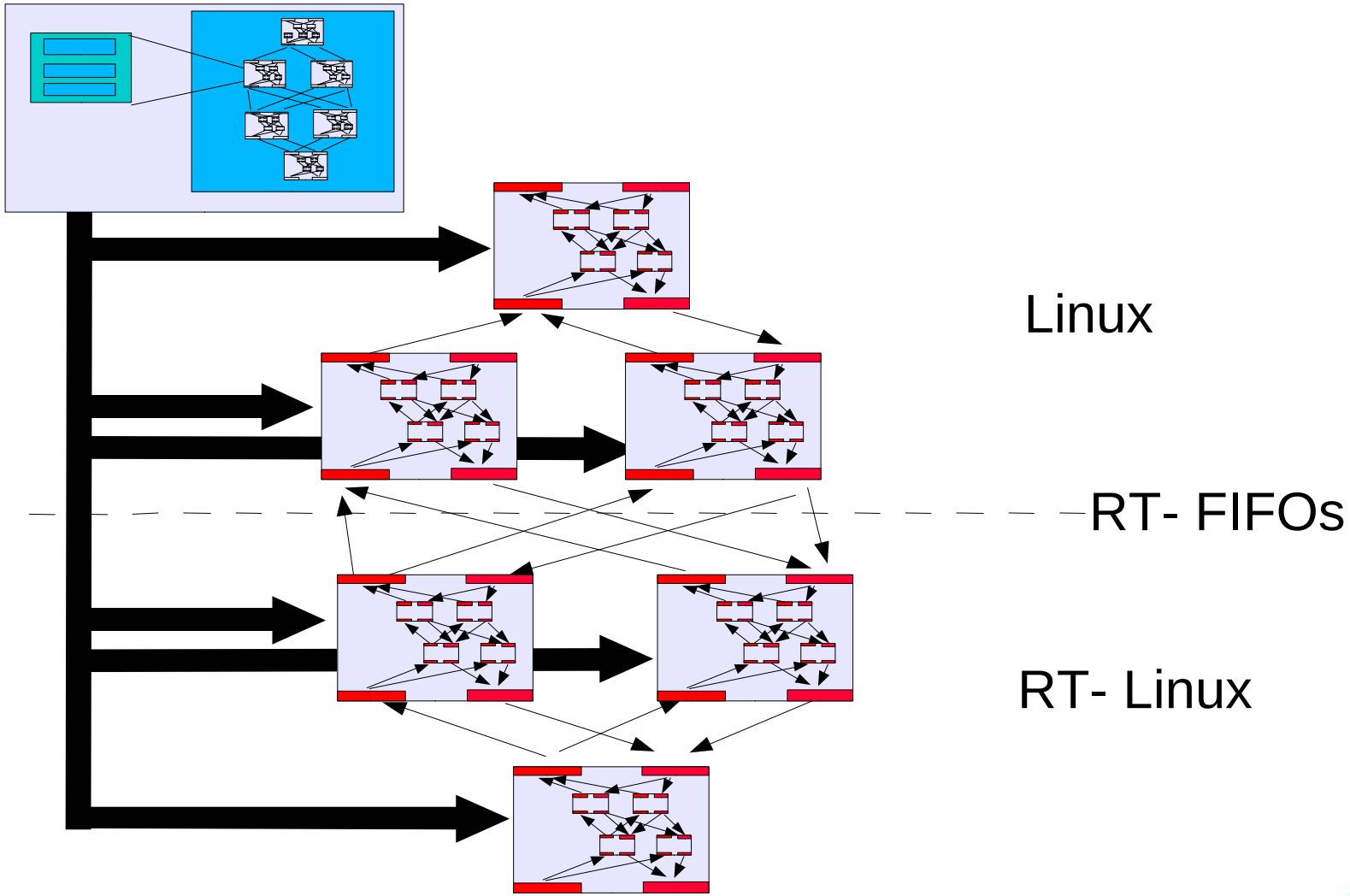
Additional interface for accessing modules and edyes by external tools

