Overview of DFKI-RIC research topics, focused on space robotics

DFKI Bremen & University of Bremen
Robotics Innovations Center
Director: Prof. Dr. Frank Kirchner
www.dfki.de/robotics
robotics@dfki.de

Universität Bremen

DFKI-RIC Space Robotics

Hardware in the Loop Simulation

Verification of Rendezvous and Capture Maneuvers and Systems

• Summary

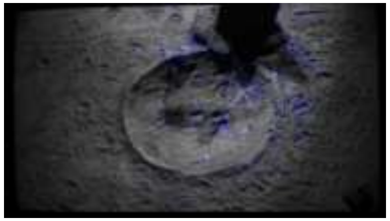
- ❒ Day 1
  - Summary of Rendezvous and Capture related developments at DFKI
  - Hardware of the HIL movement simulation system
  - Core of the attached simulation system
  - Coordinate transformations for the control of the movement system
- ❒ Day 2
  - Avoiding HIL movement system self-collisions
  - Optimizing physical workspace usage
  - Measuring and improving movement system precision
  - Simulating and evolving alternative capturing systems
  - Processing optical sensor Data

- Summary of Rendezvous and Capture related developments at DFKI

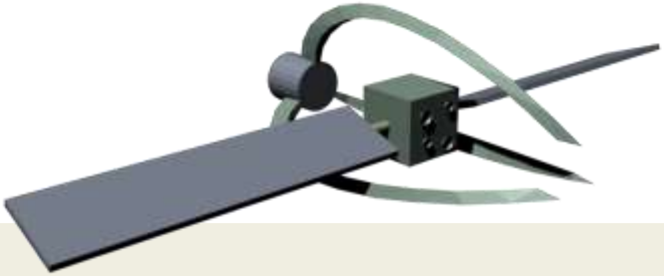
❐ Goals and Tasks

- Construction of a “Hardware in the Loop” simulator that can test real hardware (sensors like cameras and LIDAR) and software (navigation, image processing etc.) of a Servicer under realistic conditions
  - Simulation of the relative motion of up to two objects (Servicer and Client) in a workspace as large as possible
  - Transfer of the movement simulation results in realtime to the real mockups of the Client and Servicer, which are attached to a KUKA robotic arm and a cable robot.

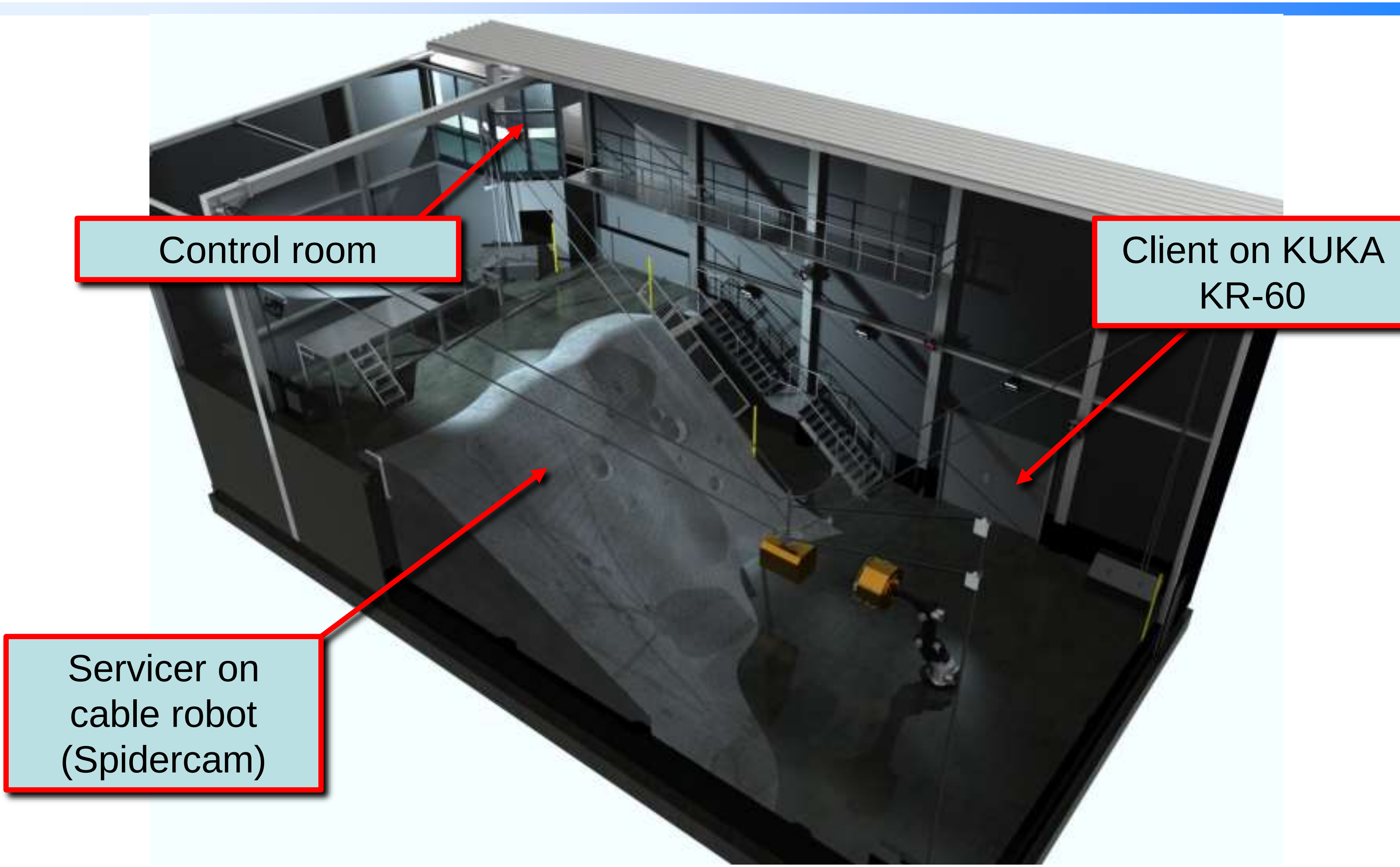
- Camera hardware und camera data preprocessing



- Development of new technologies for capturing satellites



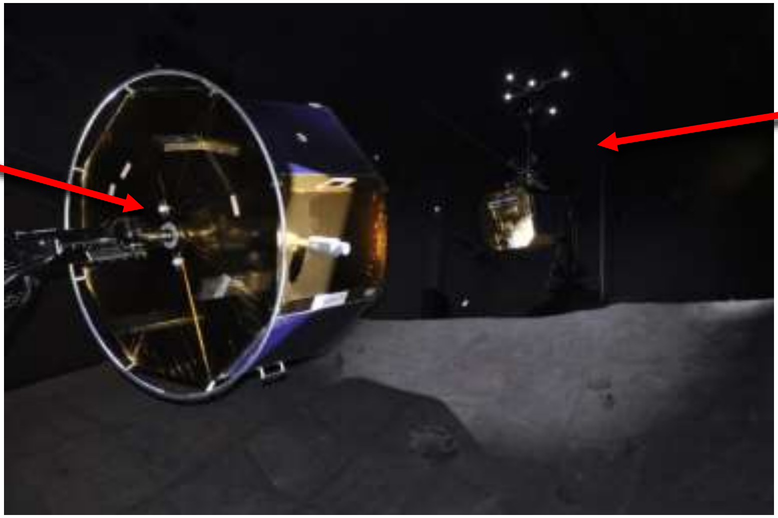
Exploration hall



❒ Advantages of „Hardware in the Loop“ simulation:

- The real sensors and real data processing hardware for sensor data and navigation can be tested according to a software simulation of the orbital dynamics

Client on KUKA KR-60



Servicer on cable robot

❒ Problem to solve:

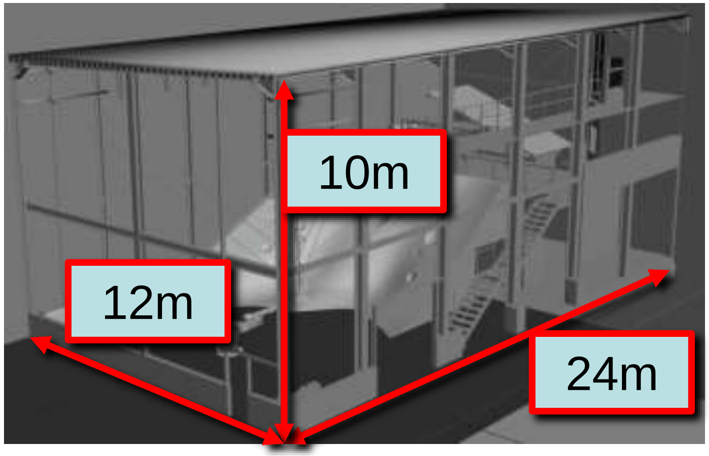
- Convert the unrestricted movements of two objects (2x6D $\Rightarrow$ 12D) to the KUKA arm (6D, restricted) and the cable Robot (3-4D, restricted)
- 12D $\Rightarrow$ 9-10D

Rendezvous and Capture related developments at DFKI

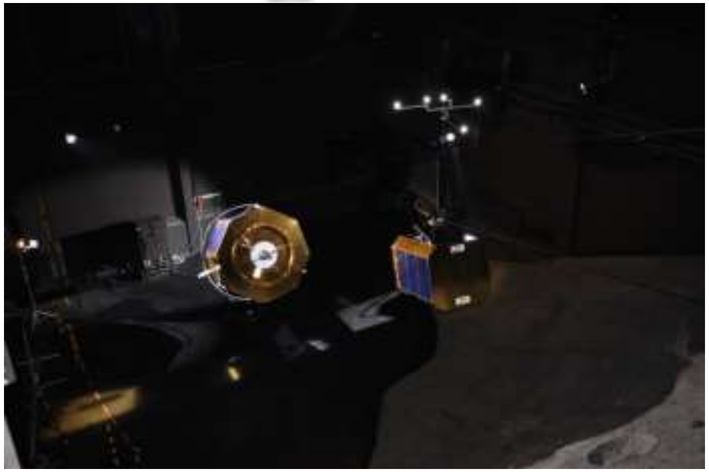
- **Hardware of the movement simulation system**

Hardware of the movement simulation system

- Exploration hall



- Length 24m
- Width 12m
- Height 10m



Hardware of the movement simulation system

❒ Exploration hall

➢ Lighting system

- Realistic optical sun light simulation from configurable directions
- 6 light sources with controllable pan-tilt units
- 5 elevators for changing the height auf the lights
- 575W / 6000K / 1000h / 49000lm
- Comparable to 2500W Daylight-lamp
- 14500 Lux at 10m/12°

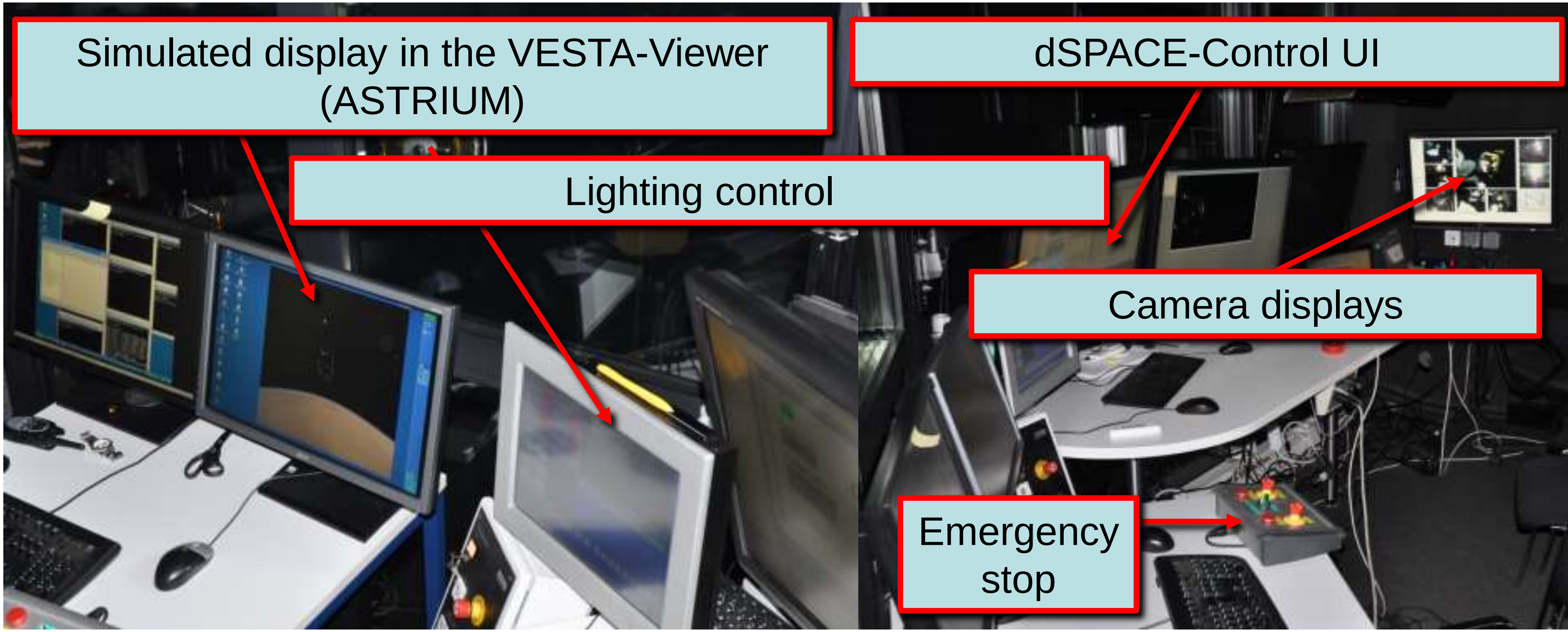
Hardware of the movement simulation system

- ❒ **Exploration hall**
  - ➢ Special light absorbing wall paint, so that only the mockups are visible
    - ⇒ More realistic visual sensor data

Hardware of the movement simulation system

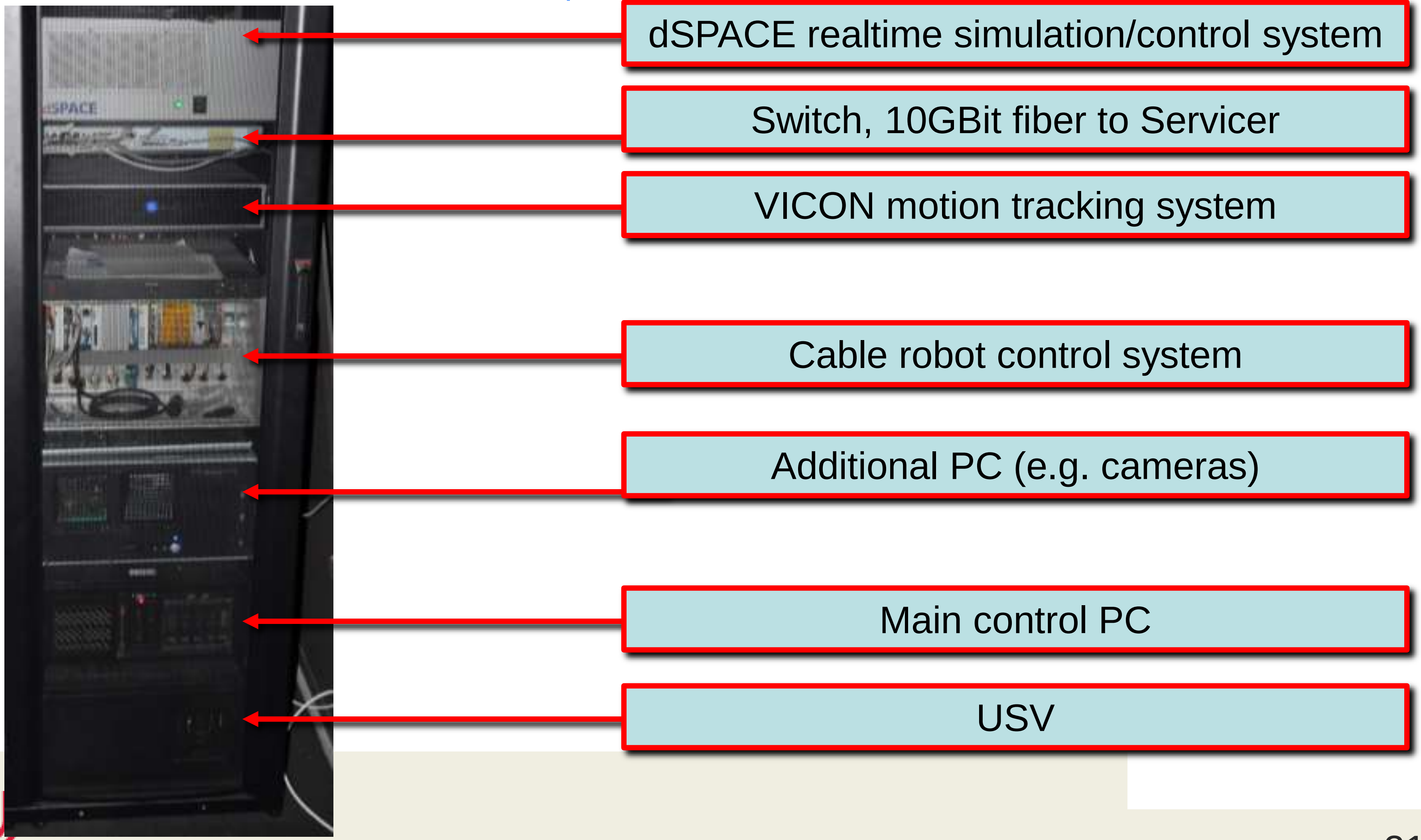
❒ Exploration hall

- Control room: Camera displays and control UIs



Hardware of the movement simulation system

- Exploration hall
  - Control room: Control Computers



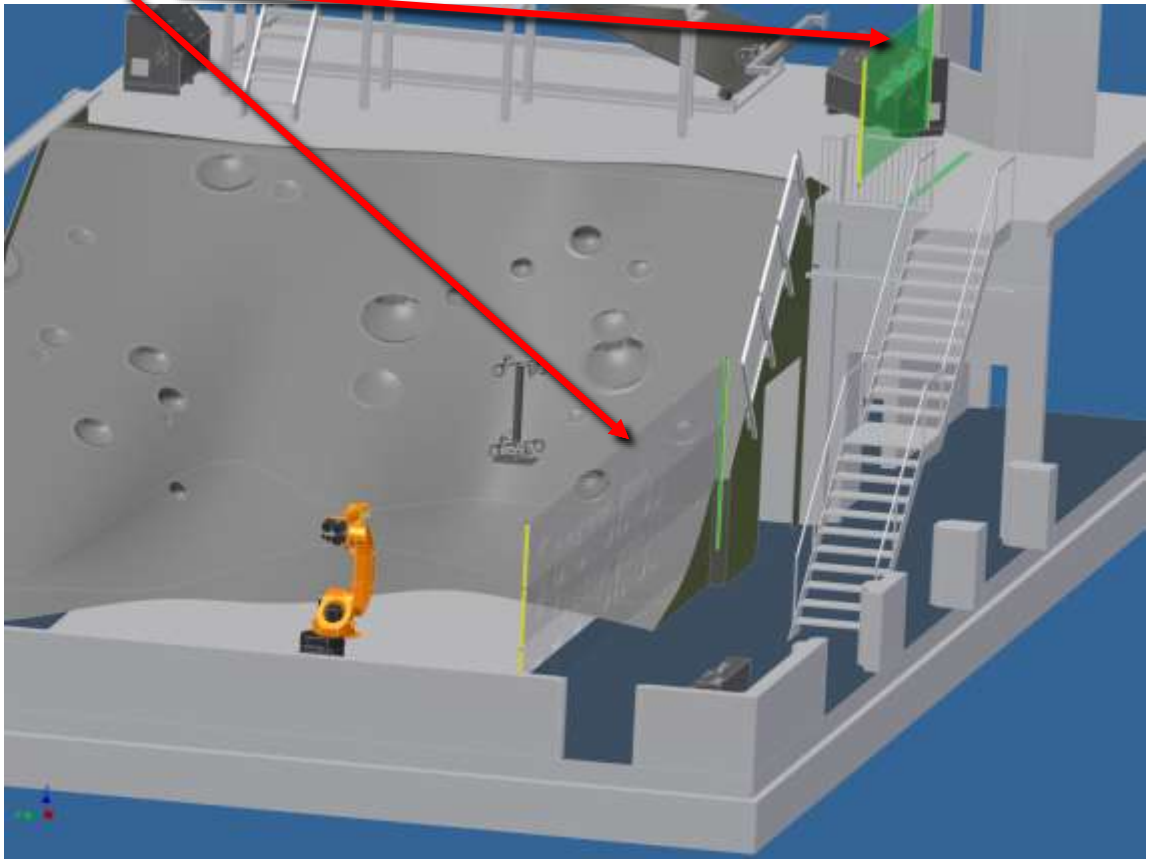
Hardware of the movement simulation system

- ❐ **Exploration hall**
  - ➢ Security systems
    - Combined emergency stop for KUKA and cable-robot in the control room

Hardware of the movement simulation system

❒ Exploration hall

- Security systems
  - Light curtains in danger zone

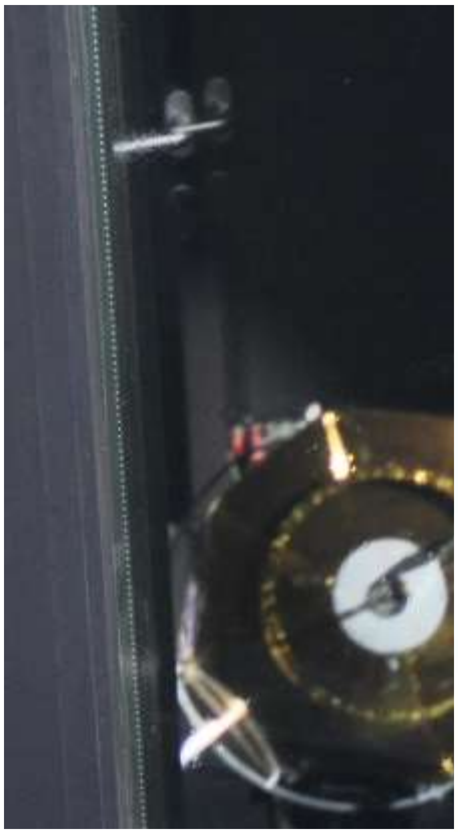


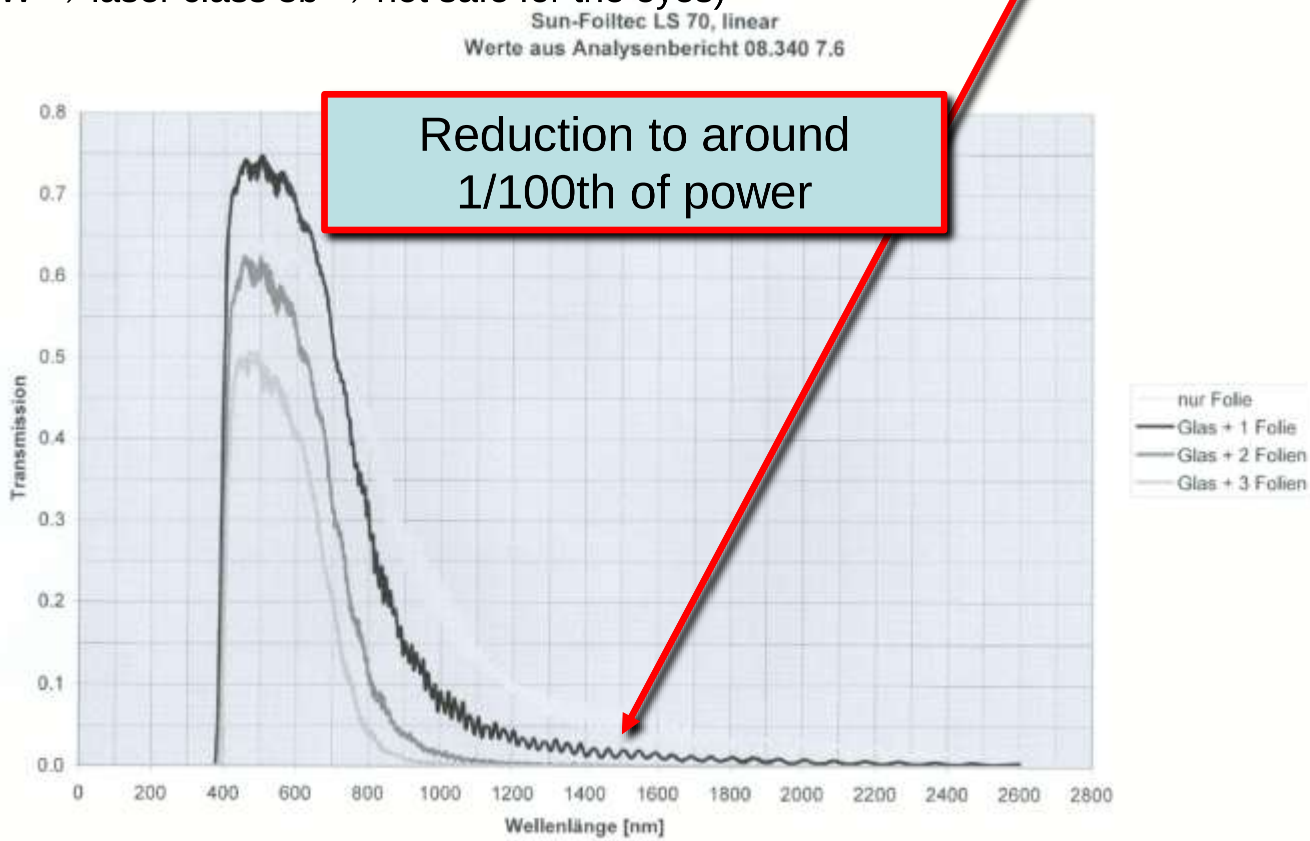
Hardware of the movement simulation system