Complete architecture

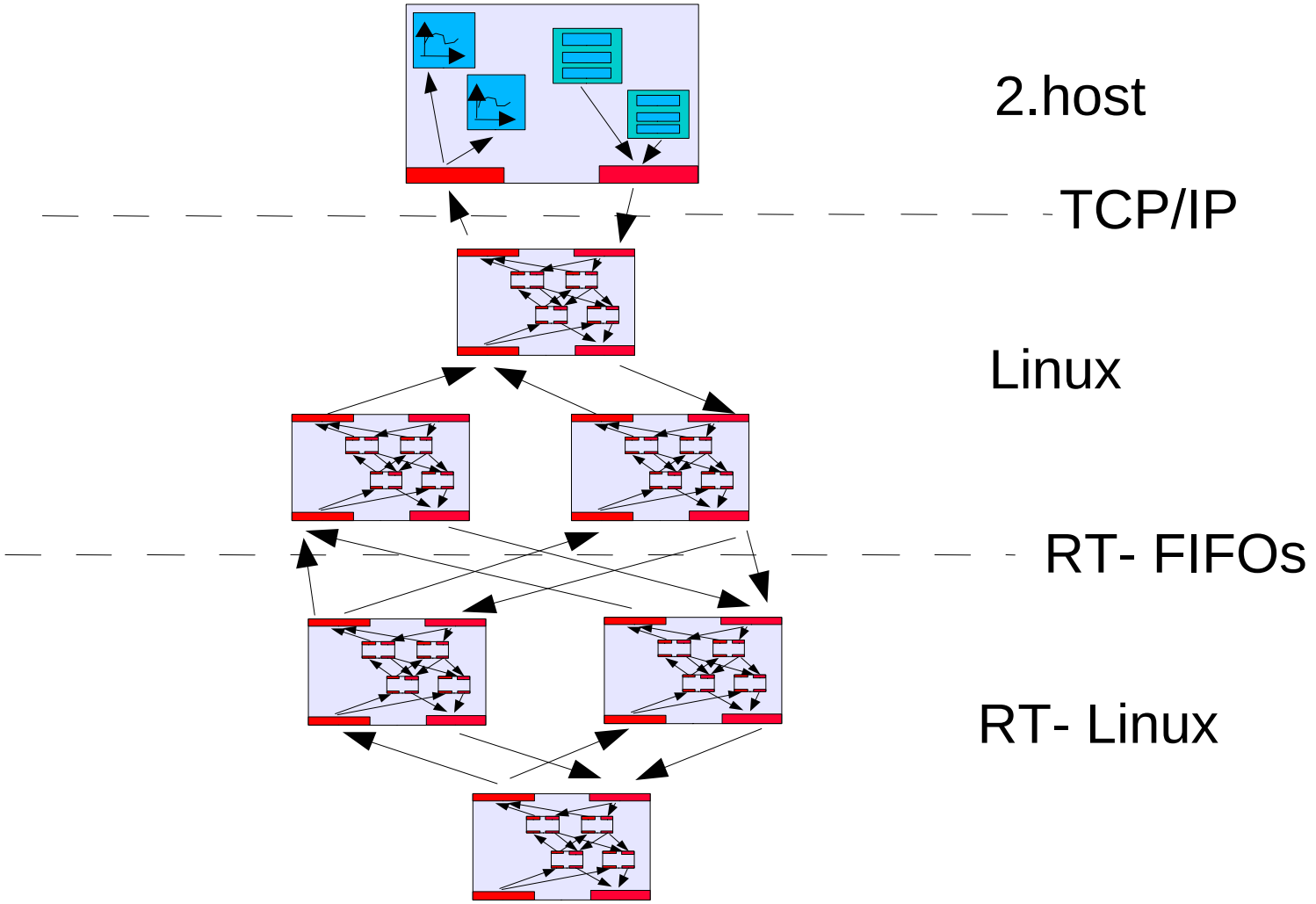


Tamim Asfour
Universität Karlsruhe

Tools: MCAbrowser



Tools: MCAGUI



MCAGUI

- Input of control values
  (Buttons, sliders, text input, selectors)
- Display of sensor values
  (LCD, LED, oscilloscope, 2D/3D- Views)
- Fast creation of complex graphical user interfaces
- Individual development of input/output widgets by creation of plugins

Tamim Asfour
Universität Karlsruhe

Additional features

- TaskContainer

  - Execution of modules/groups in separate threads with different period and priority

  -synchronisation if input/output data (non blocking)

- Blackboards

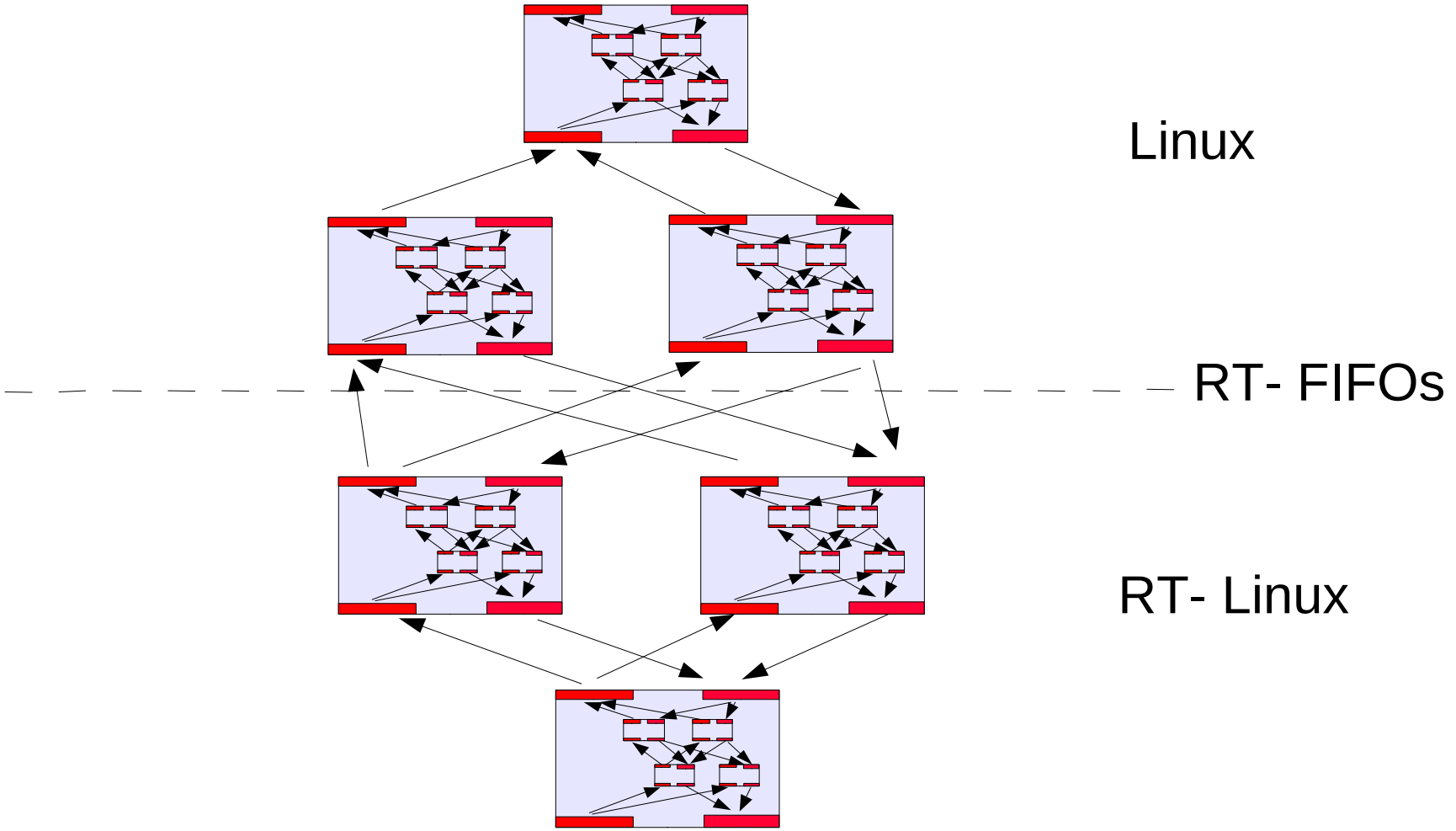
  - network transparent memory

  - Locking/unlocking mechanisms

- NestedRemoteModules

  - integration of external hardware/software components as virtual parts

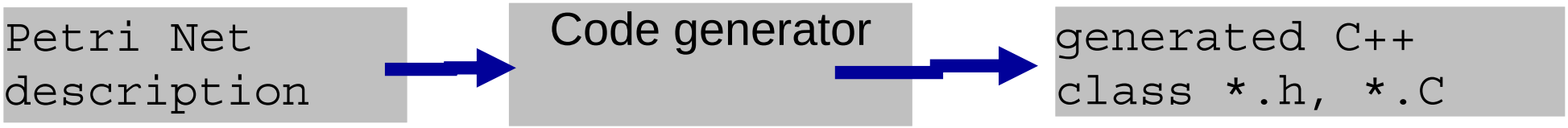
MCA2: Gesamtarchitektur



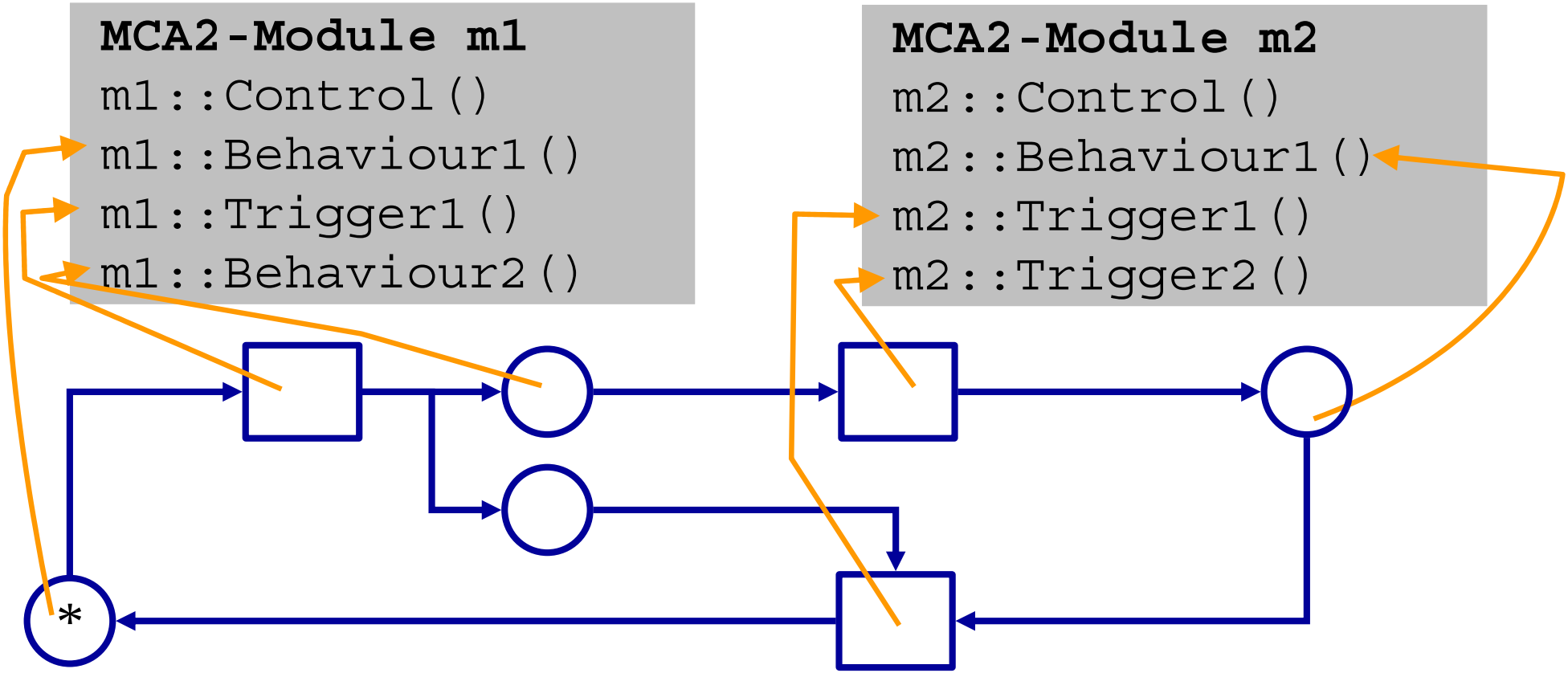
Tamim Asfour
Universität Karlsruhe

mcapetri – Petri Nets in MCA2

- Petri Nets for the coordination/synchronisation of task execution
  - Simple graphical notation
- Augmented Petri Nets with Callback functions
  - Code generator → Petri Net Implementation
  - Explicit separation between functionality & control scheme
  - Petri Nets: What to do? and When?
  - MCA2 modules: How to do?
- mcapetri: http://wwwipr.ira.uka.de/~osswald/tools.html



mcapetri – Petri Nets in MCA2



Petri Net: What to do? and When?
MCA2 modules: How to do?

Live Demo of ARMAR-II Software in MCA

- in the active Siesta ...

Tamim Asfour
Universität Karlsruhe

Imitation Learning of Human Arm Movements

- Teaching a robot can be done in a number of ways
  - robot programming language
  - simulation-based graphical programming interface
  - “teaching by guiding”, where the instructor operates a robot manipulator while its motion is recorded. The recorded motions are then added to the robot’s action repertoire.
  - Such techniques are well-suited for industrial robots
- Domain of humanoid robots
  - unexperienced users,
  - such methods are complex, inflexible
  - special programming skills or costly hardware needed

How to teach a robot?

- **Imitation Learning:** An approach that addresses both issues human-like motion and easy teaching of new tasks
- Imitation learning is the concept of having a robot observe a human instructor performing a task and imitating it when needed.
- Robot learning by imitation, also referred to as *programming by demonstration* has been dealt with in the literature as a promising way to teach humanoid robots.
- Several imitation learning systems and architectures based on the perception and analysis of human demonstrations have been proposed

Imitation architectures

- In most architectures, the imitation process proceeds through three stages: perception/analysis, recognition and reproduction
  - Y. Demiris and G. Hayes, "Imitation as a dual-route process featuring predictive learning components: a biologically-plausible computational model," *Imitation in animals and artifacts*, pp. 327–361, 2002.
- Basic ideas of imitation learning in robots as well as humans
  - S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in Cognitive Sciences*, vol. 3, no. 6, pp. 233–242, 1999.
- Imitation on robots
  - S. Calinon and A. Billard, "Stochastic gesture production and recognition model for a humanoid robot," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, 2005, pp. 2769–2774.