❐ Exploration hall

➢ Security systems

- Laser filter film to protect glass windows of control room (LIDAR: 1550 nm (IR), ~ 38mW ⇒ laser class 3b ⇒ not safe for the eyes)



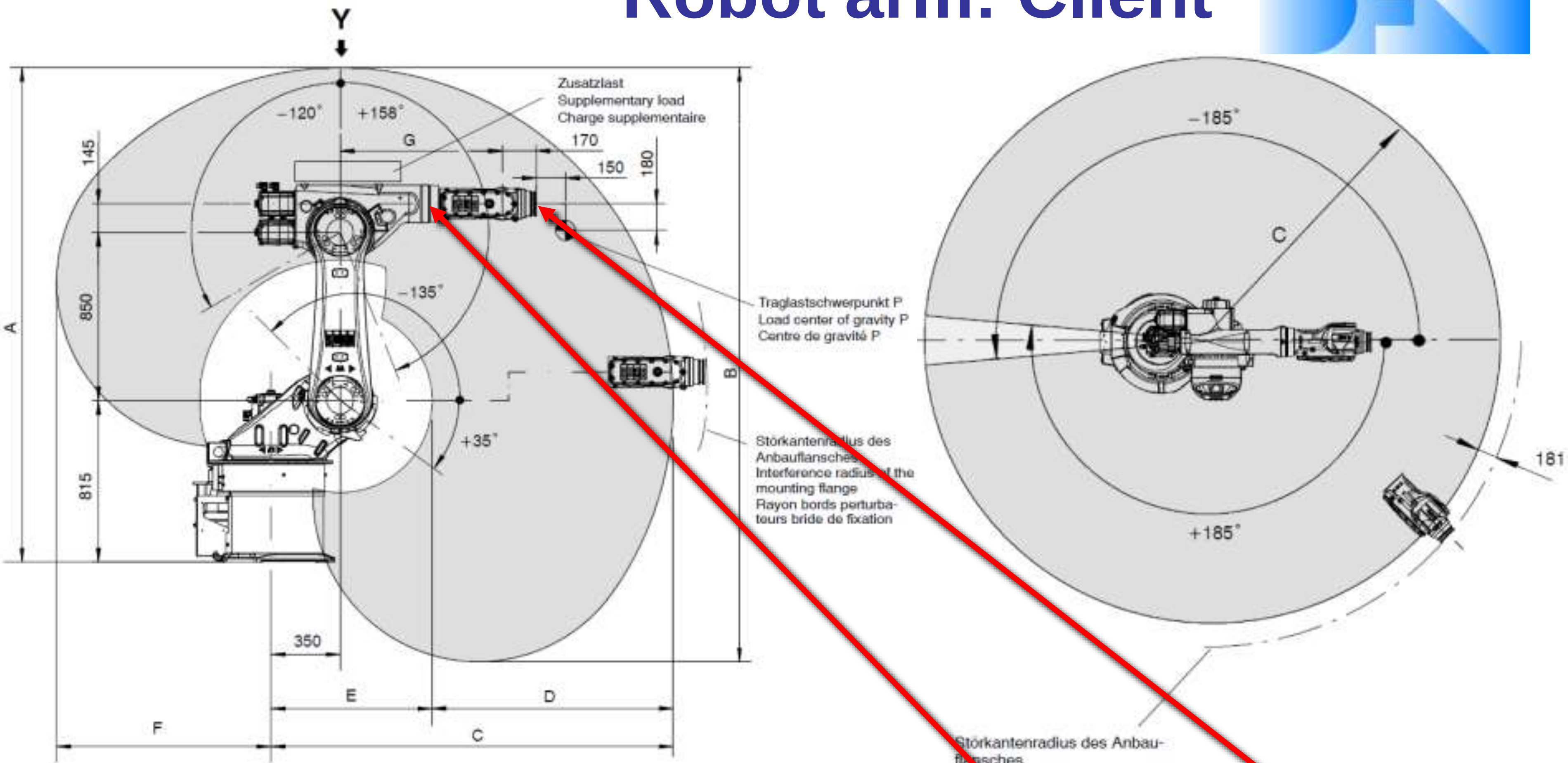


Reduction to around 1/100th of power

Robot arm: Client

Model	KR-60-3
DOF:	6
Lifting:	60kg
Control:	Remote Sensor Interface (RSI) via Ethernet
Control rate:	12ms
Accuracy (repeatability):	+/- 0.2mm by manufacturer, 0.01mm measured

Robot arm: Client

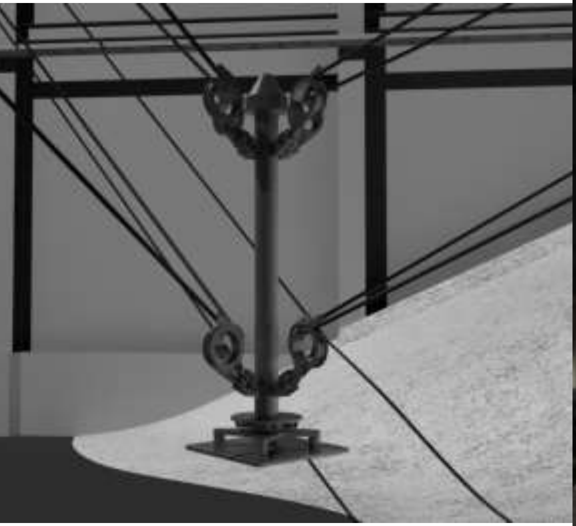


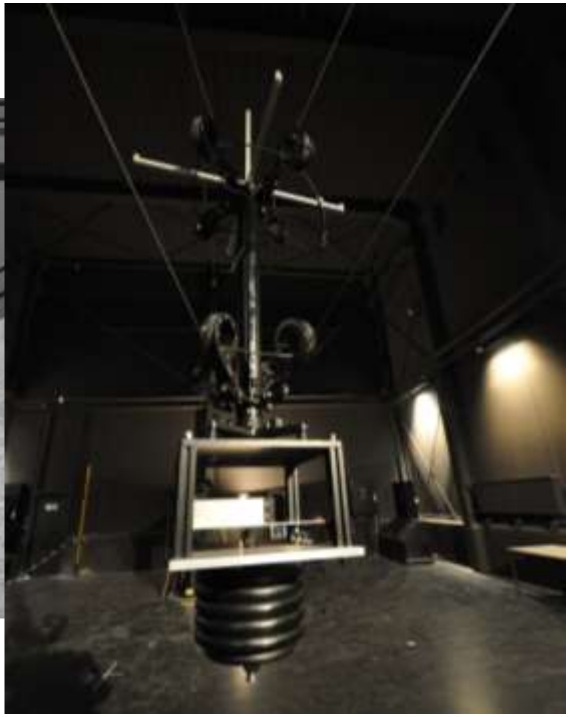
Maß	A	B	C	D	E	F	G
Länge (in mm)	2498	3003	2033	1218	815	1084	820

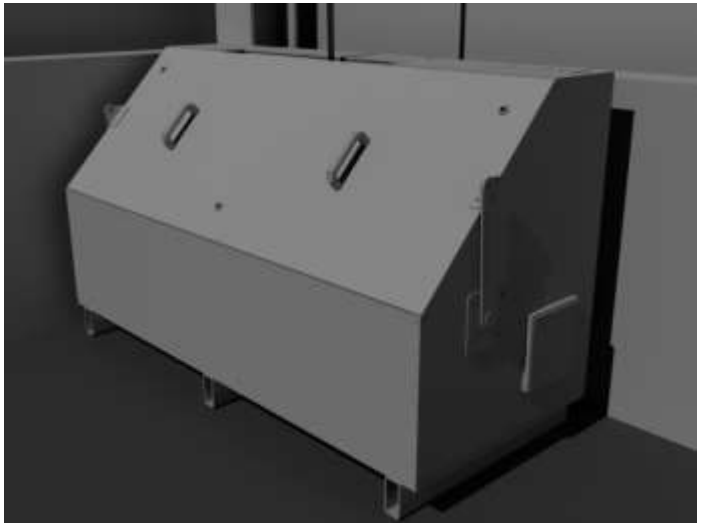
Achse	A1	A2	A3	A4	A5	A6
Winkelbereich	±185°	+35° /-135°	+158° /-120	±350°	±119°	±350°
Maximale Winkelgeschwindigkeit	128° /s	102° /s	128° /s	260° /s	245° /s	322° /s

Cable Robot: Servicer

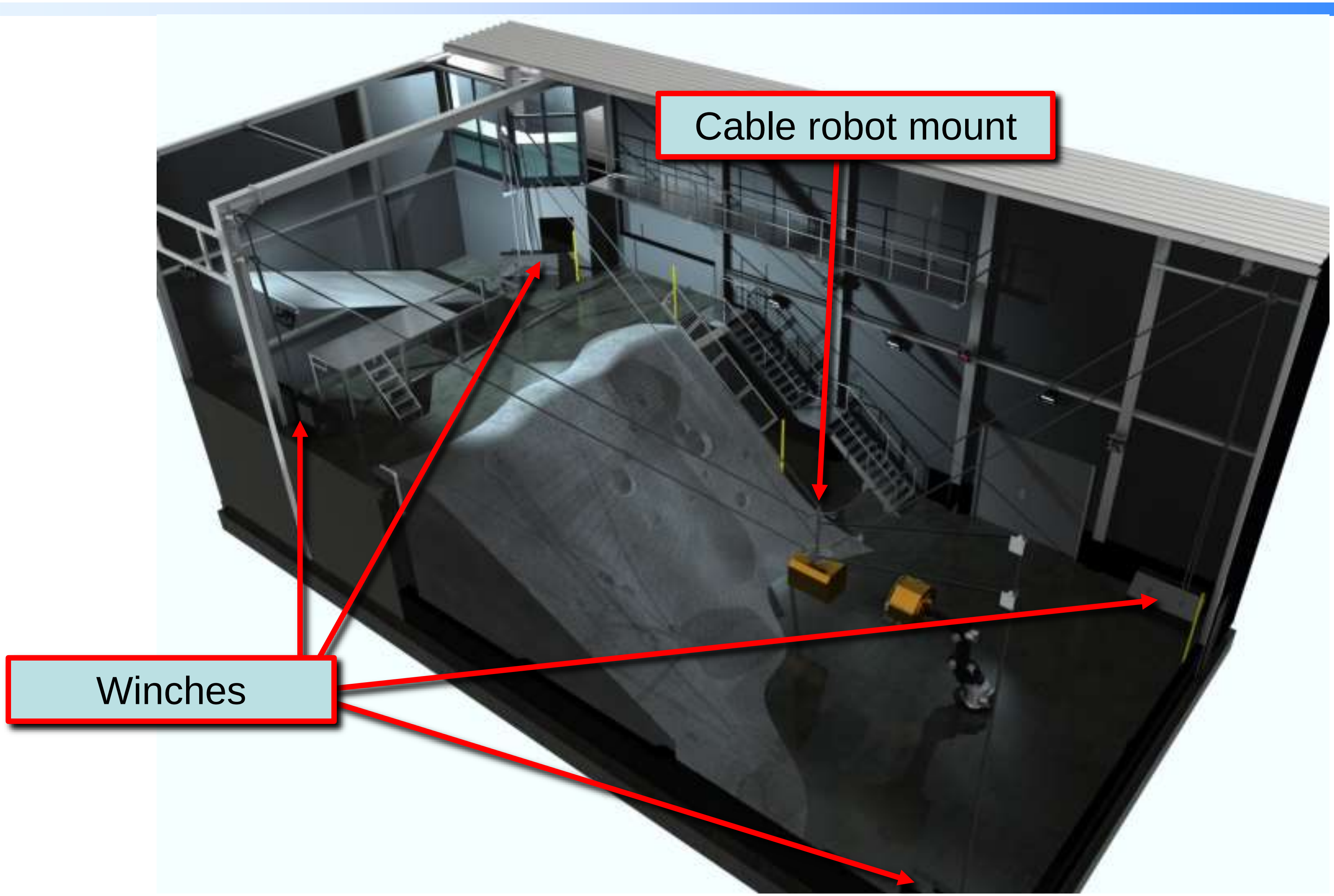
- Wire driven platform based on a SpiderCam custom design



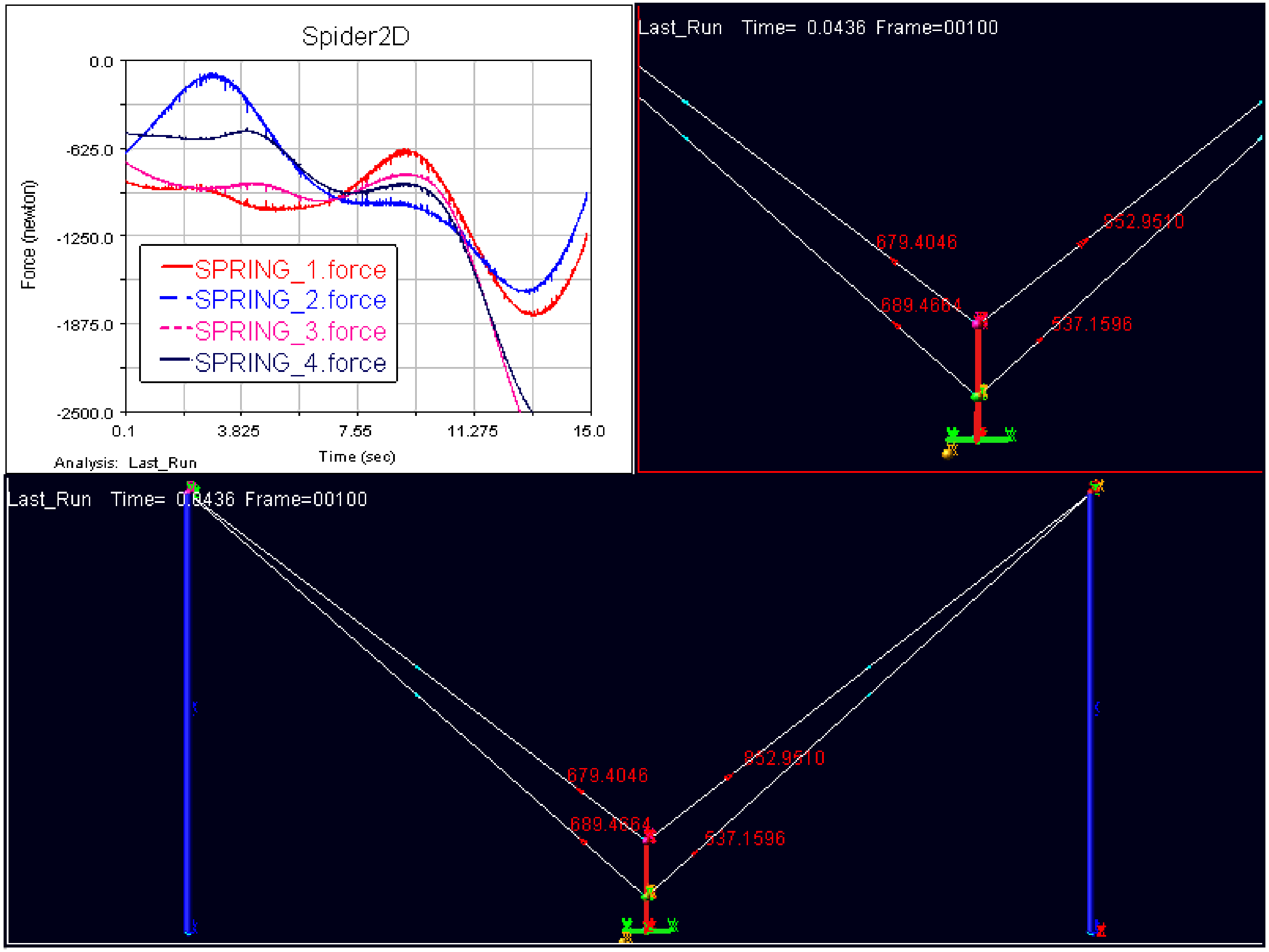




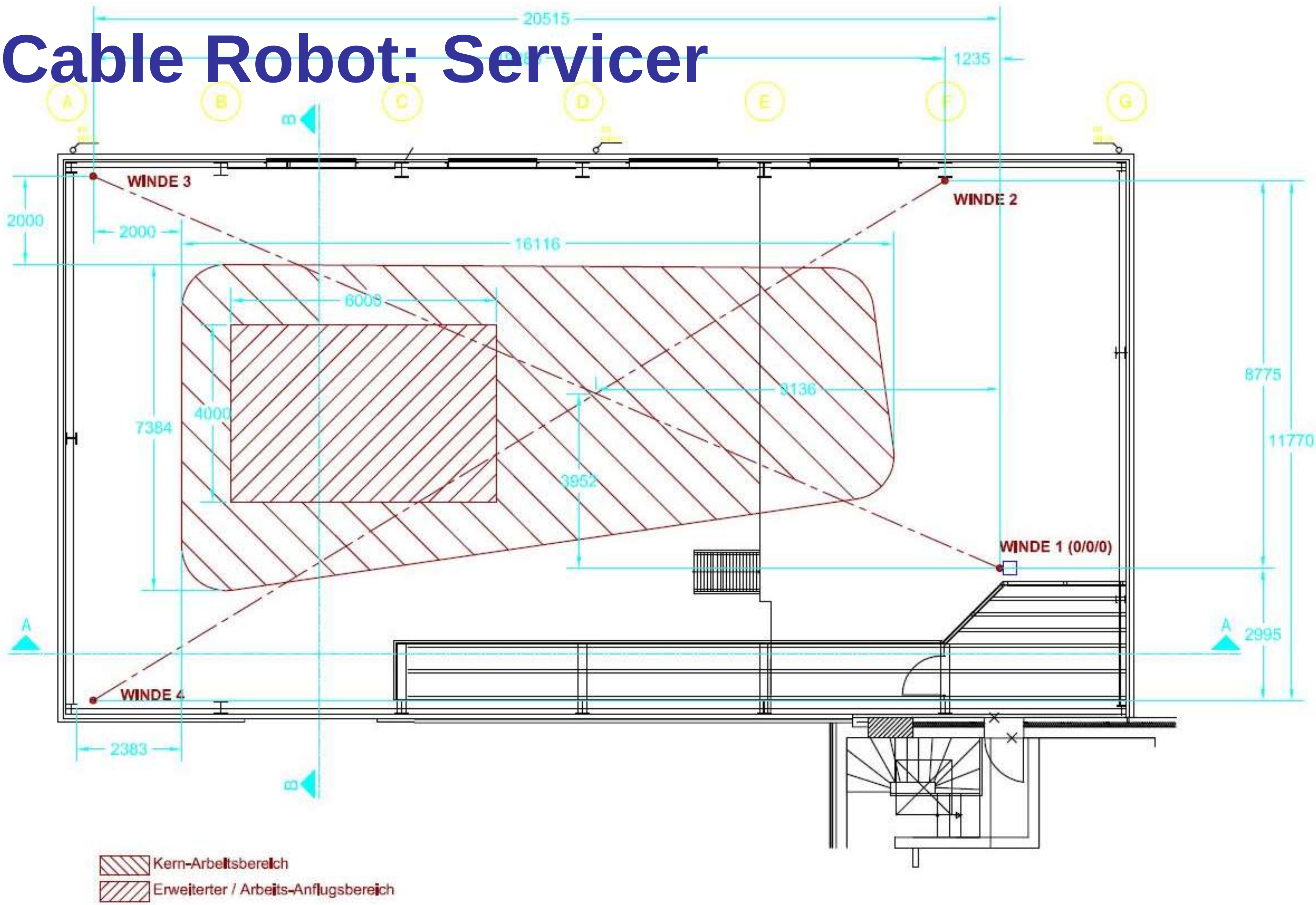
Cable Robot: Servicer



Cable Robot: Servicer



Cable Robot: Servicer

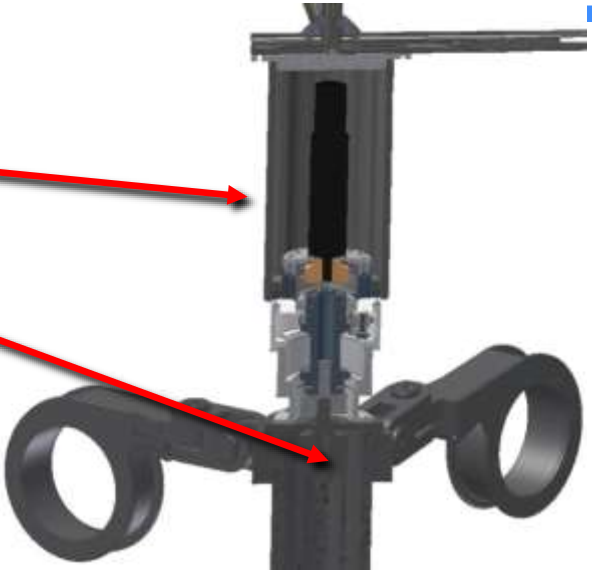


Cable Robot: Servicer

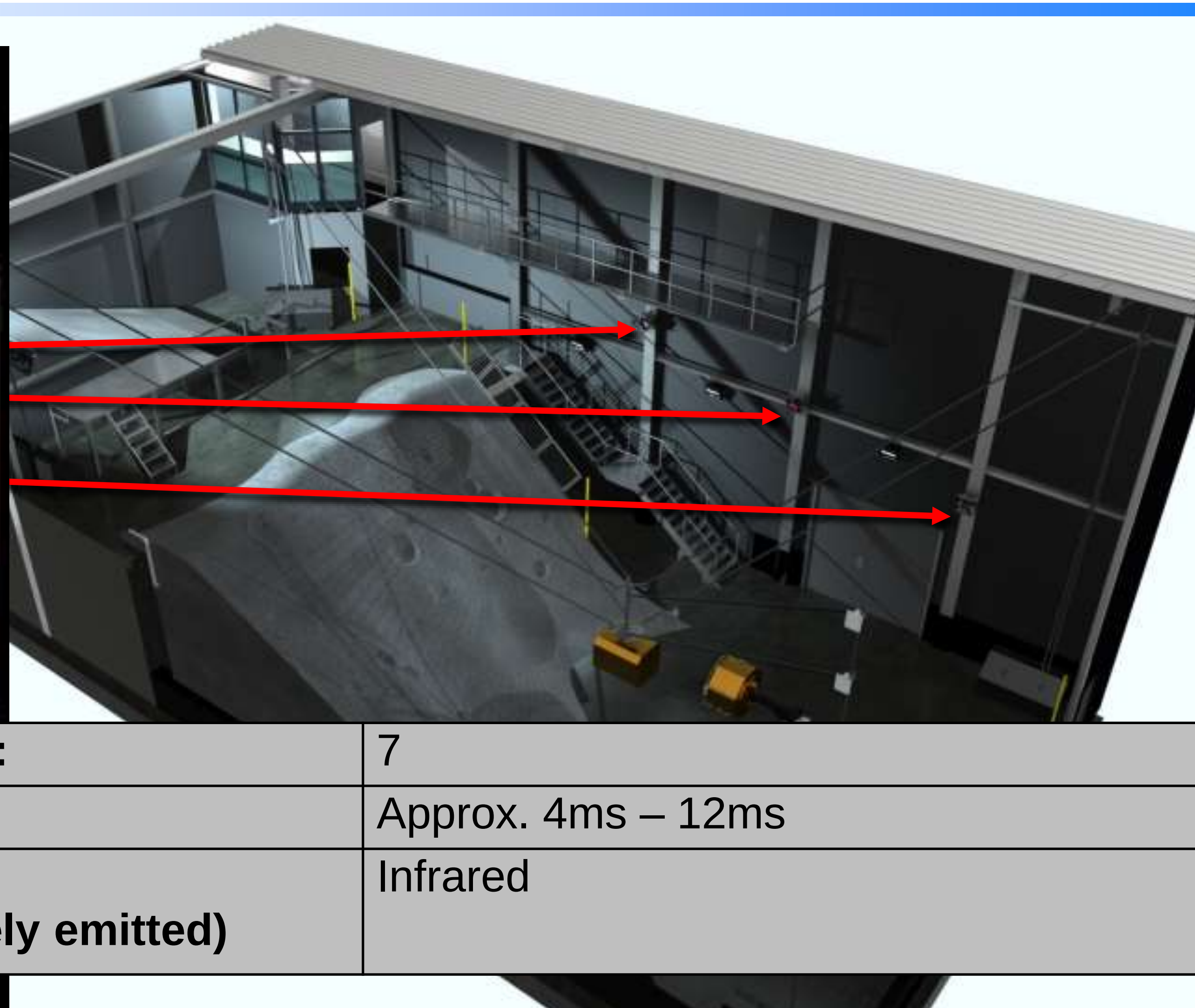
Number of wires:	8
Number of winches:	4
Core workspace with high accuracy:	6m x 4m x 4.50m (L x W x H)
Repeatability in core workspace:	ca. 1.1mm
Extended workspace:	16m x 7m x 5.50m (L x W x H)
Minimal translational speed:	0.1 mm/s
Maximum translational speed:	2m/s
Maximum acceleration:	1.5m/s²
Workload (constant):	150kg
Data transfer rate to mount:	10 GBit/s via glass fibre, switch on mount
Electric power on mount:	230V AC with1kW, 24V DC
Control:	CAN (bus system)
Control rate:	4ms

Z-axis of the cable robot

- Attachment:
  - On the upper end of the mount
  - Connected to inner axis
- Drive:
  - Faulhaber 3863-24V / 3,8A / 6700rpm / 110mNm
  - Gear 1: Planetary gear 12:1
  - Gear 2: Harmonic drive 120:1
  - Combined gear: 1440:1
  - Torque: 158Nm / 578Nm
  - Turning speed: 13s/360°
- Sensors:
  - Absolute senspr on drive axis:
    IC Haus ic-MH; 0.1° Resolution
  - Incremental sensor on motor axis:
    32 steps per 360° cycle

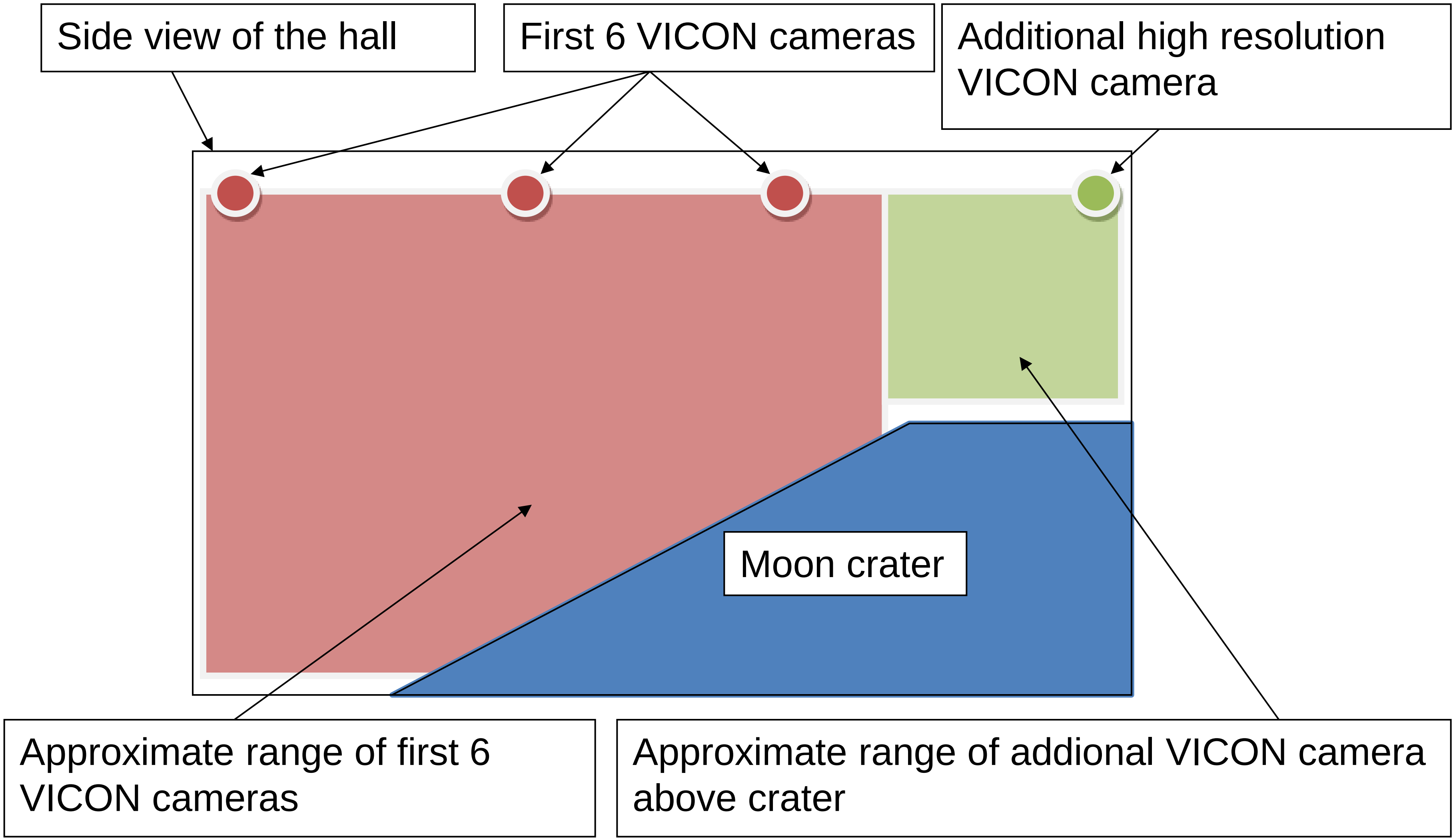


VICON tracking system



Number of cameras:	7
Data rate:	Approx. 4ms – 12ms
Wavelength range: **(received and actively emitted)**	Infrared

VICON tracking system



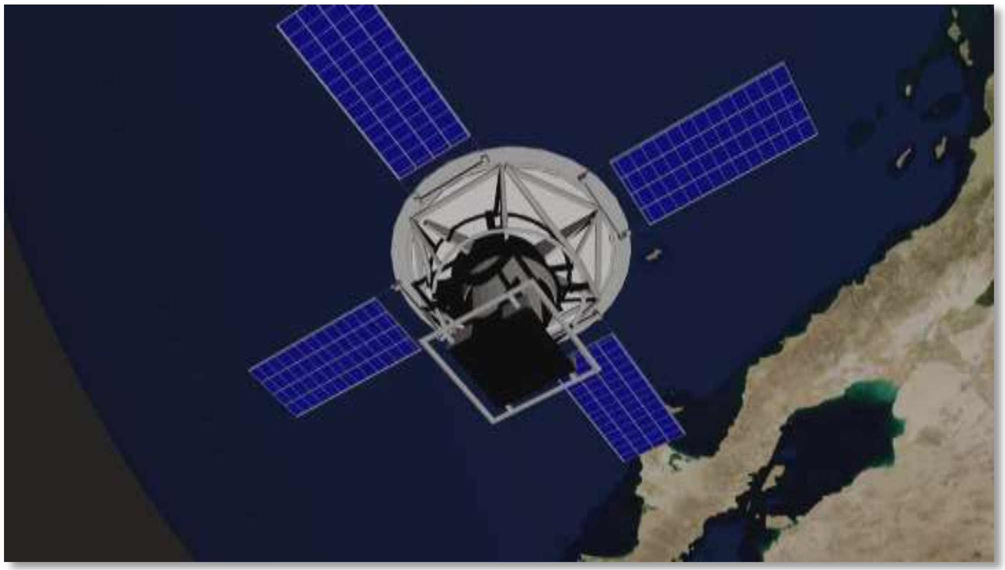
- **Questions so far?**

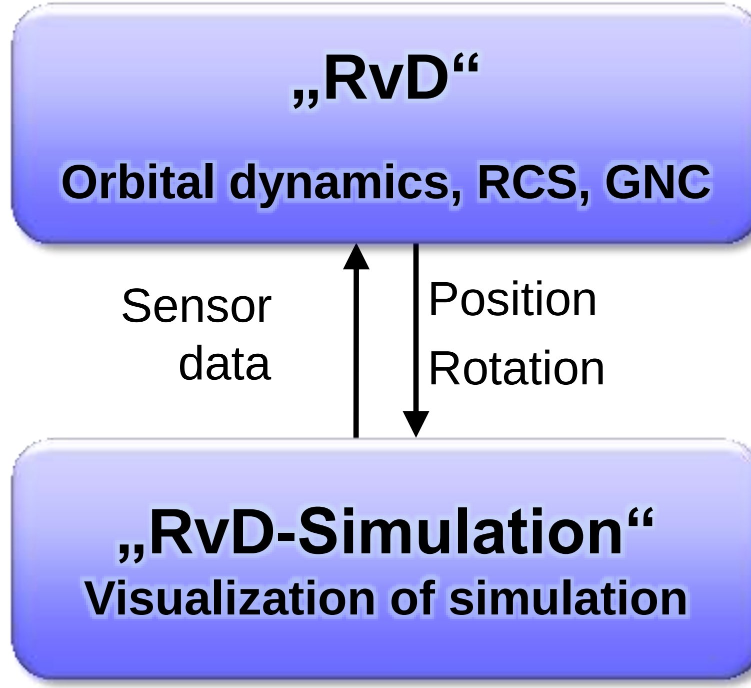
- **Core of the simulation system**

Core system

- Main part of simulation
  - Simulated aspects:
    - Orbital dynamics
  - Real „Hardware in the Loop“ components:
    - Sensor data processing
  - Simulates behavior of two satellites
    - Returns new position and rotation of objects

Core system

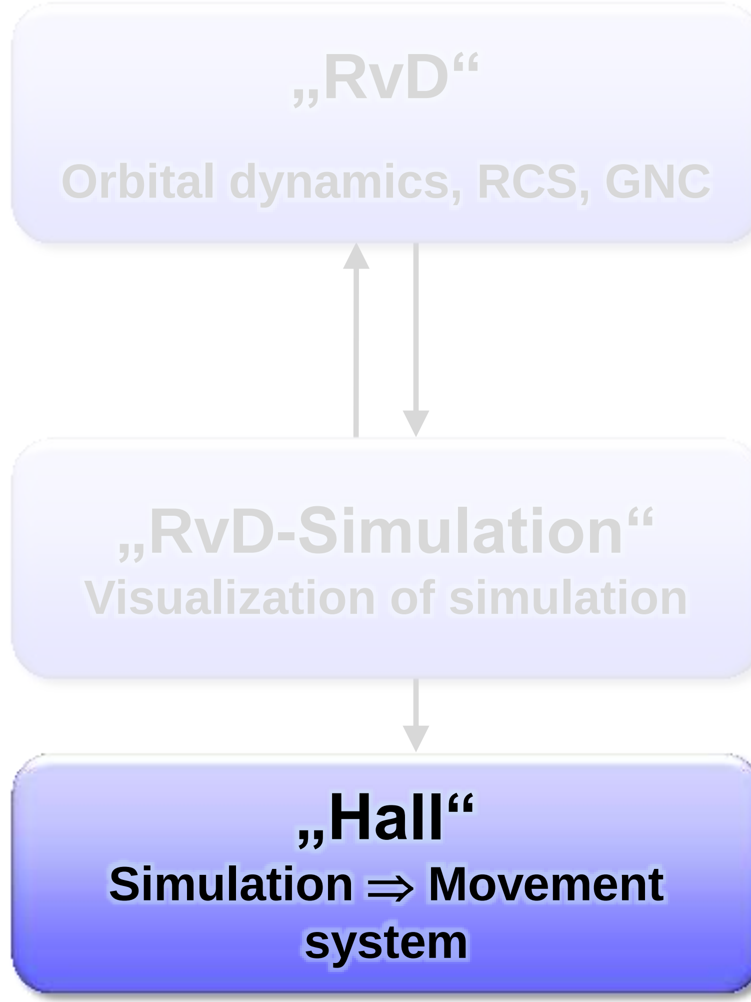




- Visualizes simulated objects and environment
- Visualization can be used for verification of “RvD” and virtual sensors

Core system

- Transfers simulation state to the real mockups
- Calculates positions and rotations for KUKA and CableRobot
- $\Rightarrow$ Solves 12D$\Rightarrow$9D-10D problem
- Ensures safety borders



Core system

- Displays the state of the motion system graphically
- Monitoring of the movement system (also **before** it moves in reality)

„RvD“

„Graphics“
Virtual display of the movement system

- KUKA arm and CableRobot each move one of the simulated objects

Core system

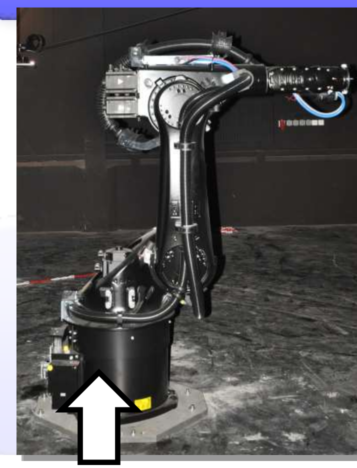
„RvD“

Orbital dynamics, RCS, GNC

- Sensordaten
  - Kameras / LIDAR / etc.

„Real“

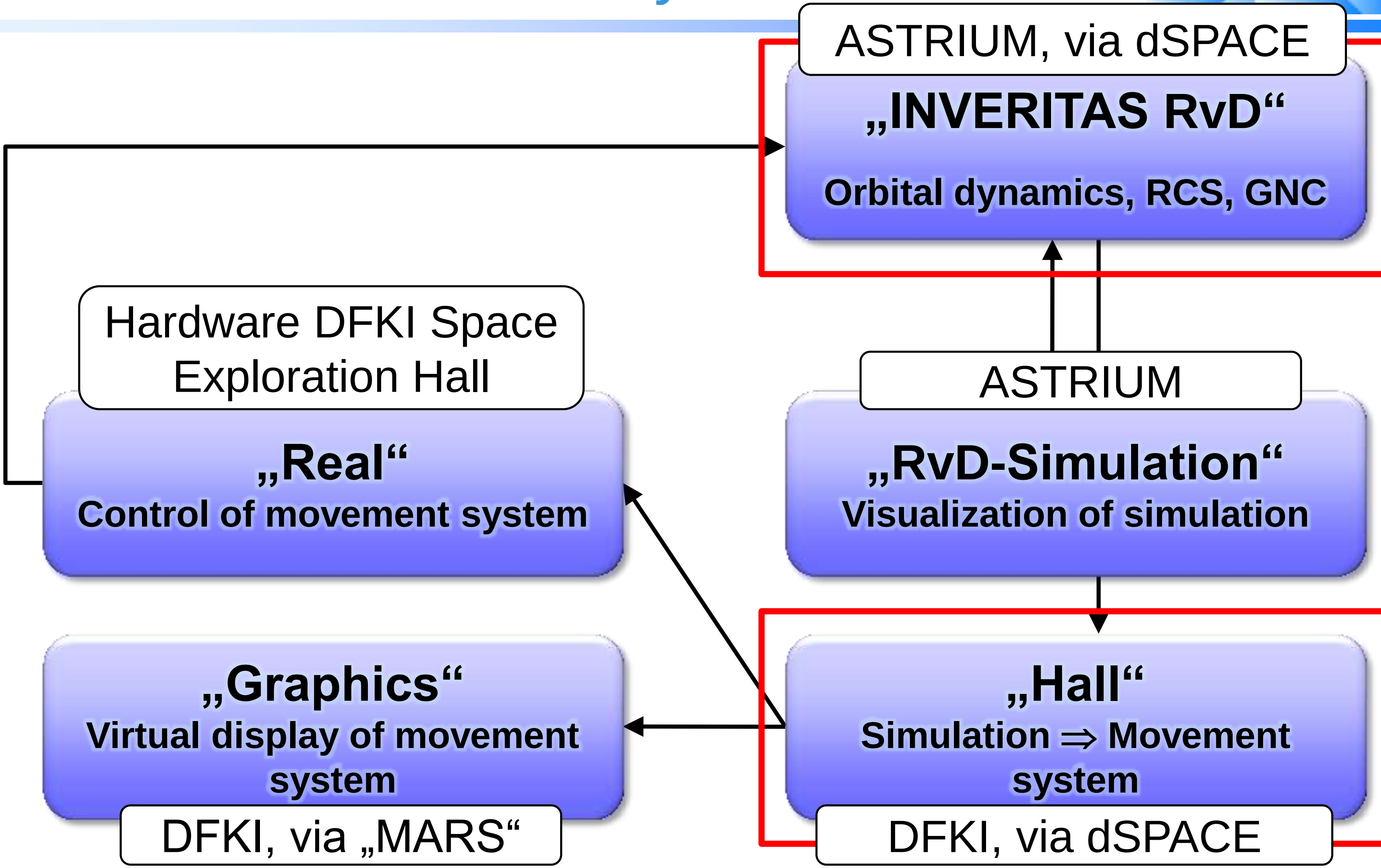
Control of movement system



- Controls movement system
- Provides real sensor data to „RvD“

(KUKA / CableRobot)

Core system



ASTRIUM, via dSPACE
„INVERITAS RvD“
Orbital dynamics, RCS, GNC
Hardware DFKI Space Exploration Hall
„Real“
Control of movement system
ASTRIUM
„RvD-Simulation“
Visualization of simulation
„Graphics“
Virtual display of movement system
DFKI, via „MARS“
„Hall“
Simulation ⇒ Movement system
DFKI, via dSPACE

Real-time system: dSPACE

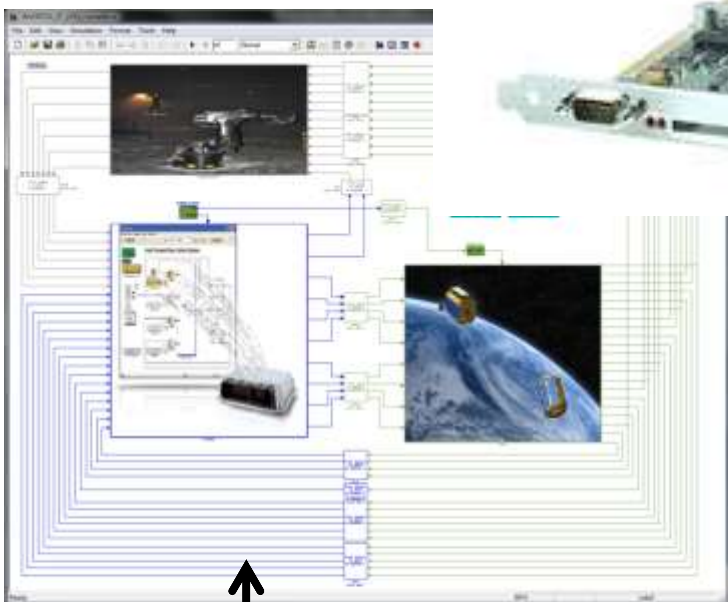
All real-time applications run on the dSPACE system

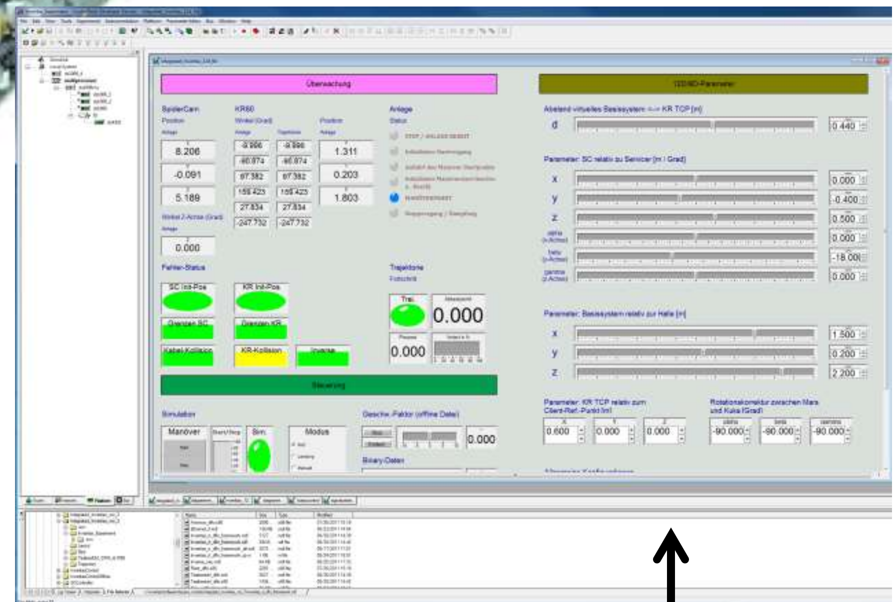
Advantages

- Hardware identical to Astrium simulation hardware
- Easily programmable using Matlab/Simulink
- Reliable and industry proven (car development)
- Many I/O boards available

Hardware:

- DS1006: 4-processor board
- DS4302: I/O board with 4 CAN channels





Design of a Simulink model

Automatic code-generation

Load application to dSPACE board

Design of a GUI: Simulink variables available via ControlDesk

Control of the application via ControlDesk (GUI)

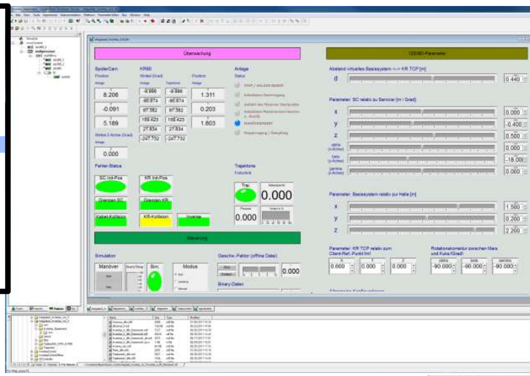
Distribution of tasks:

Each of the four cores has a particular task:

- Two cores were only used by Astrium (orbital dynamics and satellite control (GNC)) → Astrium provided compiled binaries.
- One core is used by DFKI to solve the 12D $\Rightarrow$ 9D-10D transformation and to control the hardware in der hall.
- A third core used by DFKI verwendeten runs Linux and controls the Ethernet interfaces. It exchanges data with the other DFKI core using the shared memory.
- The interfaces between the cores are defined in a Simulink model.