“passive” and “active” imitation

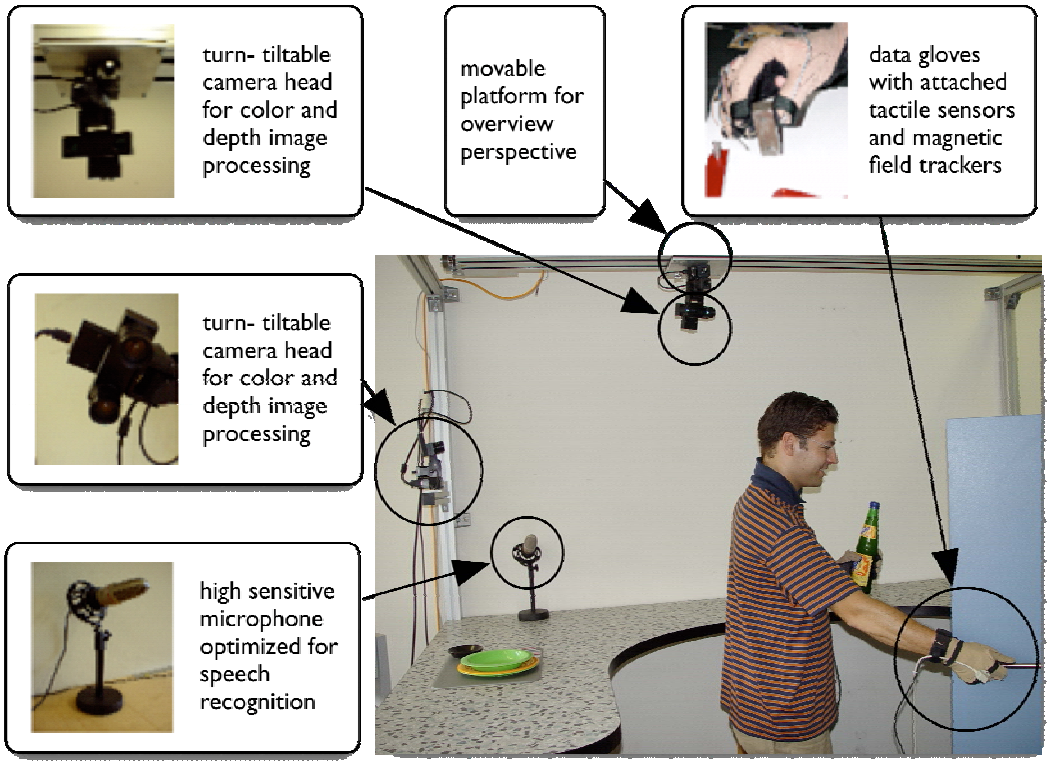
- **Passive imitation:**
  - Imitator goes through a “perceive - recognise – reproduce” cycle (passive imitation)
  - the motor systems are involved only during the “reproduce” phase
- **Active Imitation:**
  - the imitator's motor systems are actively involved even during the perception process

Our approach

- Approach to imitate "dual arm" movements shown to a robot multiple times
  - Human arm motion capturing,
  - Identify characteristic features in the trajectories that can be observed in all (or many) demonstrations,
  - Consider only those when reproducing a movement,
  - Show how to use those common features to detect possible temporal interdependencies between both arms in dual-arm tasks
- Framework:
  - Use HMMs to generalize movements
  - Training the HHMs with key points (KP) of each demonstration
  - Identify common key points (CKP)
  - Movement representation through the resulting CKPs

Human Motion Capturing (magnetic tracker)

- Magnetic tracking system (Fasttrak, www.polhemus.com)



- Output: Position and orientation of the wrist

Human Motion Capturing (Stereo vision)

- Capture 3d human motion based on the image input from the cameras of the robot‘s head only
- Approach:
  - Particle Filter framework
  - Localization of hands and head using color segmentation and stereo triangulation
  - Fusion of 3d positions and edge information
- Features
  - Automatic Initialization
  - Close to real-time: 15 fps on a 3 GHz CPU, images captured at 25 Hz
  - Smooth tracking of real 3d motion
- Output: Joint angles

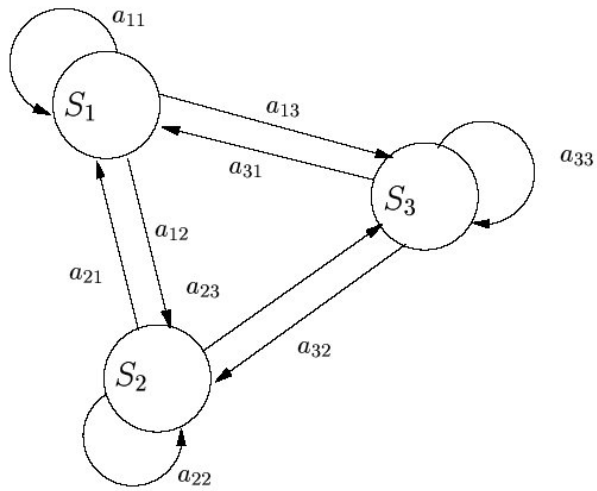
Work done by Pedram Azad [Humanoids 2004]

HMM structure

- We use three different HMM for each arm,
  - λ_p for the position of the TCP (hand path) with the Cartesian coordinates being represented by three-dimensional output distributions
  - λ_o for the orientation of the TCP (described by three angles)
  - λ_j for the joint angle trajectories where the dimension of the output distributions is equal to the number of observed joint angles.

Formal Definition of Hidden Markov Models

A Hidden Markov Model is a five-tuple consisting of:



S — The set of **States** $S=\{s_1,s_2,\ldots,s_n\}$

π — The **initial probability** distribution
$\pi(s_i)$ = probabilty of s_i being the first state of

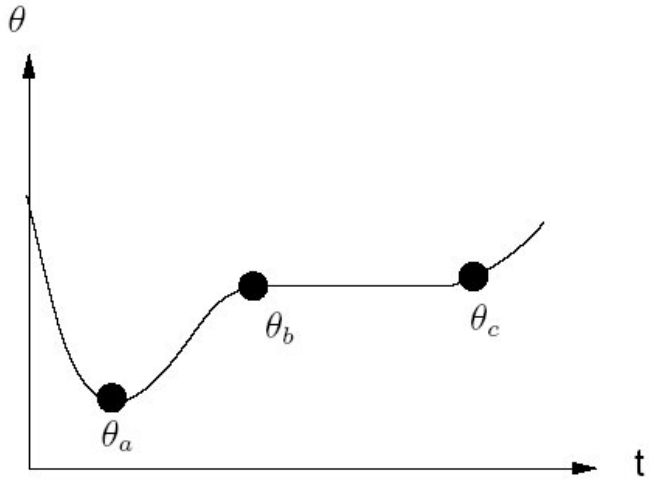
A — The matrix of **state transition probabilities**:
$A=(a_{ij})$ where a_{ij} is the probability of state s_j following s_i