Overview

GUI



Shared memory

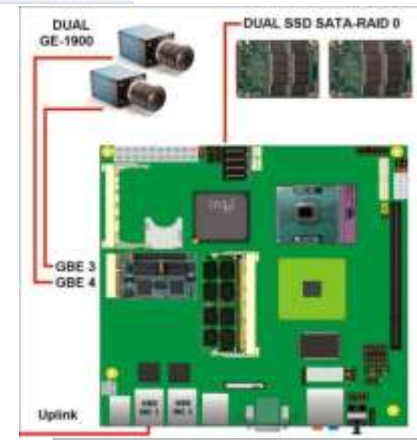
Astrium (Avionics)

Astrium (Plant)

DFKI (12d/9d control of Exp. hall)

Linux core

dSpace
Processor board with 4 cores



CAN

Ethernet

Stereo cameras LIDAR

4 ms

4-12 ms

12 ms

40 ms

CableRobot with Z-axis

VICON MTS

KUKA

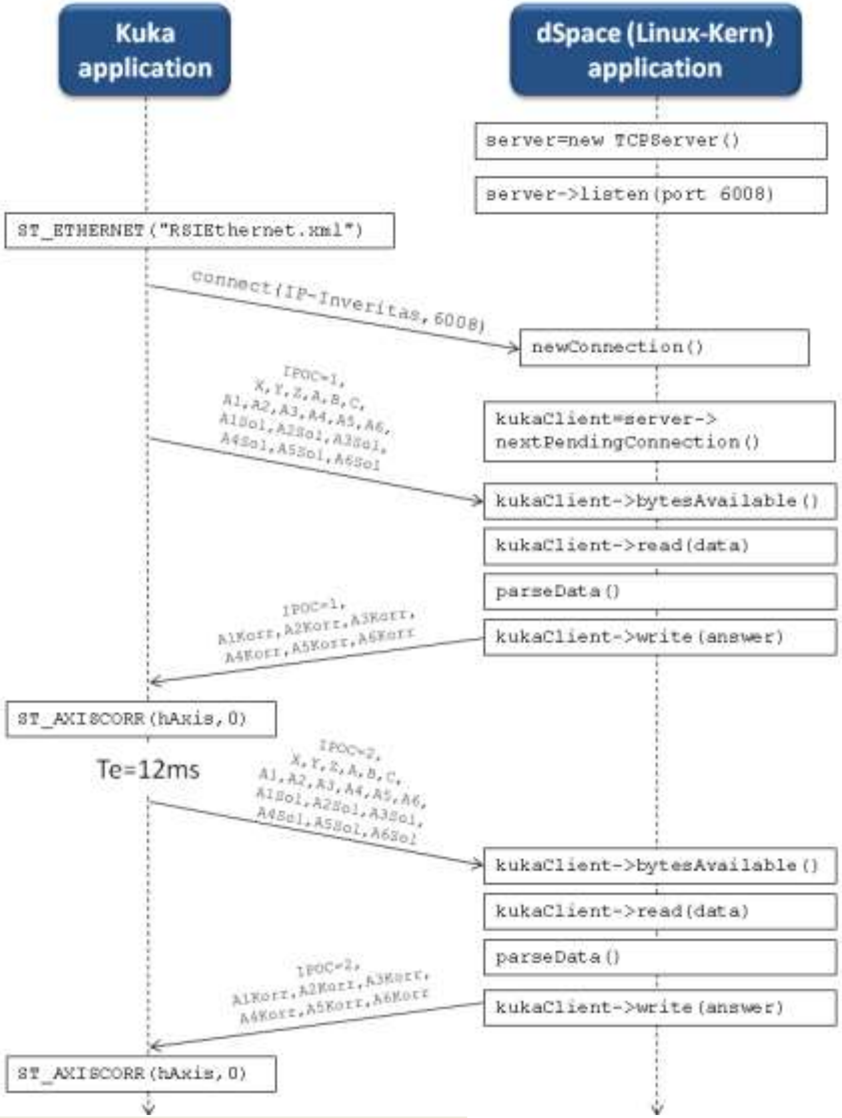
<u>Simulations:</u>

1. Exploration hall (DFKI, MARS)
2. Satellites in space (DFKI, MARS)
3. Satellites in space (Astrium, Vesta)
4. Collision avoidance (DFKI, Linux)

Control of KUKA via dSpace

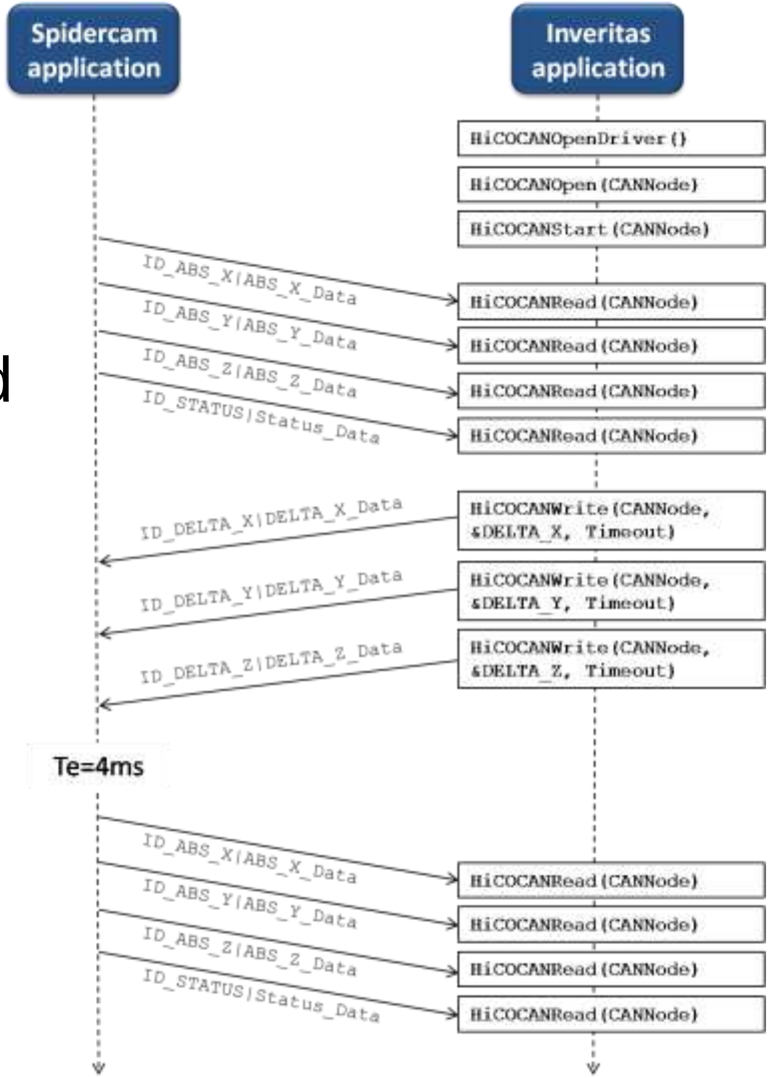
RSI-Protocol

- KUKA sends the position of the robot and an ID number in an XML text every 12ms
- The other side, an application in the Linux core of the dSpace system must immediately reply with a correction and the ID number



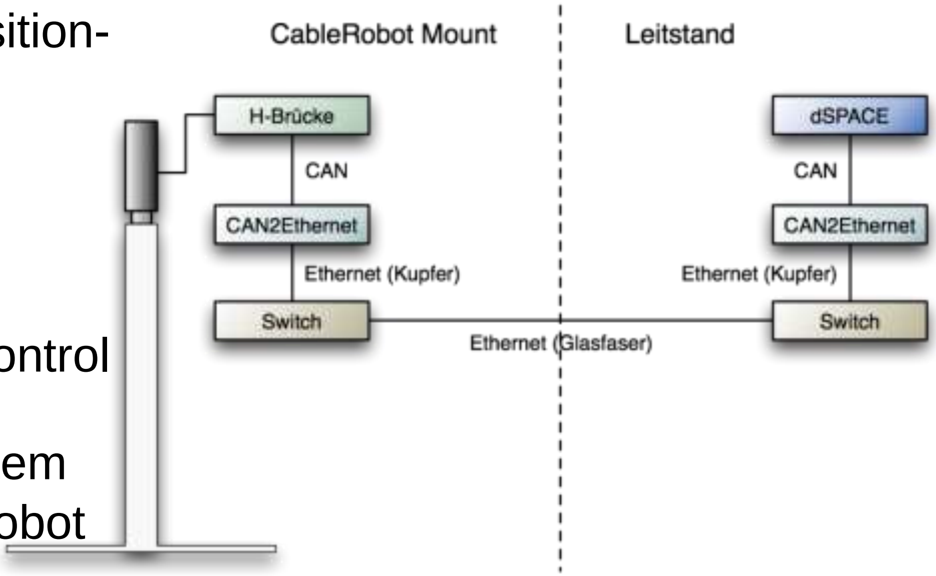
CableRobot control via dSPACE

- CAN-Bus
- 4 ms rate
- 12 ms delay between command and sensor reply
- Transformation between CableRobot coordinate system and world coordinate system necessary
- Position control



Z axis of CableRobot - control

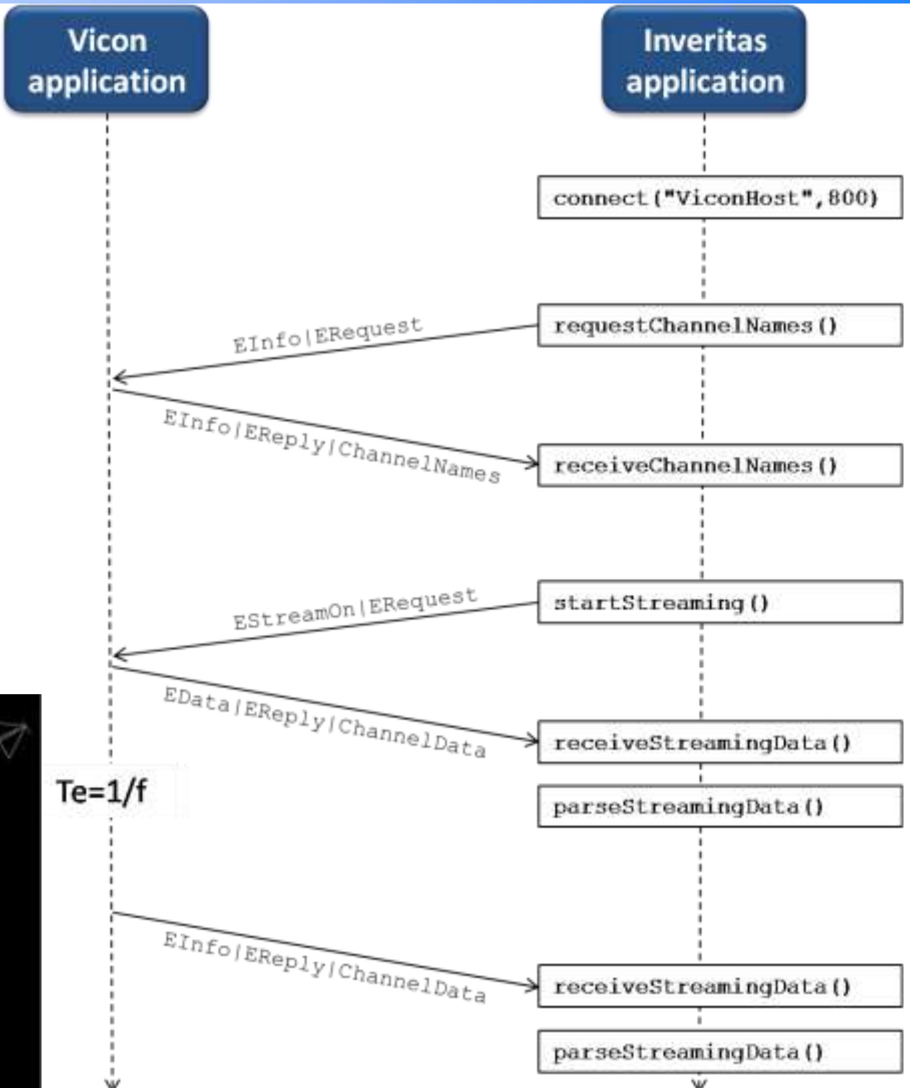
- Control:
  - H-Bridge (often used in DFKI RIC, e.g. iMoby)
  - STM32 microcontroller for a position-speed cascade controller
  - CAN interface

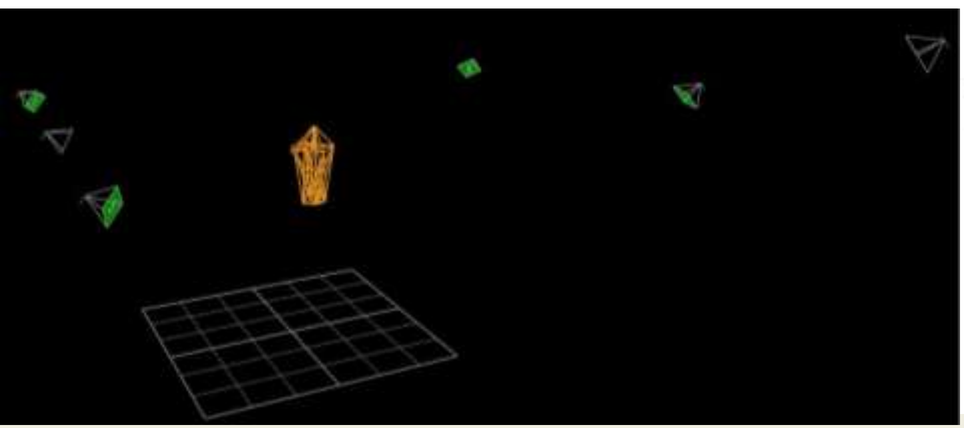


- Connection:
  - CAN to Ethernet bridge to the control room
  - Connection to the dSPACE system
  - Network connection via CableRobot glass fiber in private network
  - QoS settings can be changed if problems occur

Motion tracking system - control

- Exact position measurement of objects in workspace
- Ethernet TCP/IP
- 4 ms to 12 ms rate
- Source of world coordinate system



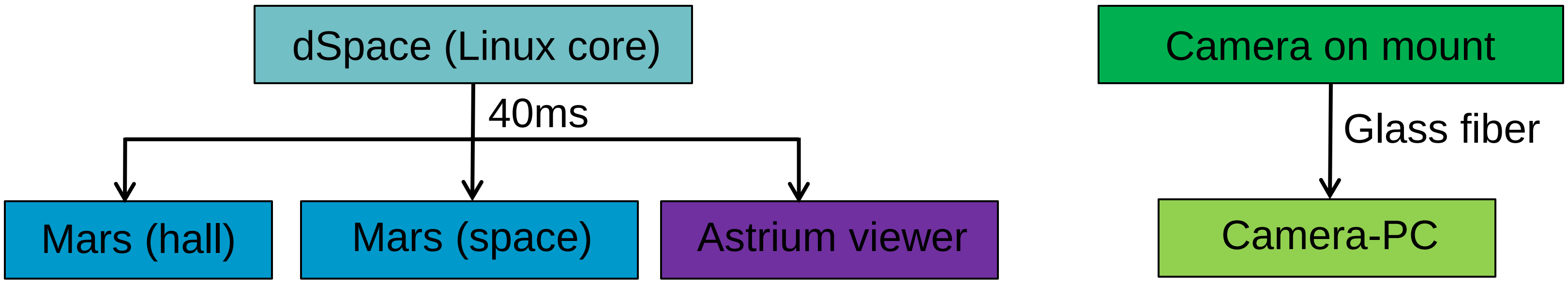


Visualizations

During the simulation the following visualizations can be used:

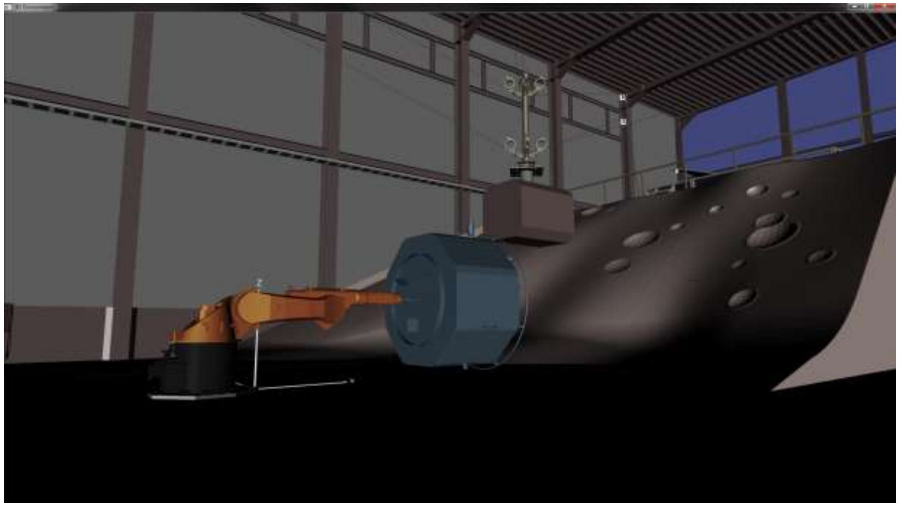
- Simulation of the hall with MARS
- Simulation of the satellites in space with MARS
- Simulation of the satellites in space with the Astrium viewer
- Camera images of the exploration hall

- These systems are connected in real-time via Ethernet

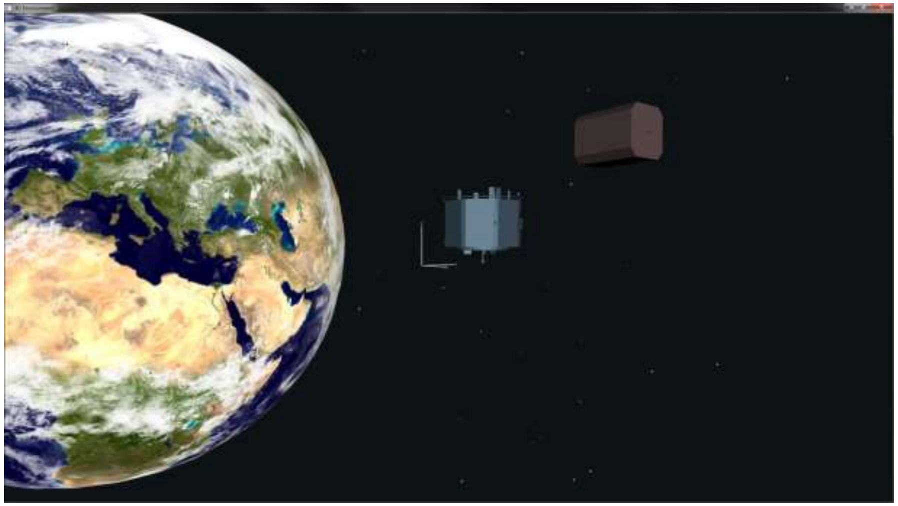


3D visualization via MARS

- The hall and the satellites in space are displayed in MARS in real-time
- In a second window the servicer camera can be simulated
- The connection is done via Ethernet (one port number per view)
- The data is sent every 40ms as standard C strings



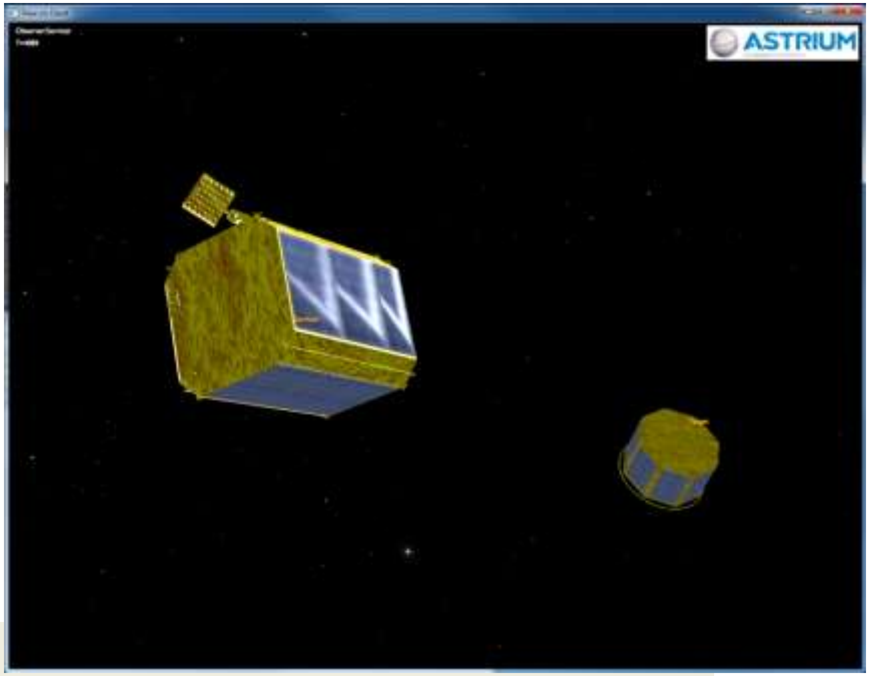
Visualization of the exploration hall



Visualization of the satellites in space

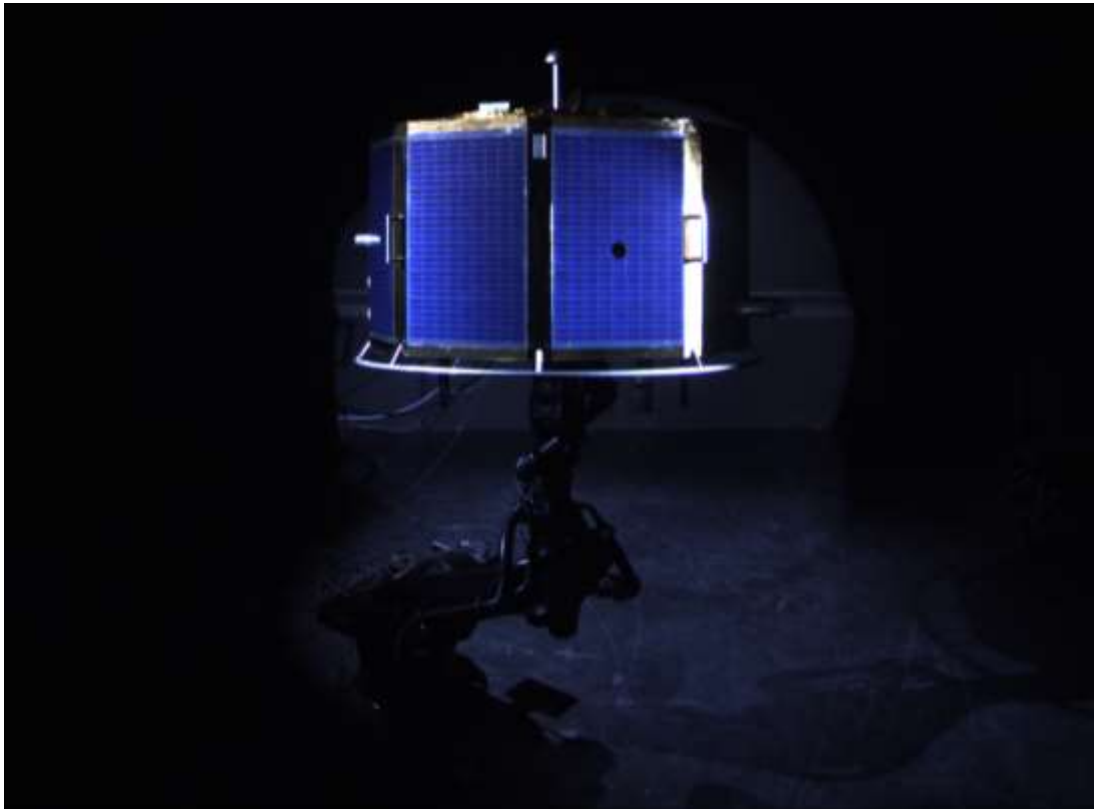
Astrium viewer

- The visualization software from Astrium runs in sync with the movement simulation system
- The data (time, position of the satellites, speed, orientation) is sent viea Ethernet every 40 ms as strings
- The camera on the servicer can also be simulated in a second window



Real camera images

- The real camera images are transferred via the glass fibers in the wires of the CableRobot (using Ethernet)

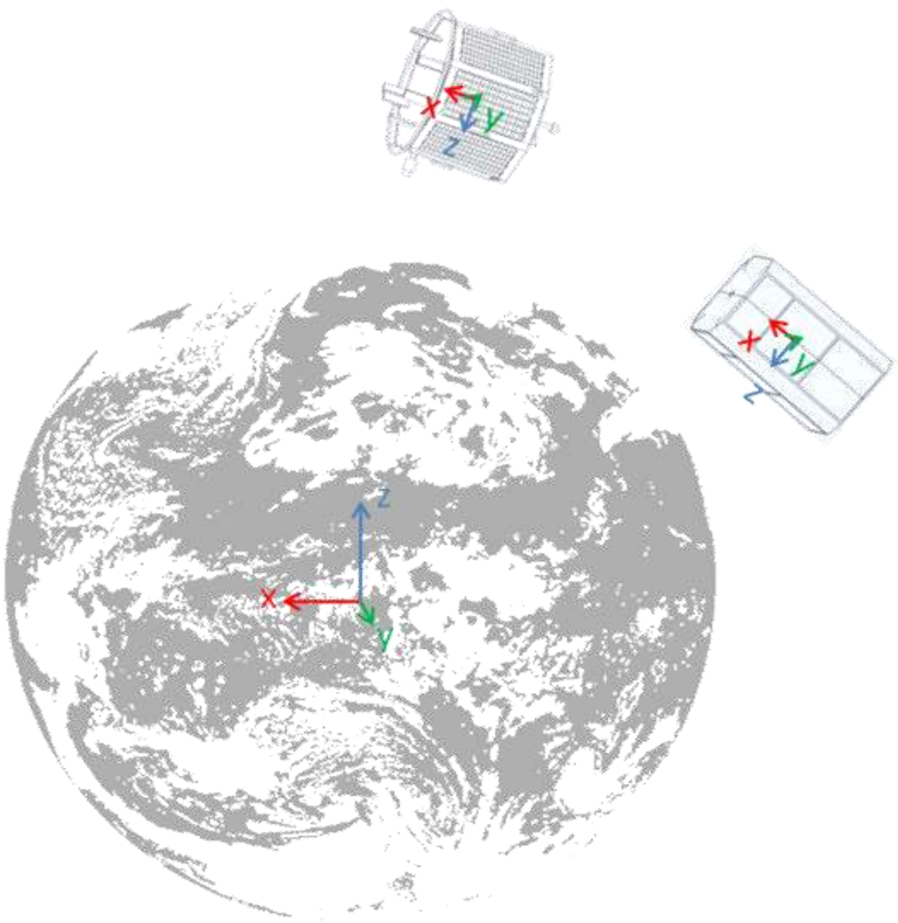


❐ Questions so far?