B — The set of **emission probability** distributions/densities,
$B=\{b_1,b_2,\ldots,b_n\}$ where $b_i(x)$ is the probabiltiy of observing x when the system is in state s_i

V — The observable **feature space** can be discrete:
$V=\{x_1,x_2,\ldots,x_v\}$, or continuous $V=\mathbf{R}^d$

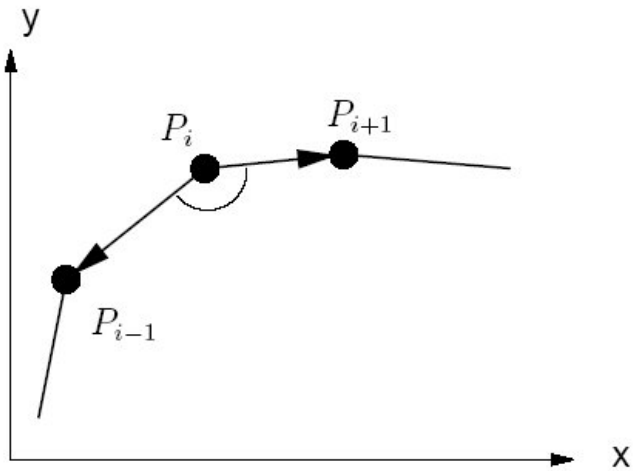
Detection of key points

- Joint trajectory
  - reaches maximum or minimum(i.e. direction changes) or
  - stops changing,
  - sufficient time has passed since the creation of the previous keypoint



- Orientation:
  - same criteria

- Position:
  - P_i is a key point, if the angle between the vector that goes from P_i to P_{i-1} and the vector that goes from P_i to its P_{i+1} is less than ±180



$$\angle\left(\vec{P}_i - \vec{P}_{i-1}, \vec{P}_i - \vec{P}_{i+1}\right) < 180° - \varepsilon$$

Training of the HHMs

- HMM for different kinds of observations (joint angles, TCP position, TCP orientation)
- Each HMM is trained with the key points of all demonstrations using the Baum-Welch algorithm for multiple observations.
- Each training sequence consists of the key points of the respective demonstration.
- For a given observation sequence, the Viterbi algorithm returns the optimal state sequence of the HMM with respect to that observation sequence, i.e. the sequence of states most likely to generate that observation sequence.

Which states (KPs) should be used for the reproduction?

- key points of a movement that are common to all (or almost all) demonstrations "common key points, CKP"

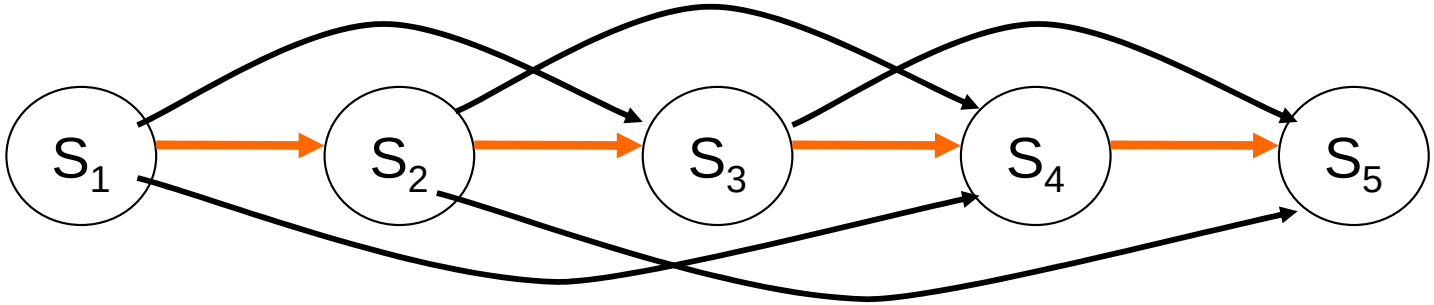
- How to know what key point in one demonstration corresponds to some key point in another demonstration?

- We use HMM to match key points across demonstrations.

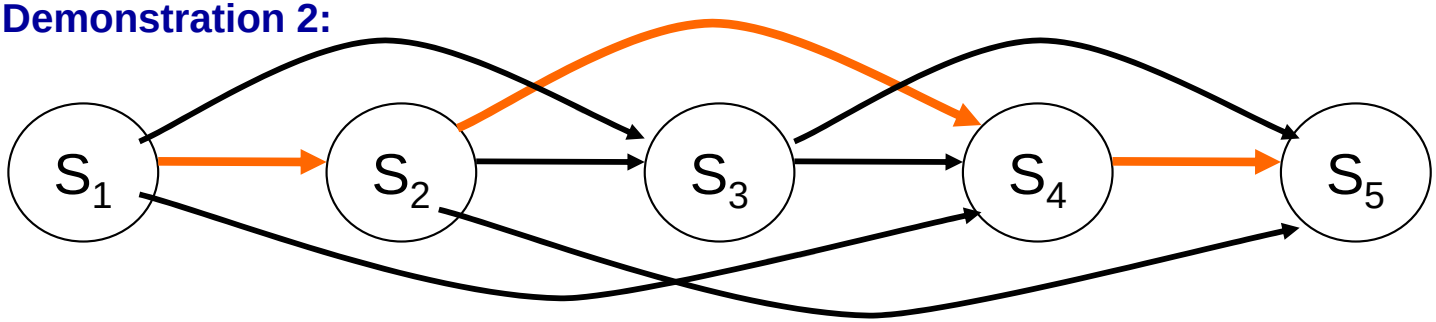
  → Use only those states of the HMM for the reproduction that represent key points which are shared by all (or almost all) demonstrations.