- ❐ **Next topic**
  - ➢ Coordinate transformations (12D⇒9-10D)

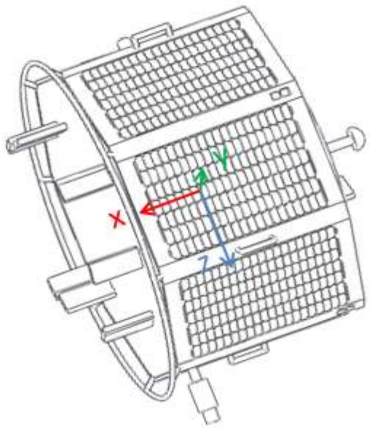
- **Reduction of DOF and Coordinate transformations**

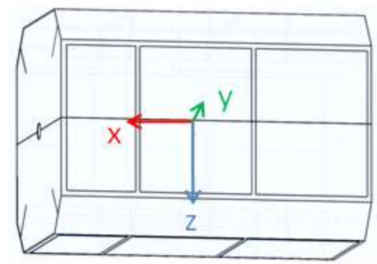
Reduction of DOF and Coordinate transformations



Reduction of DOF and Coordinate transformations

$$T_{Client}^{Servicer} = {T_{Servicer}^{ECI}}^{-1} \cdot T_{Client}^{ECI}$$





12D/10D

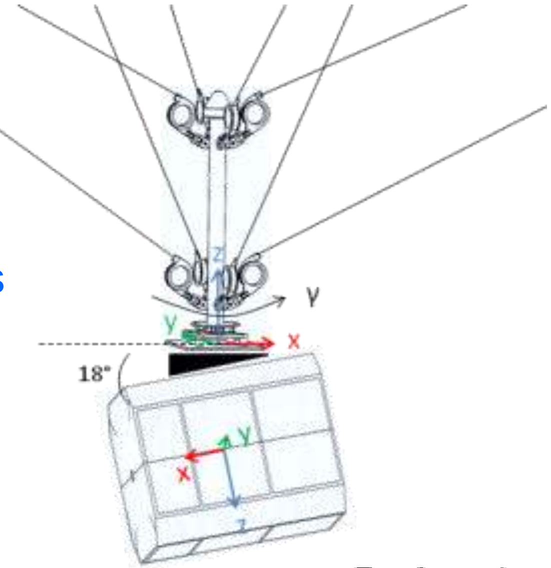
12D/9D – Client Fix

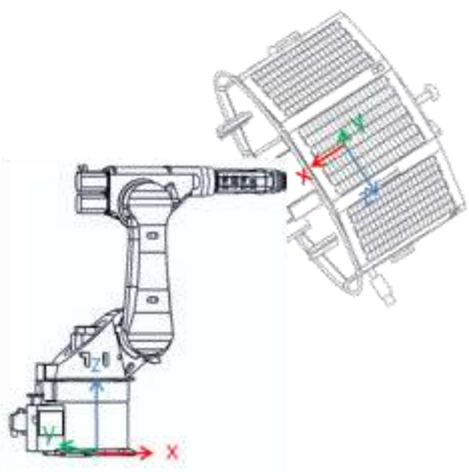
12D/6D – KUKA 3D

Reduction of DOF and Coordinate transformations

12D/9D – Client Fix

- KUKA
  does only rotations
- CableRobot
  does only translocations





- ❐ **Manually place client in WCS**
- ❐ **Calculate Client orientation**

$$R_{Client}^{Exploration\ hall} = R_{Servicer}^{Exploration\ hall} \cdot R_{Client}^{Servicer}$$

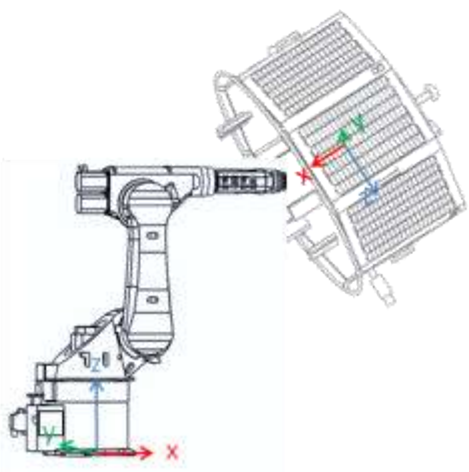
- ❐ **Calculate Servicer position**

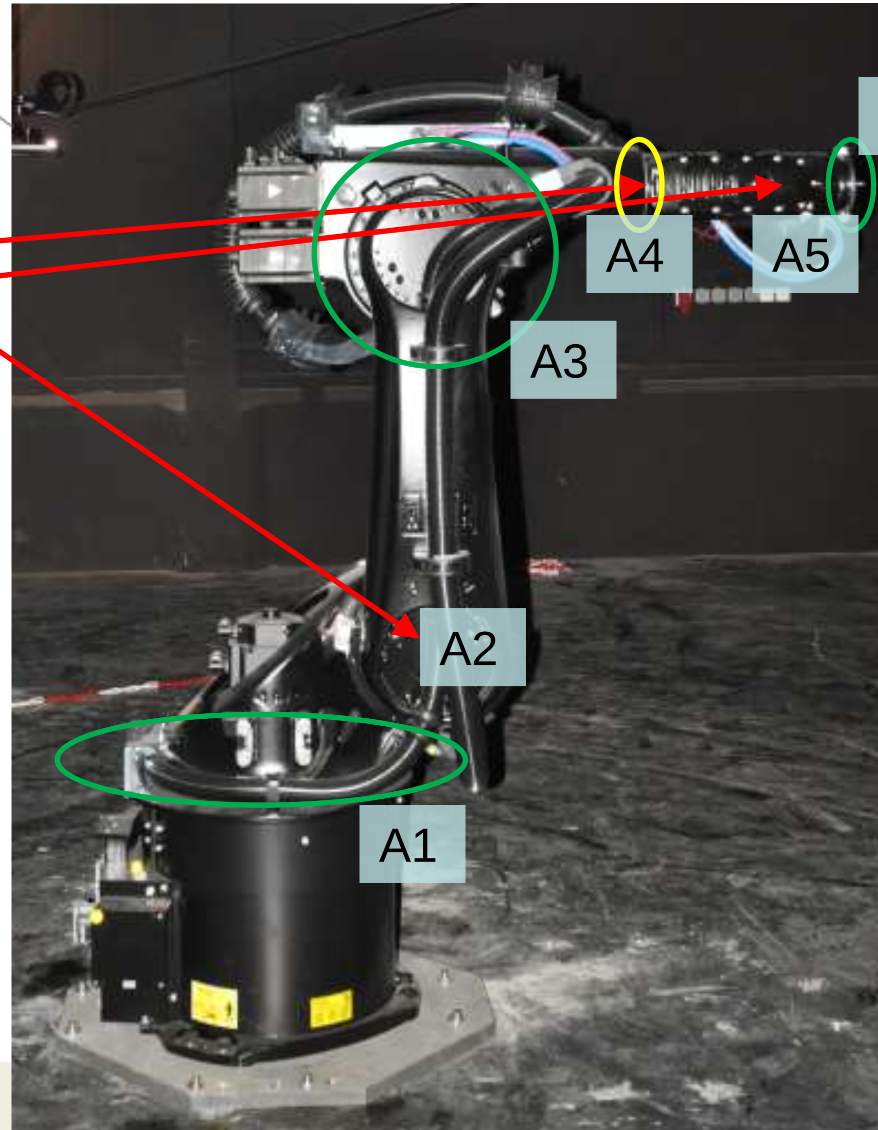
$$P_{Servicer}^{Exploration\ hall} = P_{Client}^{Exploration\ hall} - R_{Servicer}^{Exploration\ hall} \cdot P_{Client}^{Servicer}$$

Reduction of DOF and Coordinate transformations

12D/6D – KUKA 3D

- KUKA
  - A2 = -90°
  - A4=90°
  - A5=0°
- Less load
- More translocation





ate Client
ation

ation hall · $R^{Servicer}_{Client}$

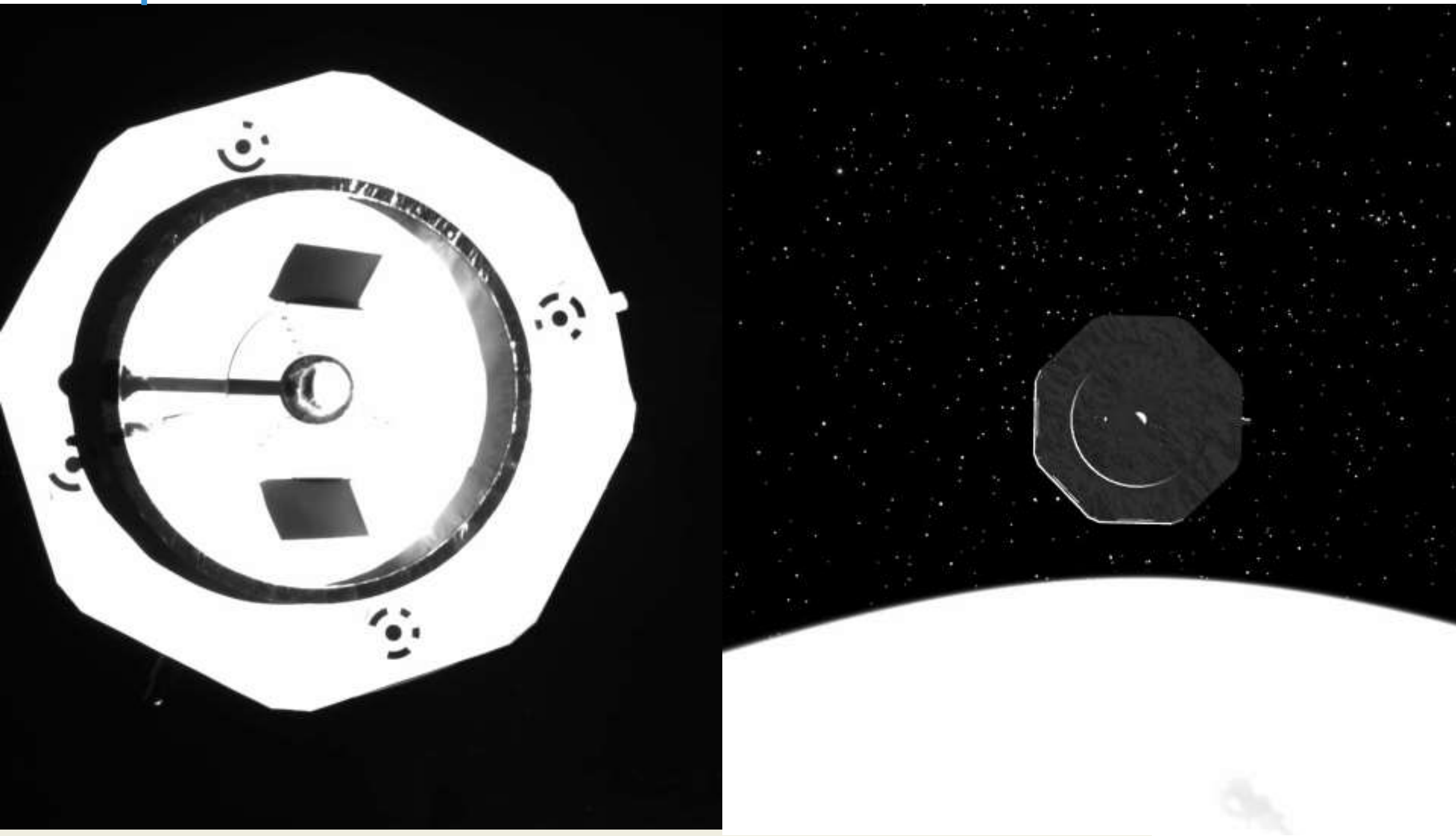
ate Client
on

$+ P^{Exploration\ Hall}_{Kuka}$

ate Servicer
on

ation hall · $P^{Servicer}_{Client}$

Simulated and real images in comparison

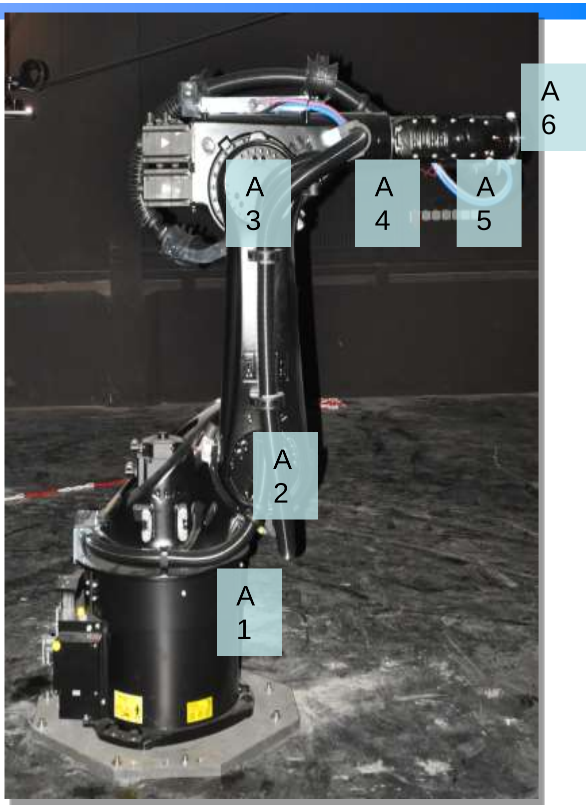


- **Questions so far?**

- **Collision avoidance**

Collision avoidance

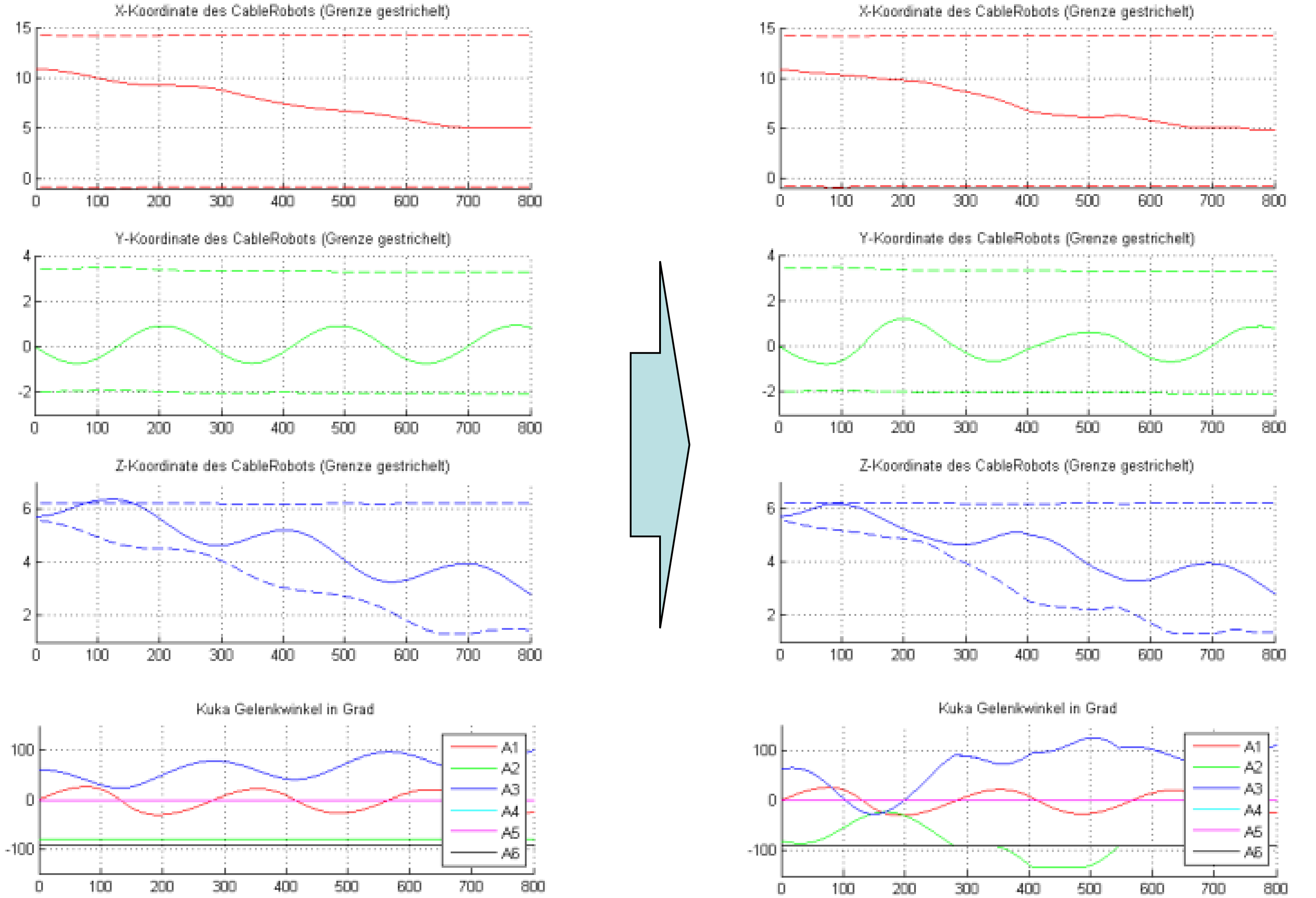
- **Maximum use of workspace**
  - A2/A3 automatic
    - For 12D/6D – KUKA 3D
    - Increases vertical workspace of the CableRobot
  - A4/A6 automatic
    - For 12D/6D – KUKA 3D
    - Delays the necessity to manually reconfigure the HIL-simulation, e.g. when doing tumbling Client motions
  - Z axis control
    - Increases maximum approach distance by using the diagonal of the workspace
    - Enables compensating unwanted rotations of the CableRobot



- **Detection of possible collisions**
  - Servicer – environment
  - Client – environment
  - Client – Servicer
  - Client– KUKA