Common Key Points (Beispiel)

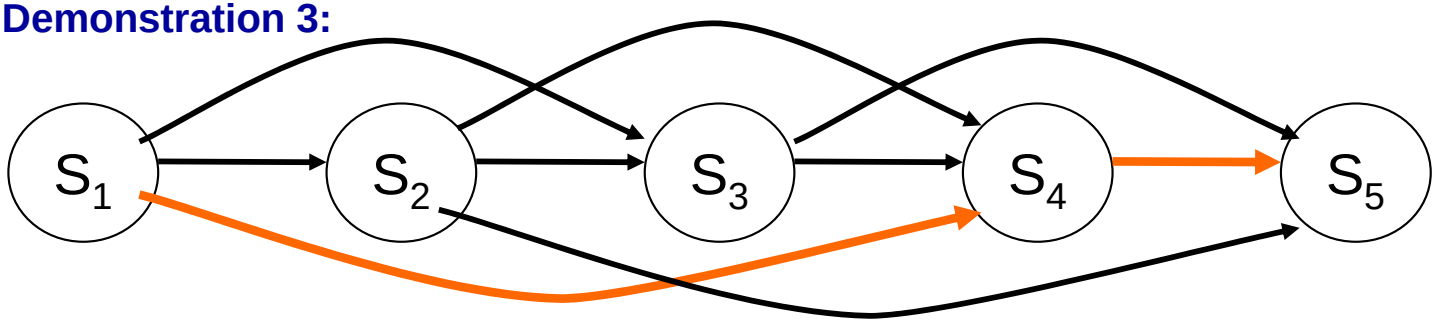
Demonstration 1:



Demonstration 2:



Demonstration 3:



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

$C_1 = (1,4,5)$
$C_2 = (1,3,4)$
$C_3 = (1,2,3)$

Generating a generalized movement

- Consider only those states of the HMM that correspond to common key points;
- Take the means of the output density functions (PDFs) of those states (in order) → a sequence of positions, orientations and joint angles that can be used to reproduce the movement.

But

- What timestamps should be associated with each of those common key points?
  - Average of the timestamps of all the key points that constitute a common key point.
- Positions and orientations are of no use to reproduce a movement on a robot or a software model of the human body
  - transform to joint angles (inverse kinematics algorithms)

Generalisation / Reproduction

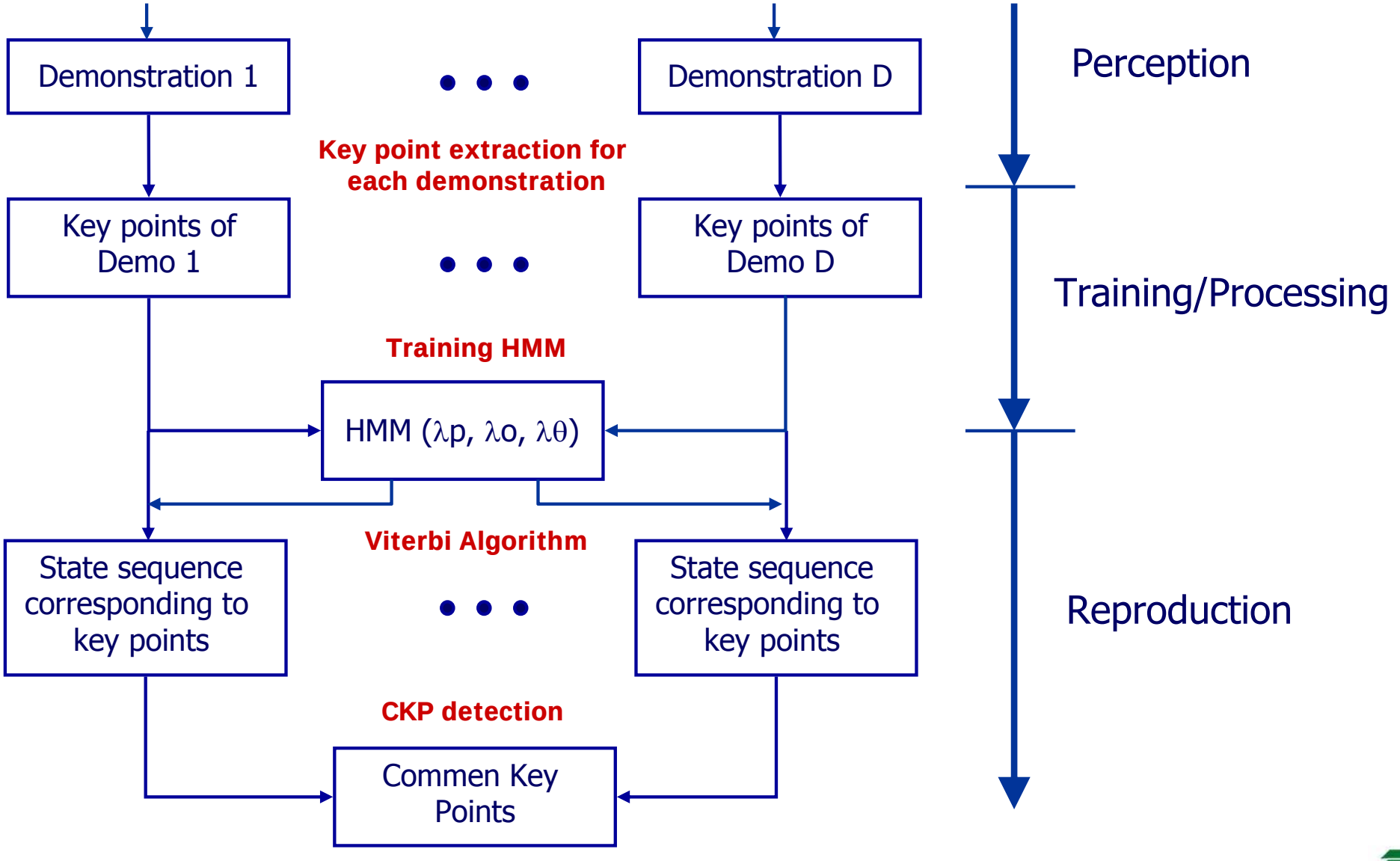
- Interpolation between common key points (position-, orientierungs- and joints-CKP)

- Inverse Kinematics → joint angles sequences
  - The resulting joint angle sequences are different from the joint angle sequences obtained directly from the joint angle HMM.
  - A weighting factor $\omega \in [0, 1]$ is used to determine the relative influence of the two sequences on the joint angles to be executed by the robot:

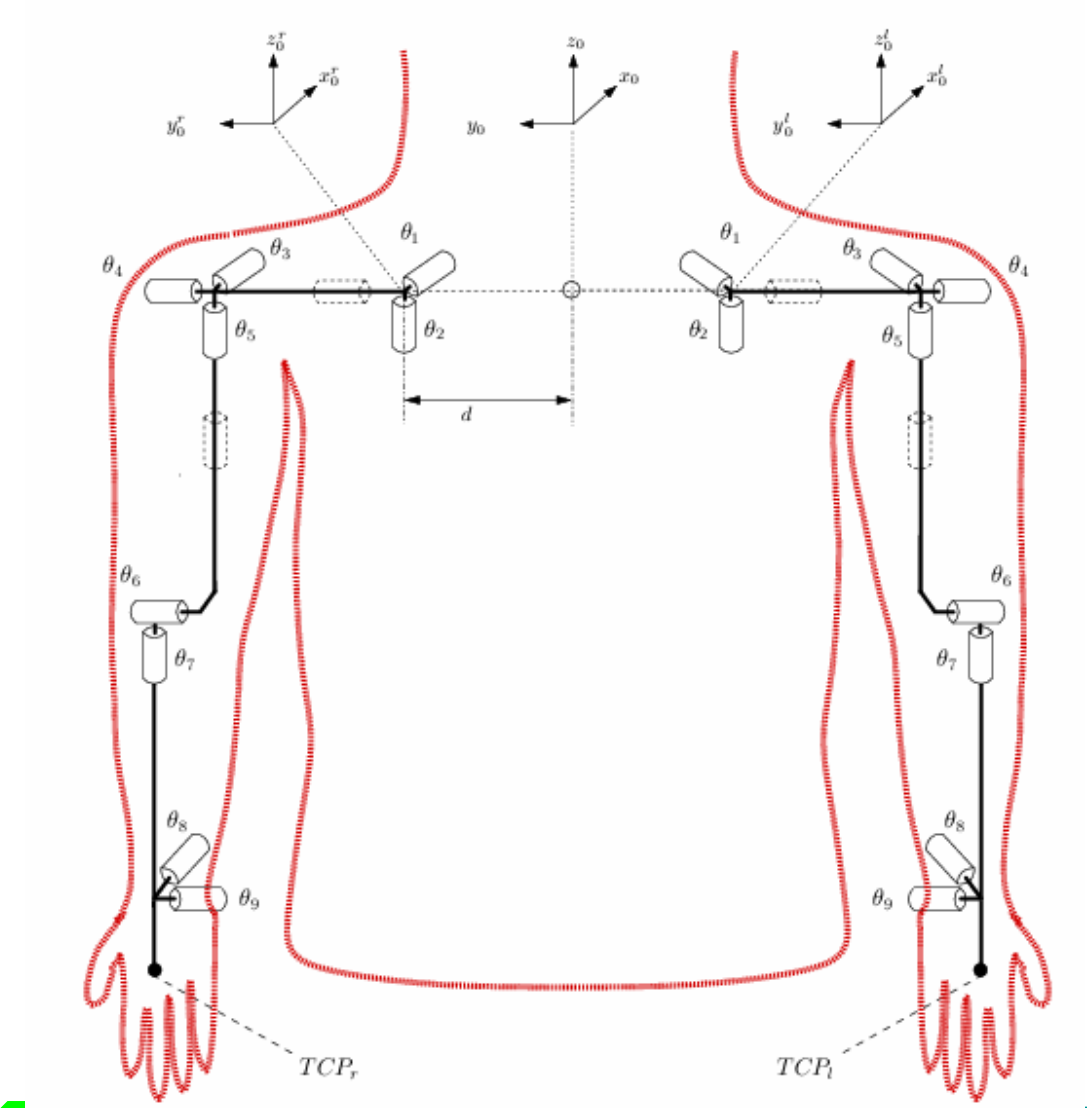
$$\tilde{\theta}_i^j = \omega \cdot \hat{\theta}_i^j + (1-\omega) \cdot \check{\theta}_i^j \qquad \omega \in [0,1]$$