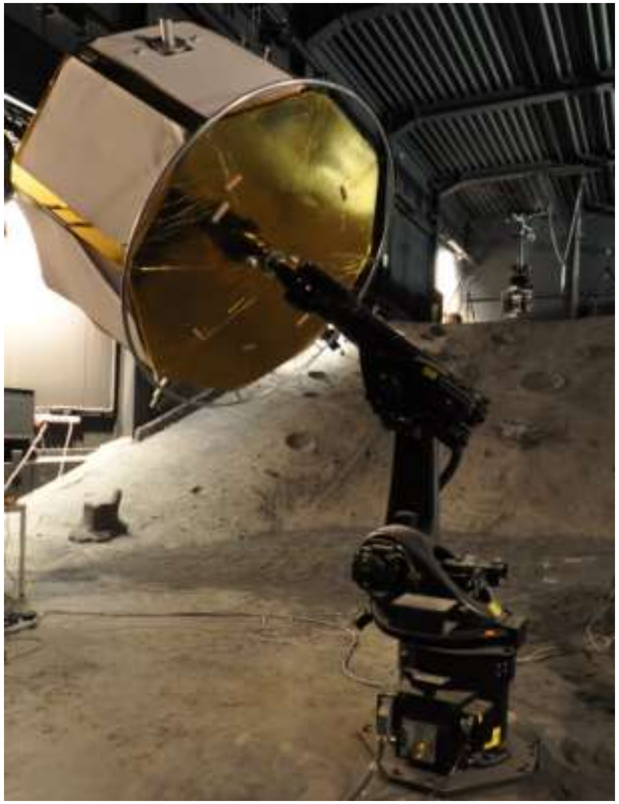
Collision avoidance

- A2/A3 automatic



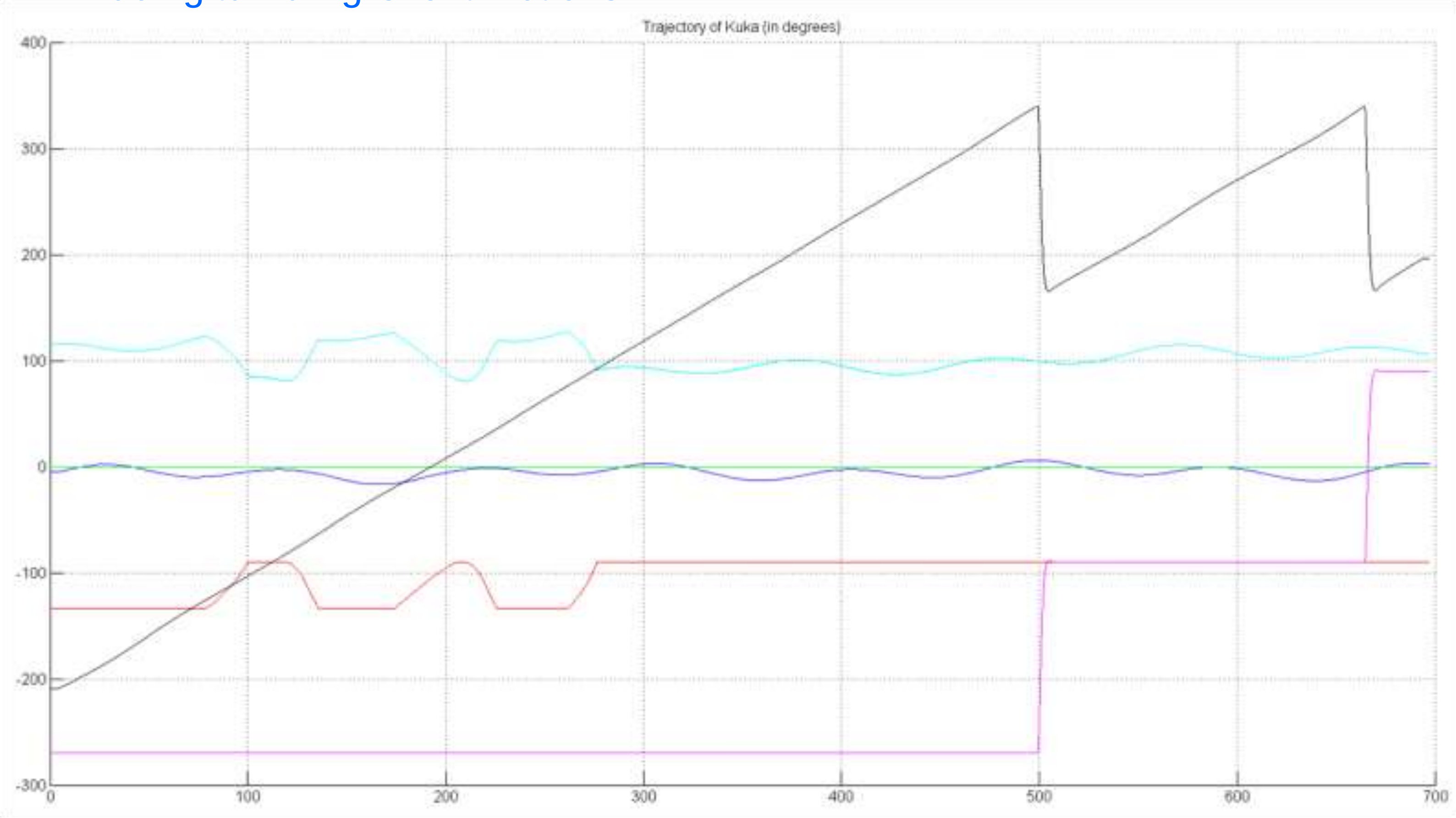
Collision avoidance

- **A2/A3 automatic**
  - Increases vertical workspace of the CableRobot



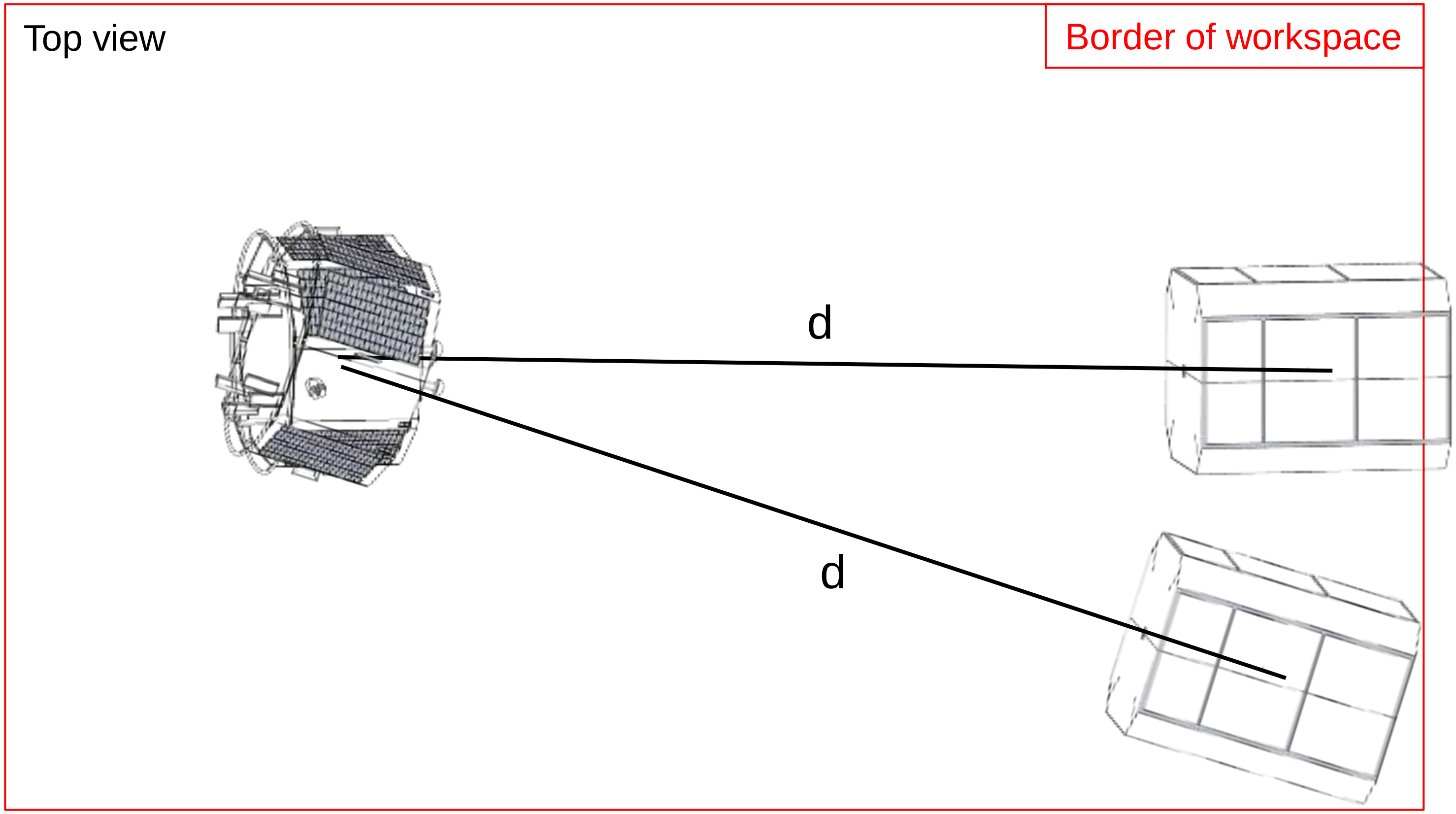
Collision avoidance

- **A4/A6 Automatik**
  - Delays the necessity to manually reconfigure the HIL-simulation, e.g. when doing tumbling Client motions



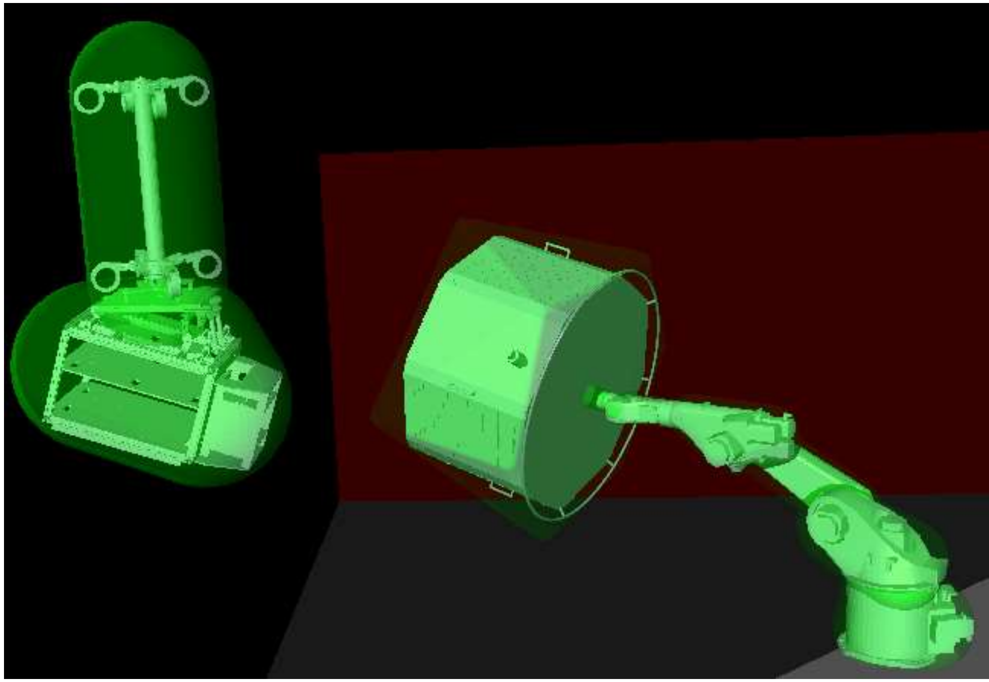
Collision avoidance

- **Z axis control**
  - Increases approach distance
  - Reduces necessary movement of CableRobot



Collision avoidance

- **Detection of possible collisions**
  - Simplifies the complex CAD data to convex hulls
    - Low loss in precision
    - Enables collision detection in real-time
  - Send current KUKA angles and CableRobot position via TCP/IP
  - Calculate the positions of all volumes
  - Calculate the minimal distance between all volumes
  - Return minimal distance and the involved objects
  - Stop all movements if necessary
  - Different thresholds possible formanual and automatic control
  - Timeout monitoring for safety

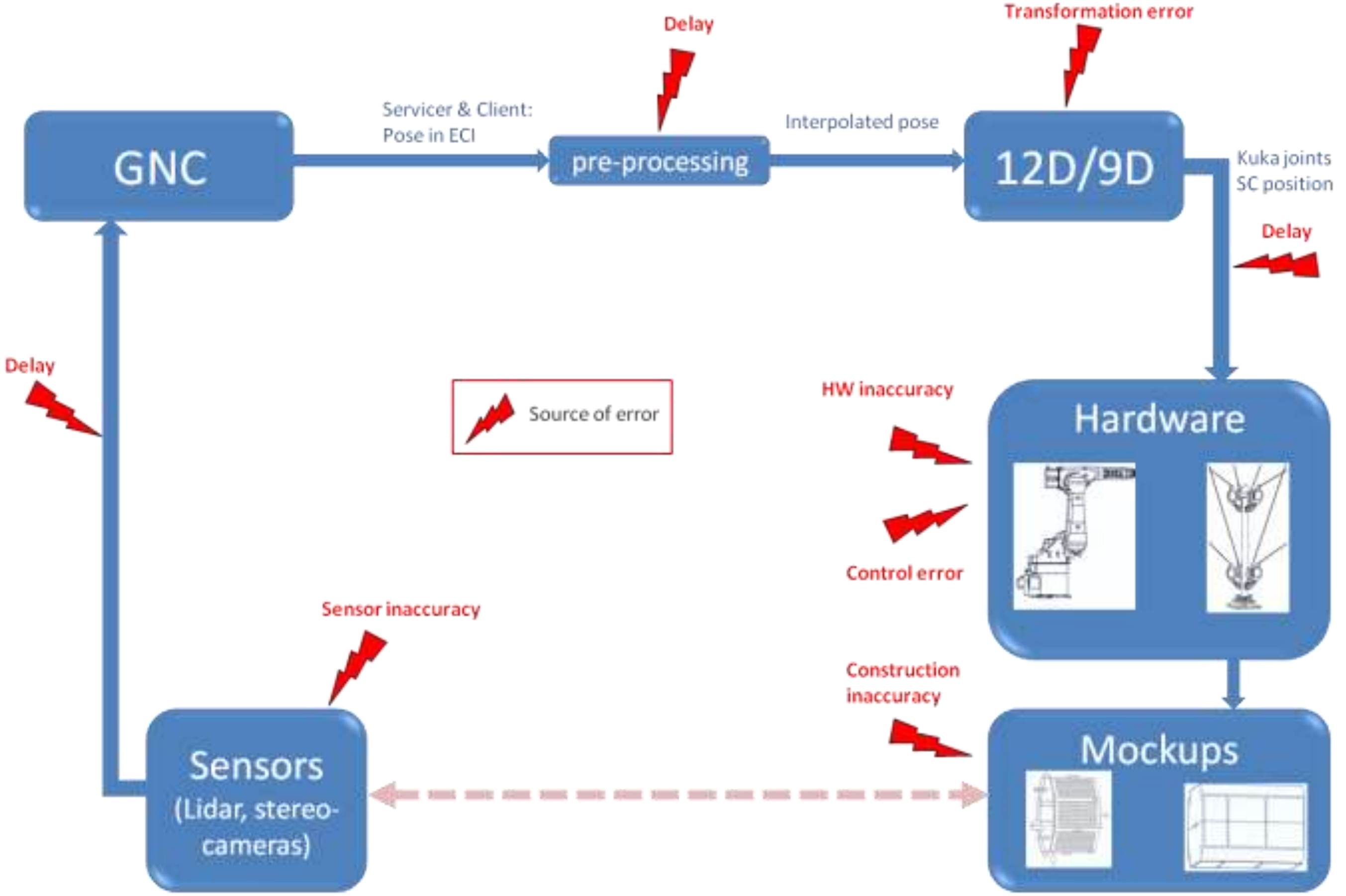


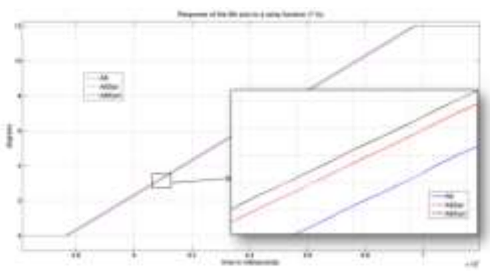
- **Questions so far?**

- **System accuracy**

System accuracy

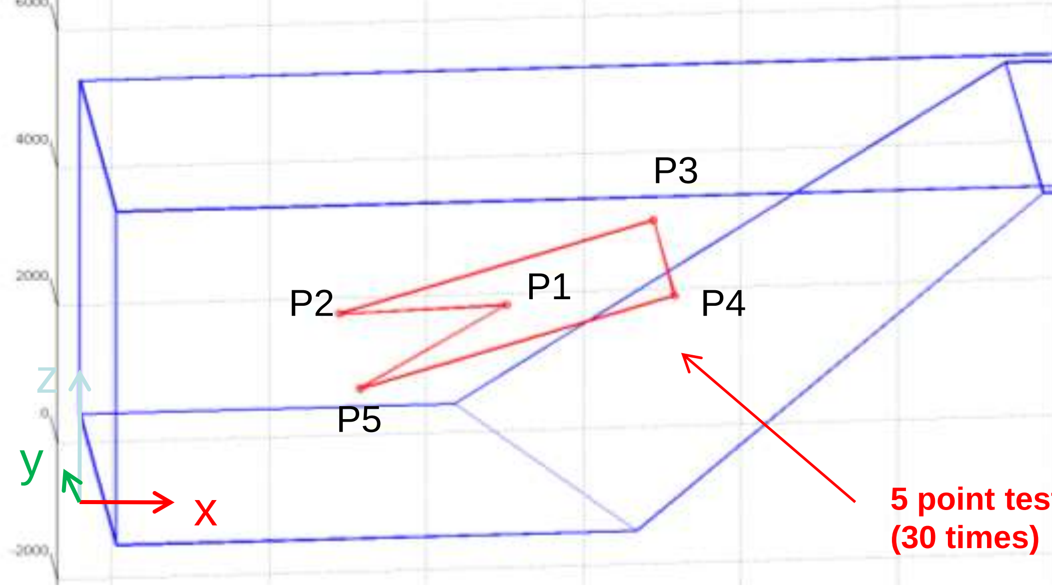
- **Possible error sources**

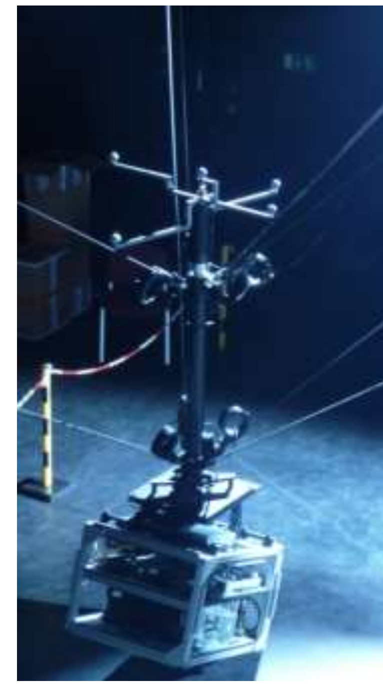




System accuracy

- **Measurement of repeatability and absolute accuracy**
  - Using Norm ISO 9283
  - Laser tracker as reference measurement

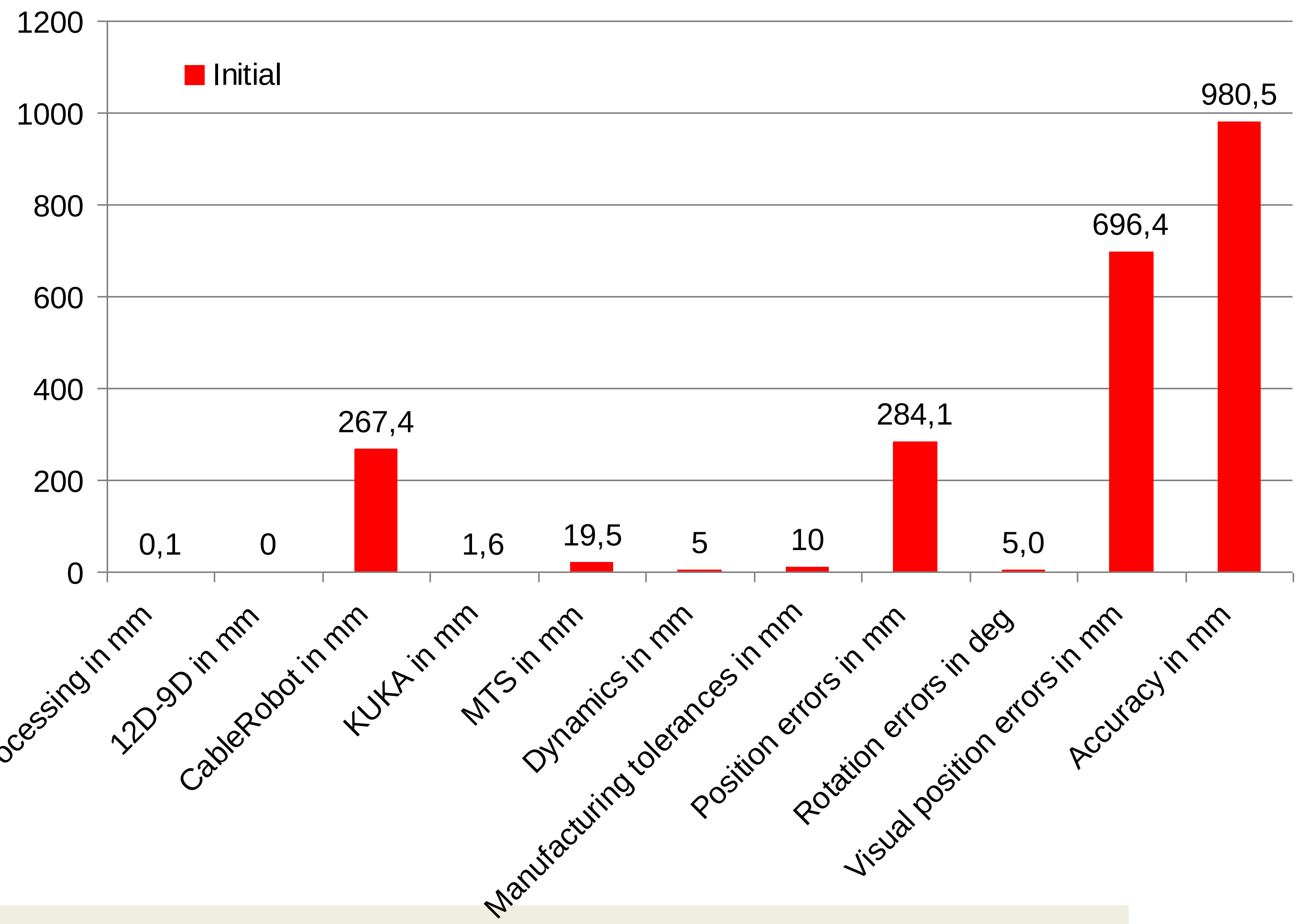




	Absolute accuracy	Repeatability
KUKA without correction	1.6 mm	0.01 mm
CableRobot without correction	267.4 mm	1.1 mm
MTS without correction	19,5 mm	0.6 mm

System accuracy

❒ Resulting accuracy

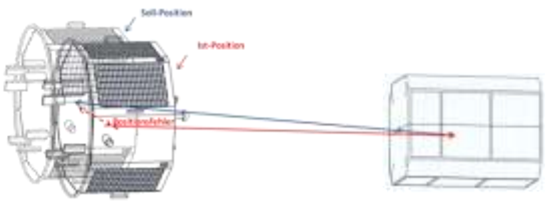


Absolute accuracy

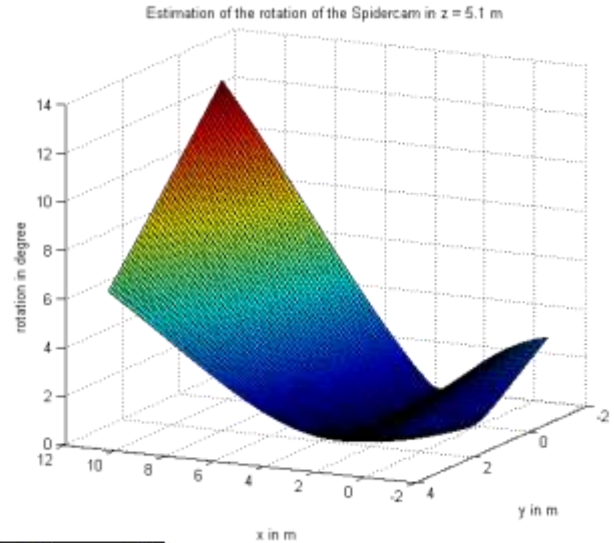
System accuracy

❒ Visual position error

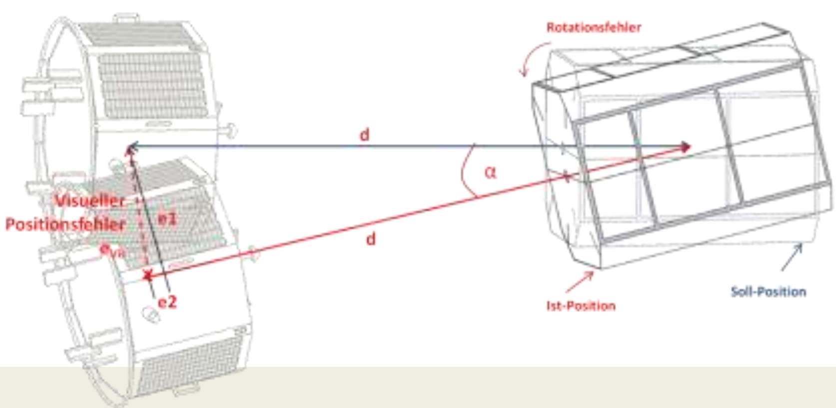
Position error



Rotation error



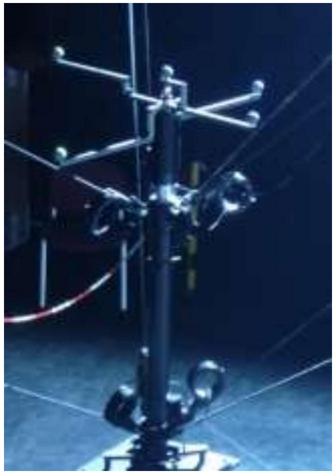
Visual position error

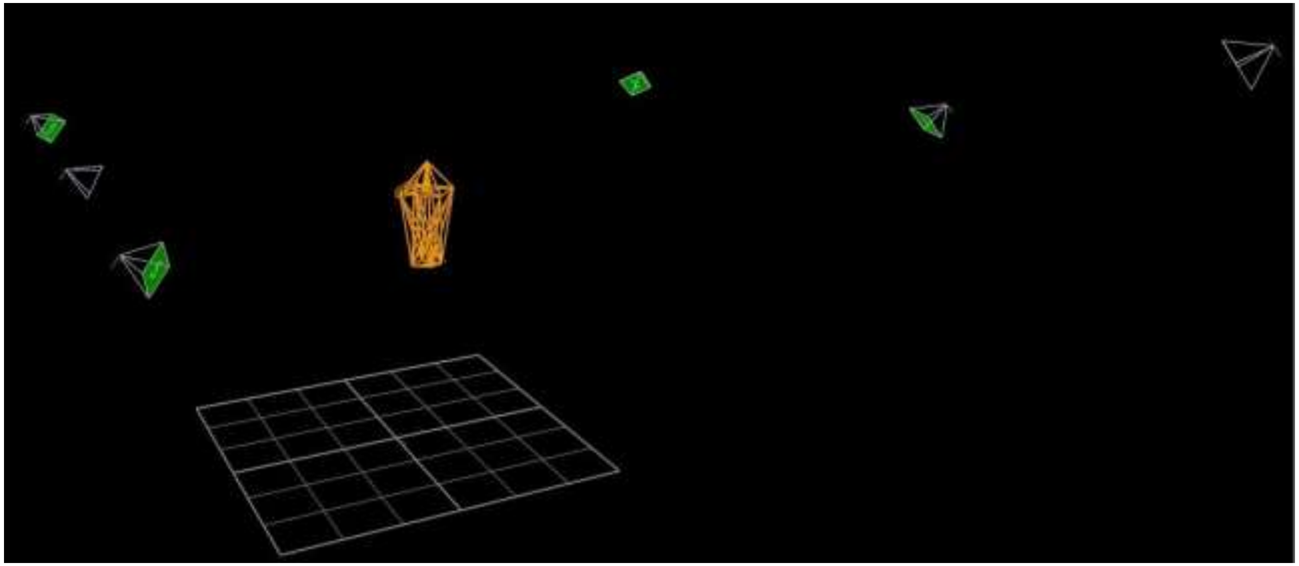


$$
\begin{aligned}
e_{Vis} &= \sqrt{e1^2 + e2^2} \\
&= \sqrt{(d \cdot sin(\alpha))^2 + (d - d \cdot cos(\alpha))^2} \\
&= d \cdot \sqrt{2 \cdot (1 - cos(\alpha))}
\end{aligned}
$$

System accuracy

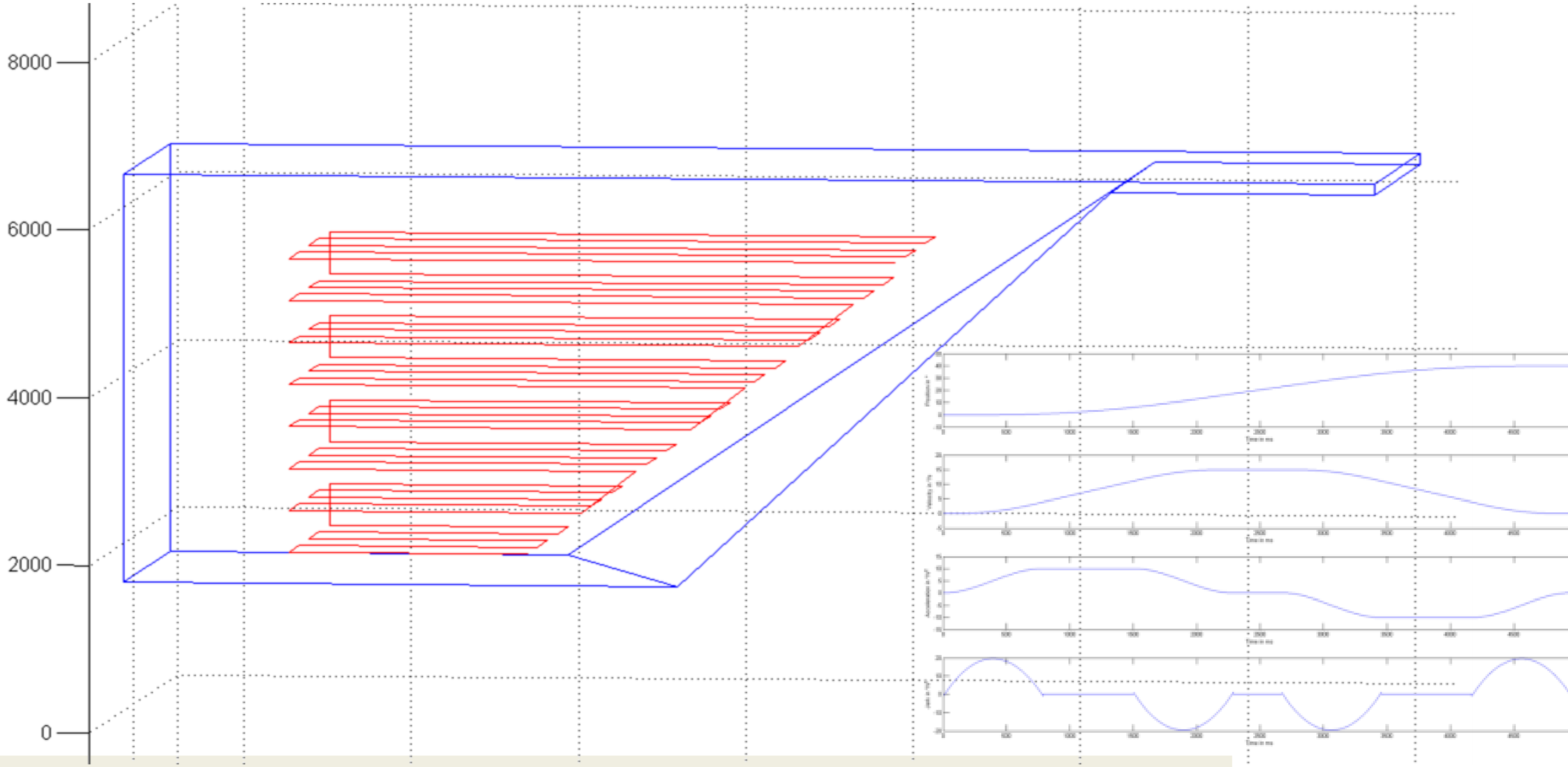
- **Improvements to the motion tracking system**
  - Changed the calibration object
  - Manufactured a new coordinate origin setup
  - Using bigger markers
  - Bought additional camara with 16 times the resolution
  - Optimized camera positions and directions
  - Using more markers





System accuracy

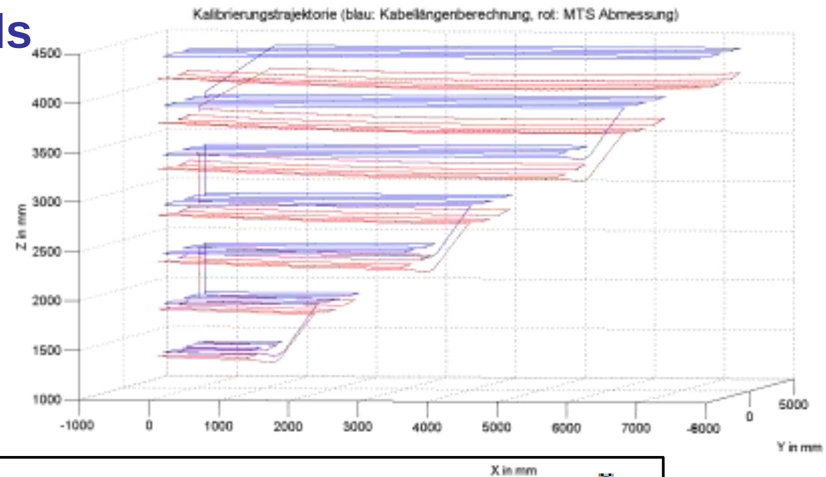
- **Calculating correction polynomials**
  - Calibration trajectory
  - Laser Tracker as reference measurement device



System accuracy

❐ Calculation of correction polynomials

- 0.5 million measurement values
- Repeatability very good
- Calculated polynomial of 3rd degree



$$\varepsilon_x = x - x_{laser} = [x^3\ x^2y\ x^2z\ xy^2\ xyz\ xz^2\ y^3\ y^2z\ yz^2\ z^3\ x^2\ xy\ xz\ y^2\ yz\ z^2\ x\ y\ z\ 1] \begin{bmatrix} p_{19}^x \\ \cdots \\ p_0^x \end{bmatrix}$$

$$\varepsilon_y = y - y_{laser} = [x^3\ x^2y\ x^2z\ xy^2\ xyz\ xz^2\ y^3\ y^2z\ yz^2\ z^3\ x^2\ xy\ xz\ y^2\ yz\ z^2\ x\ y\ z\ 1] \begin{bmatrix} p_{19}^y \\ \cdots \\ p_0^y \end{bmatrix}$$

$$\varepsilon_z = z - z_{laser} = [x^3\ x^2y\ x^2z\ xy^2\ xyz\ xz^2\ y^3\ y^2z\ yz^2\ z^3\ x^2\ xy\ xz\ y^2\ yz\ z^2\ x\ y\ z\ 1] \begin{bmatrix} p_{19}^z \\ \cdots \\ p_0^z \end{bmatrix}$$

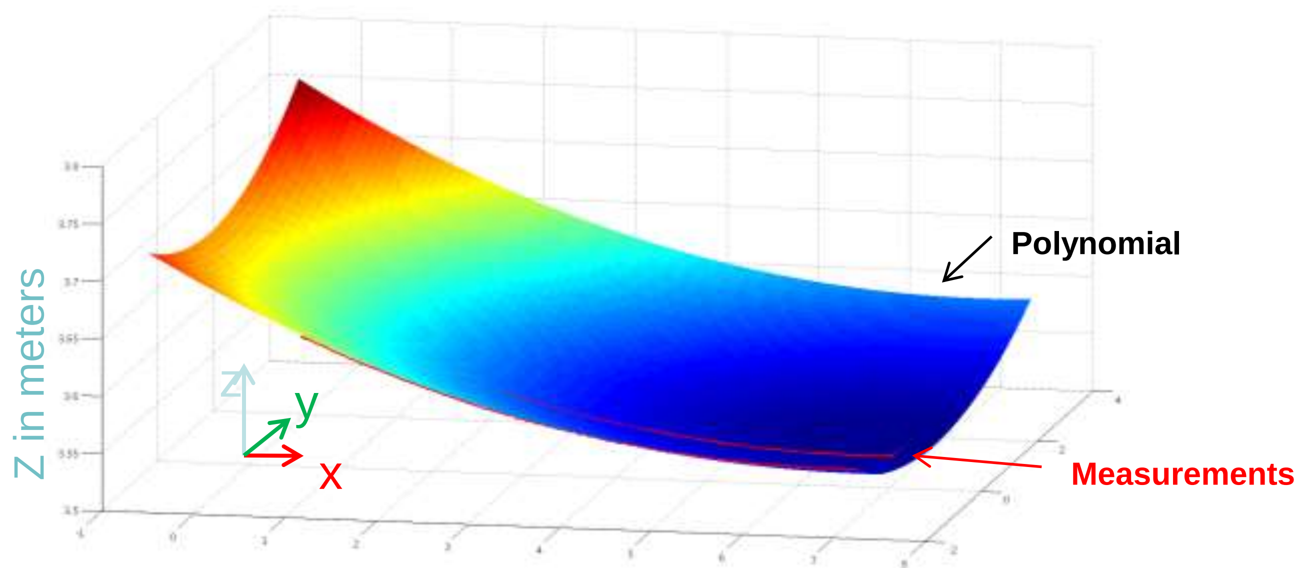
Position error

x, y and z are the coordinates of the CableRobot

20 polynomial parameters

System accuracy

❒ Calculation of correction polynomials



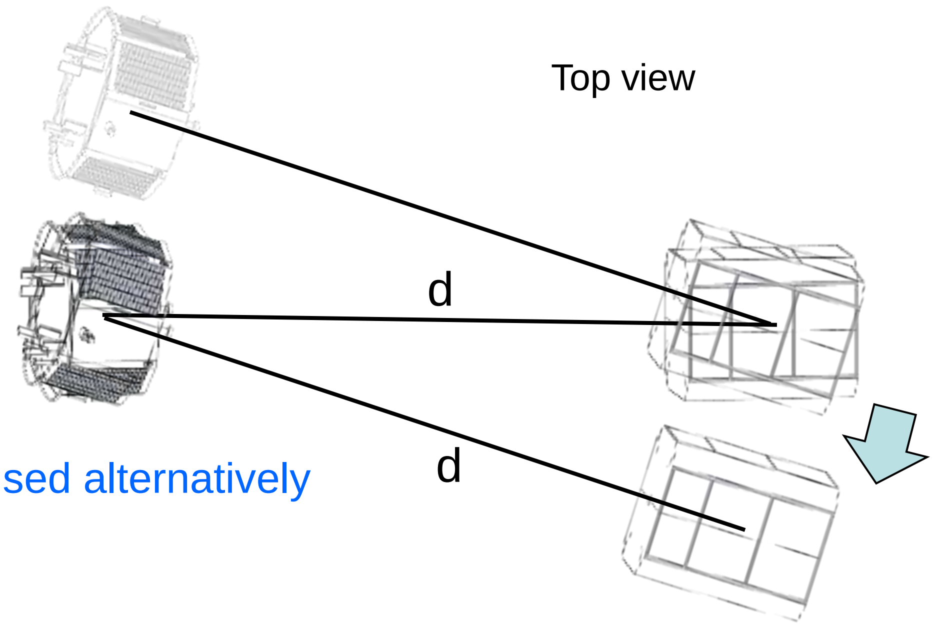
Plane z=3,95m

	Absolute accuracy	Repeatability
KUKA without correction	1.6 mm	0.01 mm
CableRobot without correction	267.4 mm	1.1 mm
CableRobot with correction	6.6 mm	1.1 mm
MTS without correction	8.9 mm	0.6 mm
MTS with correction	5.3 mm	0.6 mm

System accuracy

❒ Rotation compensation

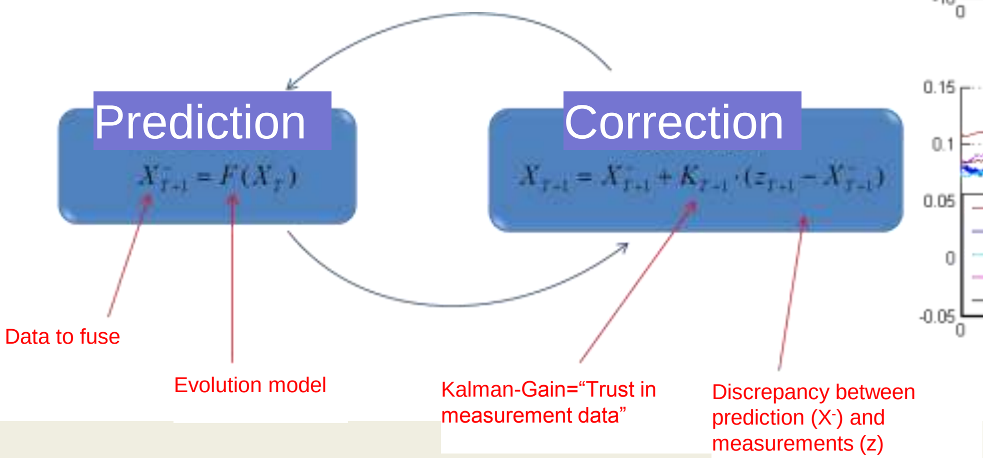
- Takes filtered rotation values from CableRobot and uses it as additional parameter for the DOF reduction
- CableRobot and KUKA do compensation movements
- Results in new rotation value
  $\Rightarrow$ New correction
  - Converges, stable
  - possible vibrations
- Z axis of the CableRobot can be used alternatively

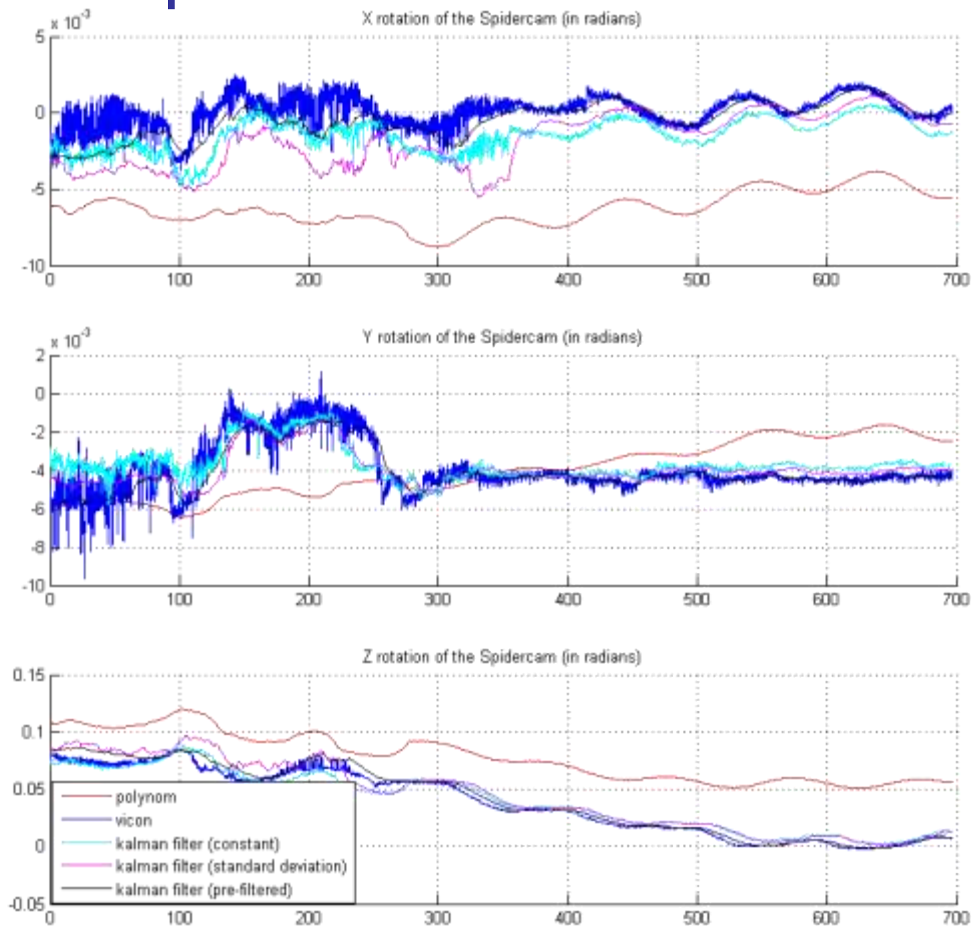


System accuracy

❒ Integrating the MTS into the control loop

- Necessary as installation deforms
- Sensor fusion necessary
  - Pose of CableRobot
  - Pose from correction polynomial
- Kalman-Filter
  - Trust in measurements according to quality
  - Median pre-filtering

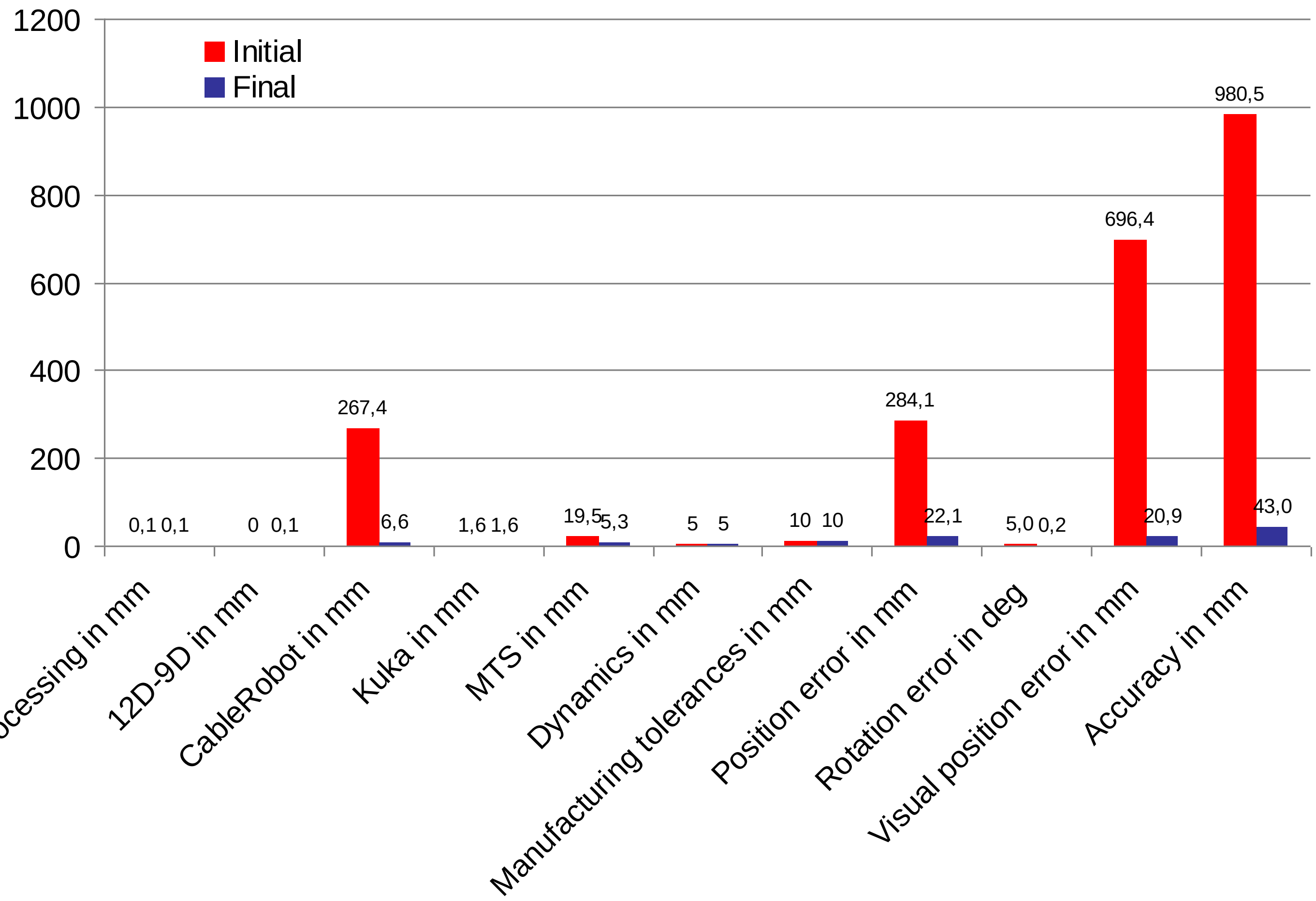




Universität Bremen

System accuracy

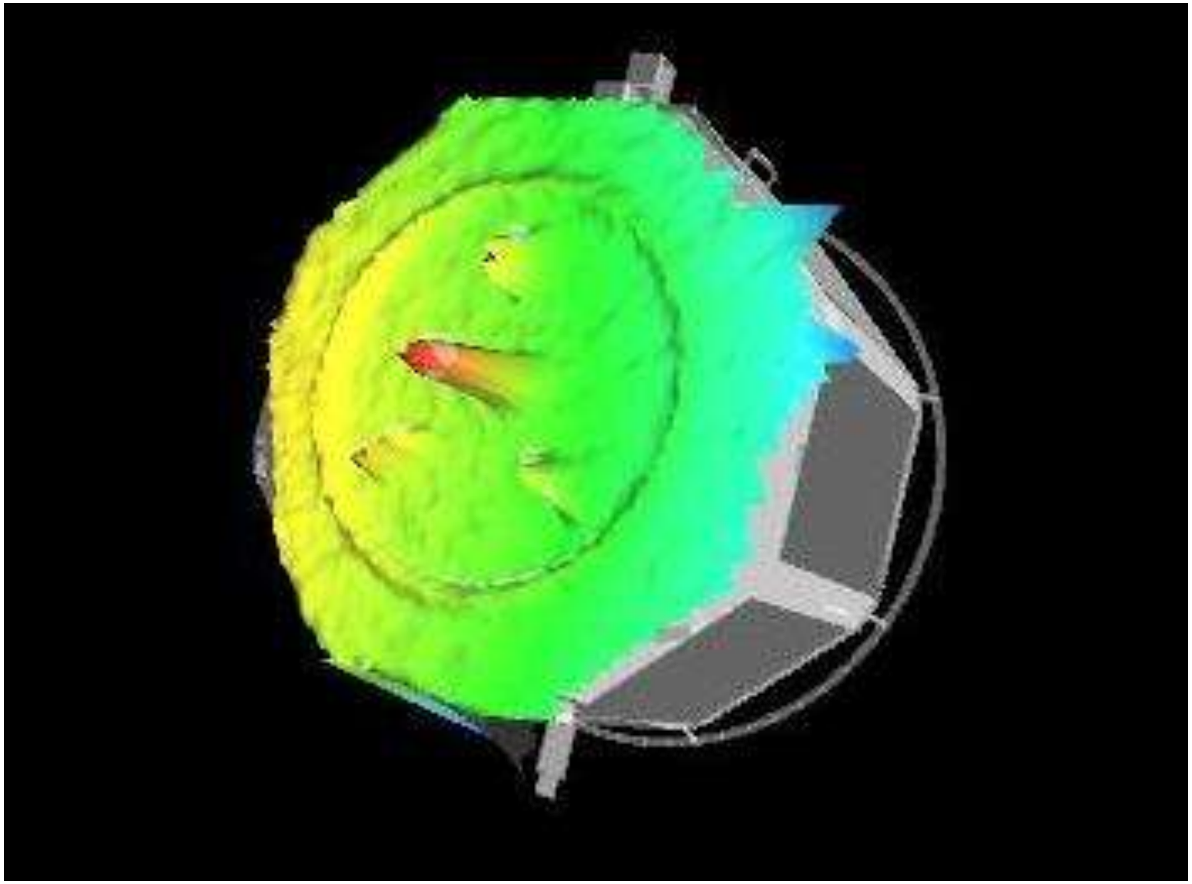
❒ Resulting accuracy



Universität Bremen

System accuracy

- **Final experiments**
  - Qualitative comparison between LIDAR and ground truth