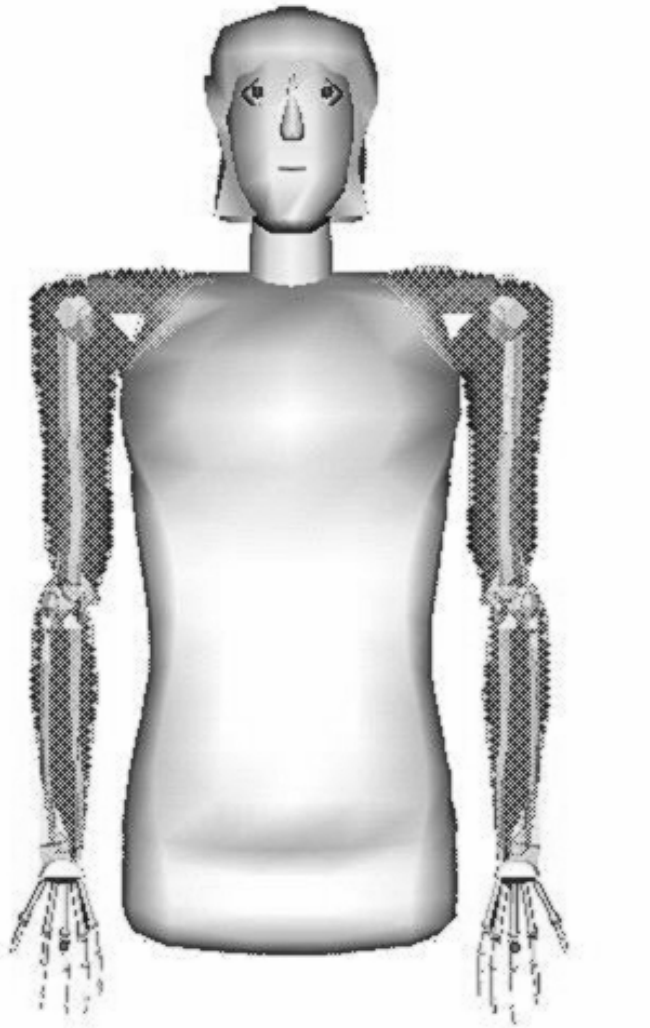
HMM IK

Approach: from perception to reproduction

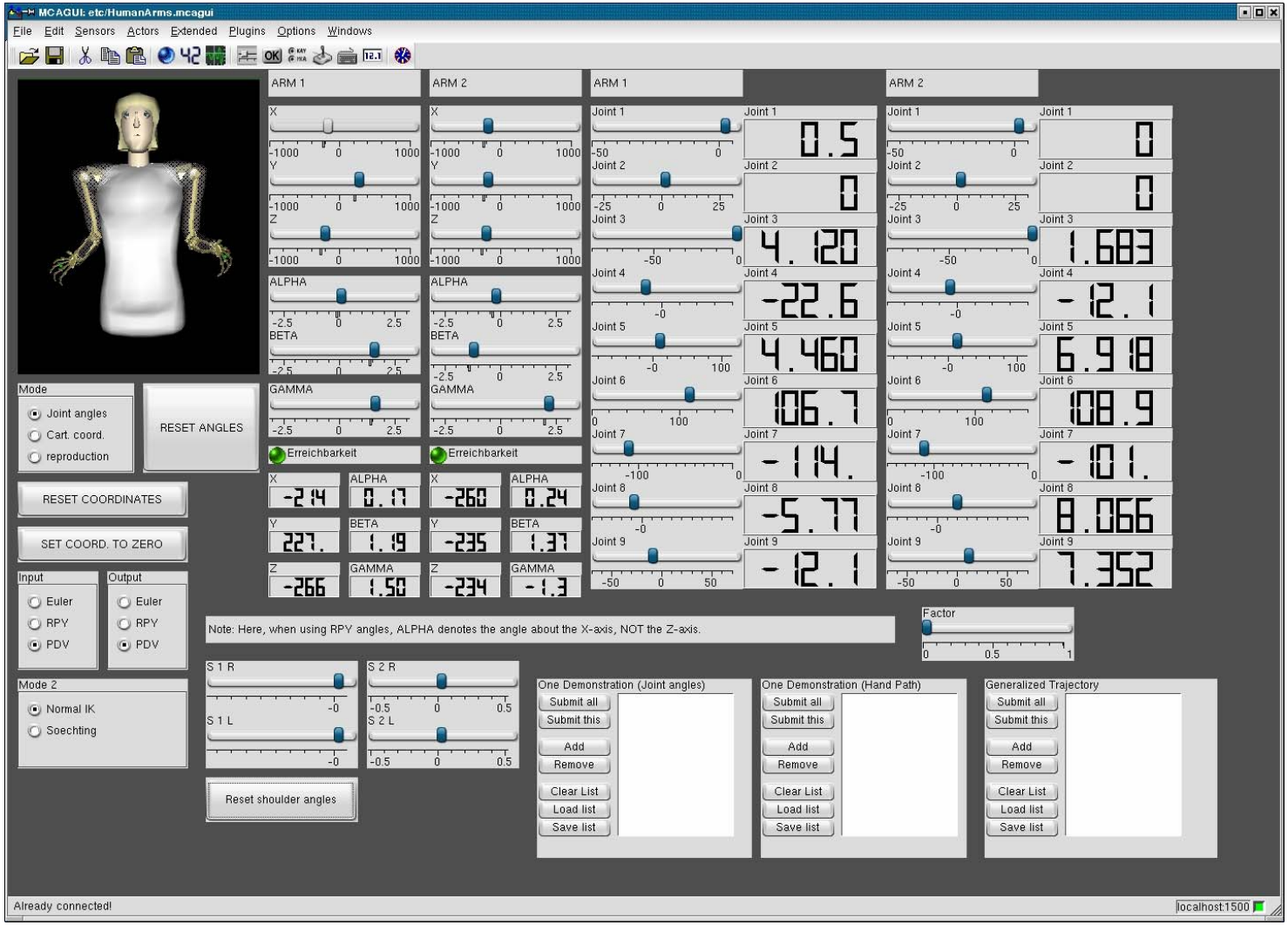


Reproduction on a human kinematics model





Simulation in MCA2

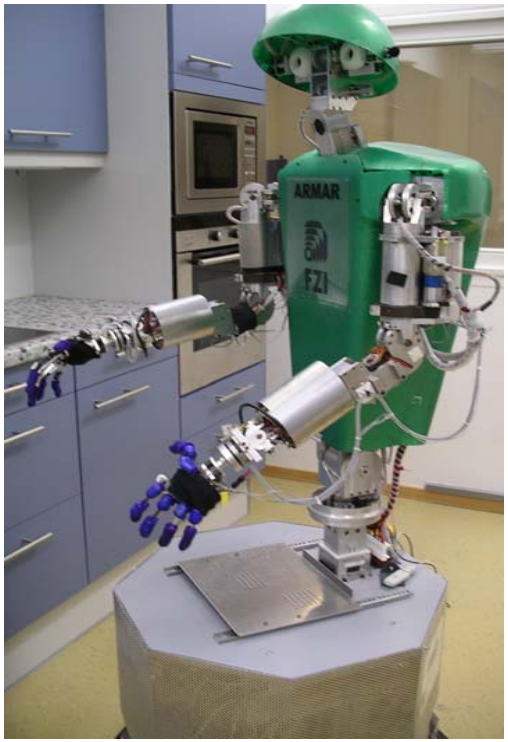


Live-Demo in the active Siesta!

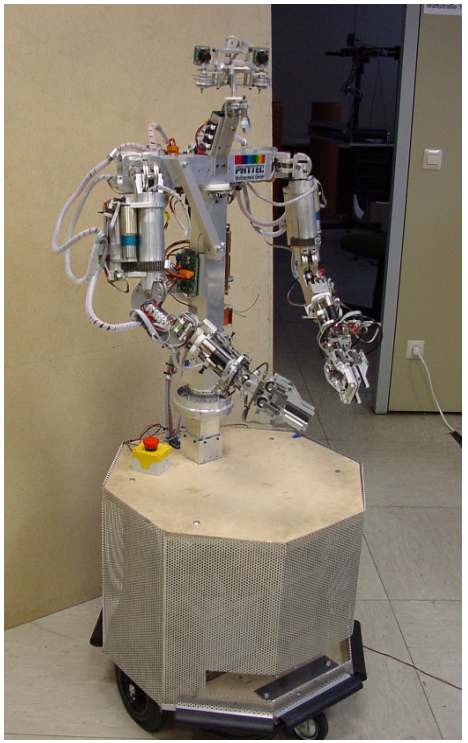
Thank you for your attention

Karlsruhe, Germany, SFB 588

ARMAR-III, 2006



ARMAR-II, 2002



ARMAR, 2000

KAWAI-II, 2005