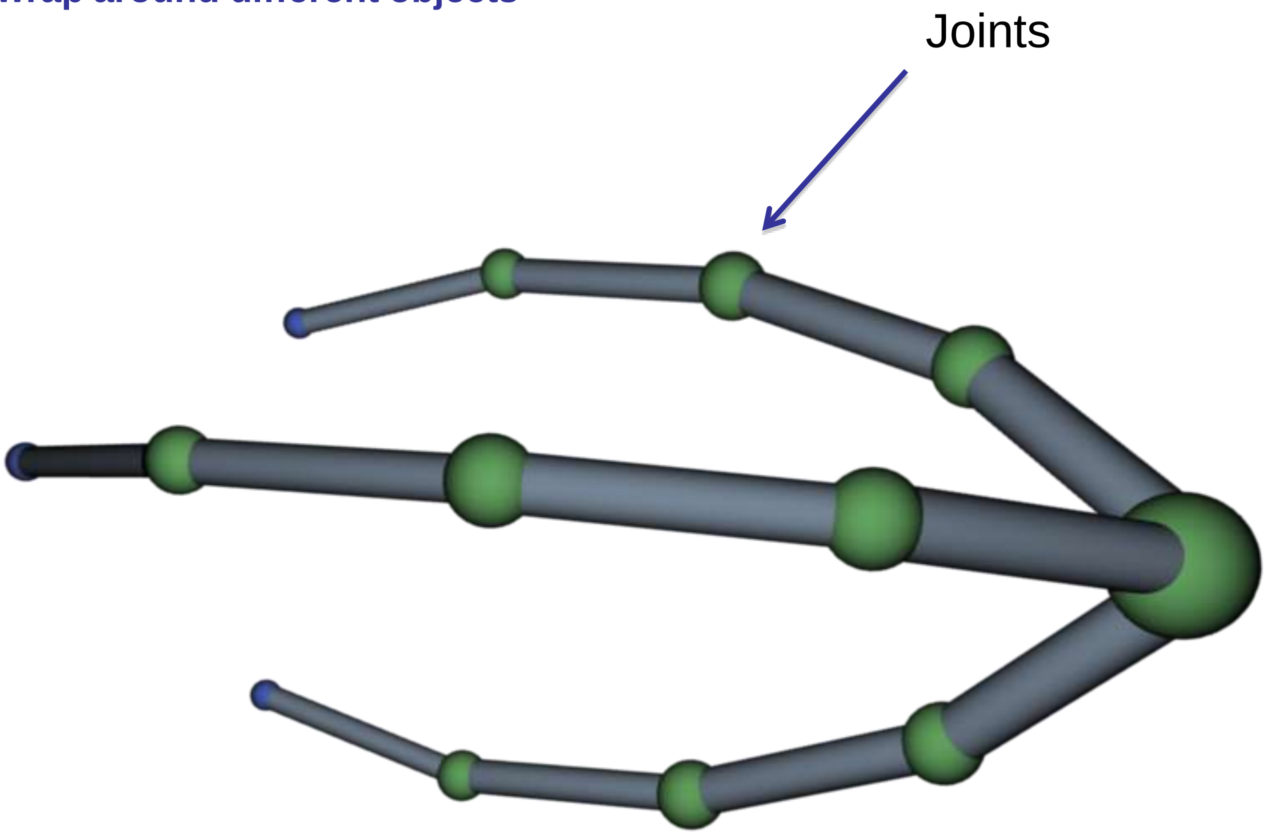


- **Questions so far?**

- **Alternative capturing systems**

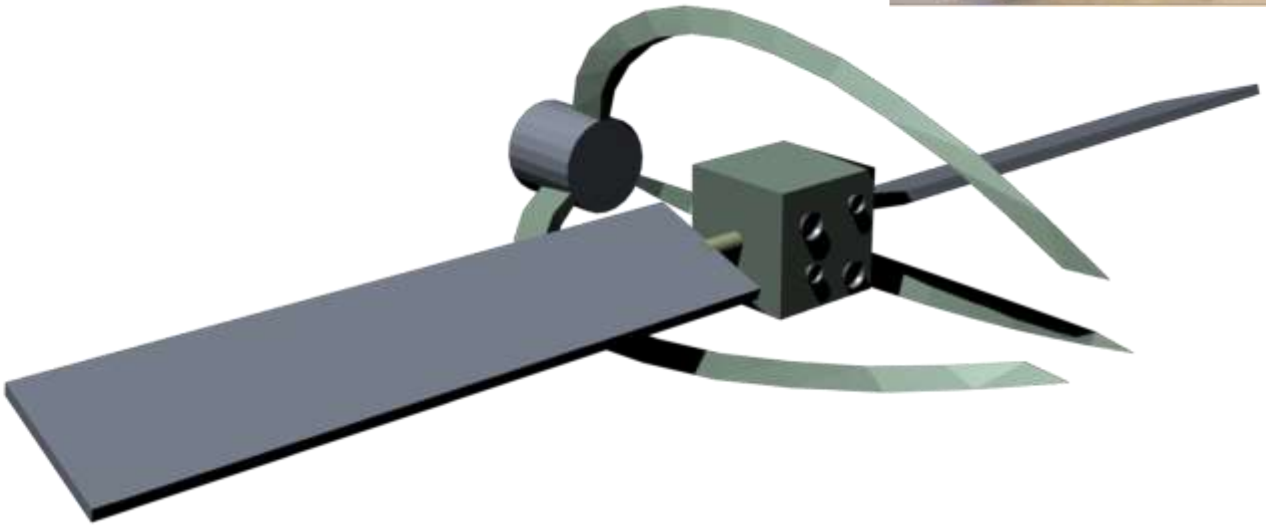
Concept 1

- Octopus arms
  - Several joints per arm
  - Arms can wrap around different objects (satellites)



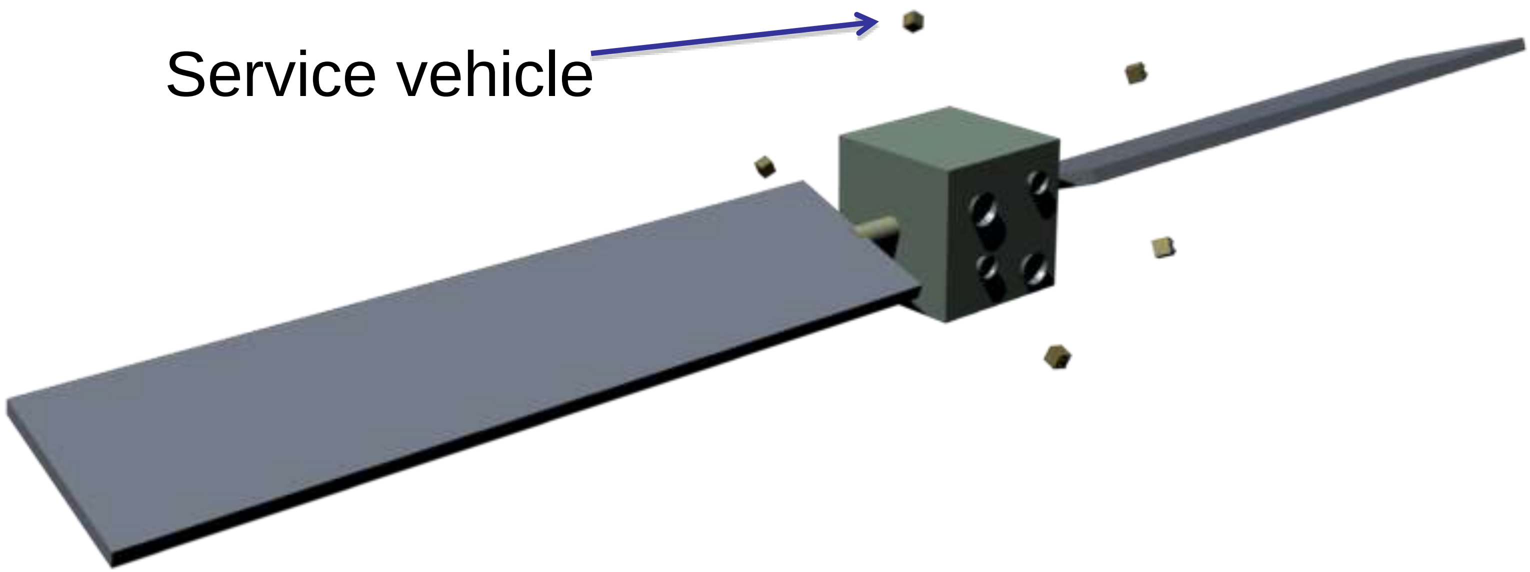
Concept 2

- One big gripper
  - Only one joint on the servicer
  - In the gripper flexible structures similar to the flexible DLR wheels could be used to adapt to objects



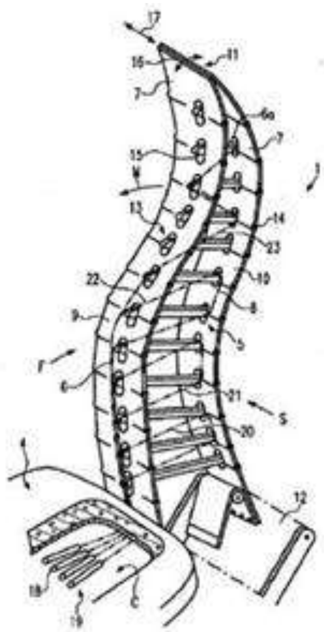
Concept 3

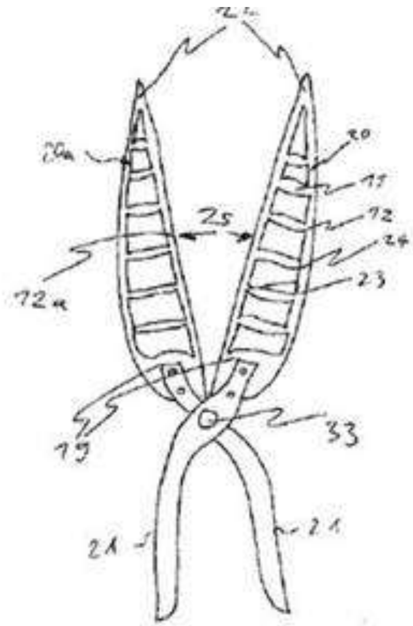
- **Swarm of service-vehicles**
  - **The vehicles can dock to defined spots on the satellite and apply thrust forces.**
  - **Concept needs swarm control.**



Concept 4

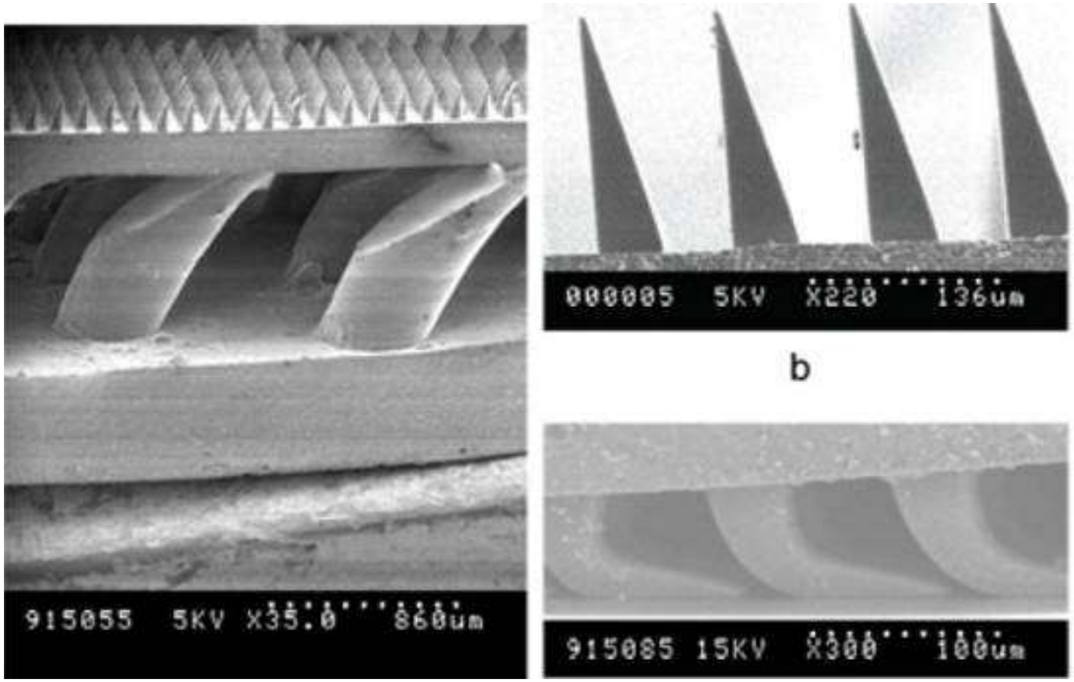
- **Gripper structures using Fin Ray effect**
  - **The Fin Ray effect causes the gripper to passively close around an object by a contact on the inner side.**
  - **This concept could be combined with the other concepts.**



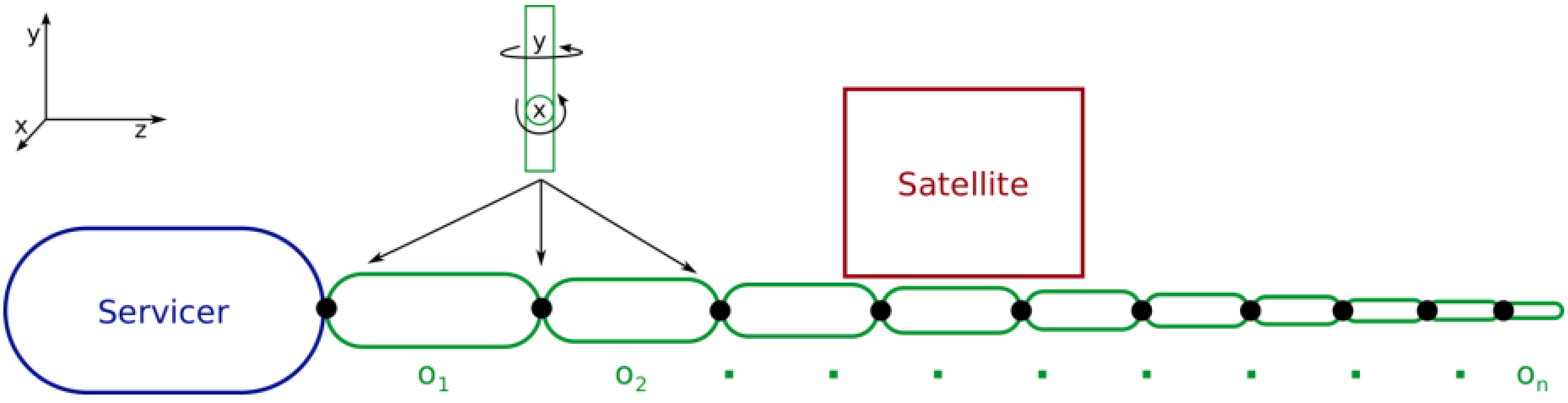


Concept 5

- **Stickybot**
  - **Using Van-Der-Waals forces contact is established on a microscopic level.**
  - **Enables gripping on flat surfaces.**



Arm optimization



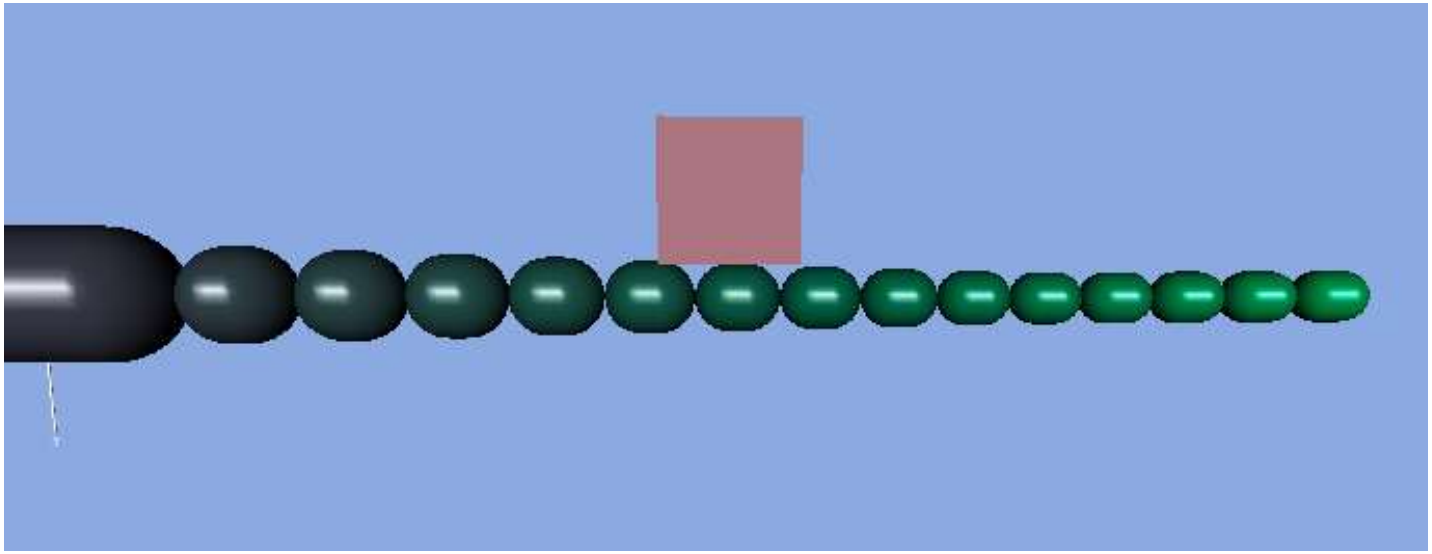
Parameters for optimization:

1. Number of elements
2. Length of first element
3. Radius of first element
4. Speed of first axis of first joint
5. Speed of second axis of first joint
6. Factor for the lengths
7. Factor for the radius
8. Factor for the speed of the first axis
9. Factor for the speed of the second axis
10. Number of not-used joints

Arm optimization

Setup:

1. **The arm is created in simulation.**
2. **The evaluation is aborted if the arm does not reach the box.**
3. **The joints are activated for eight simulated minutes.**
4. **Then, eight further minutes a force is applied to the box.**
5. **The evaluation is based on the distance of the box to its starting point and the sum of all collision forces between arm and box.**

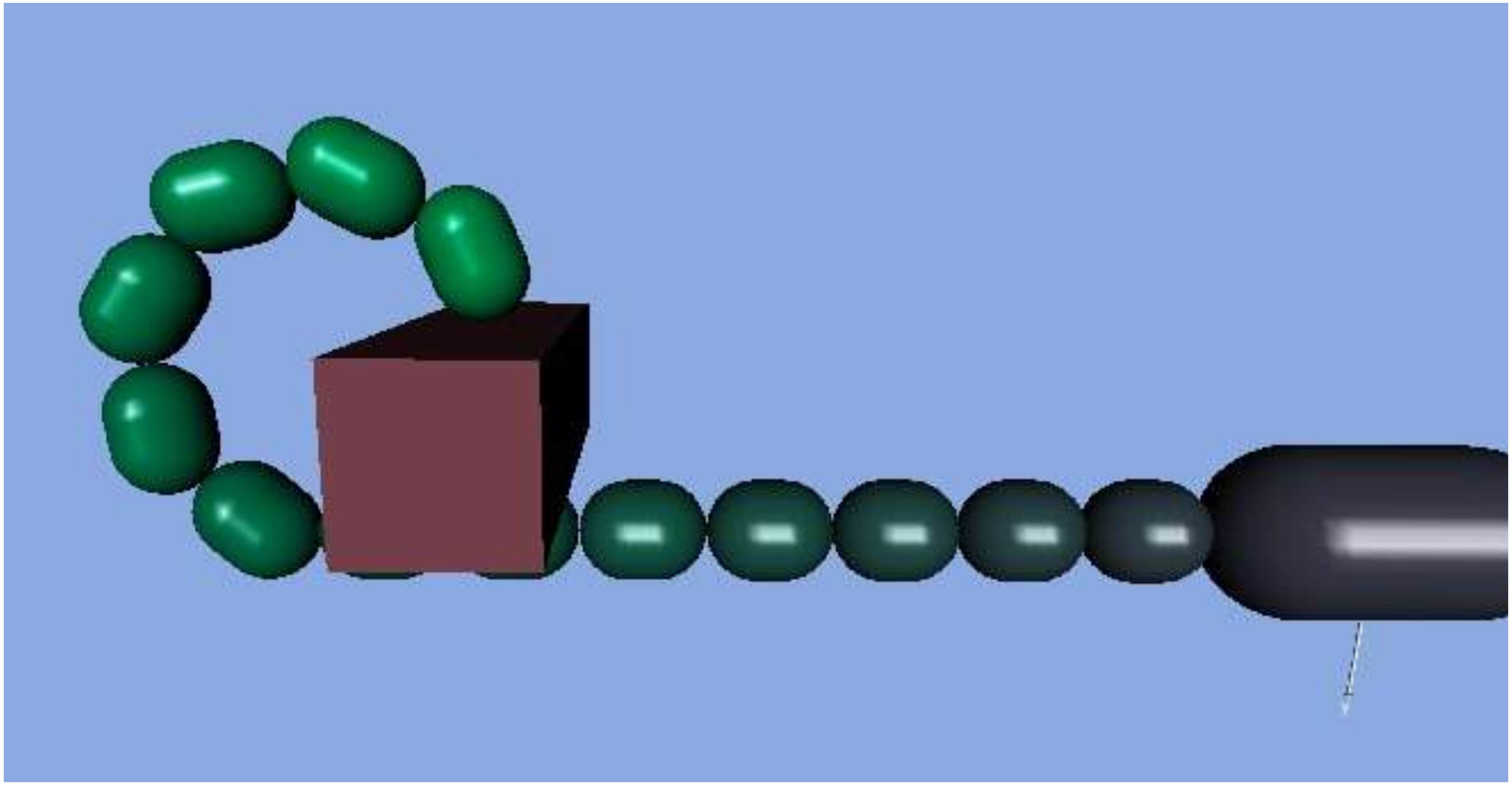


Arm optimization

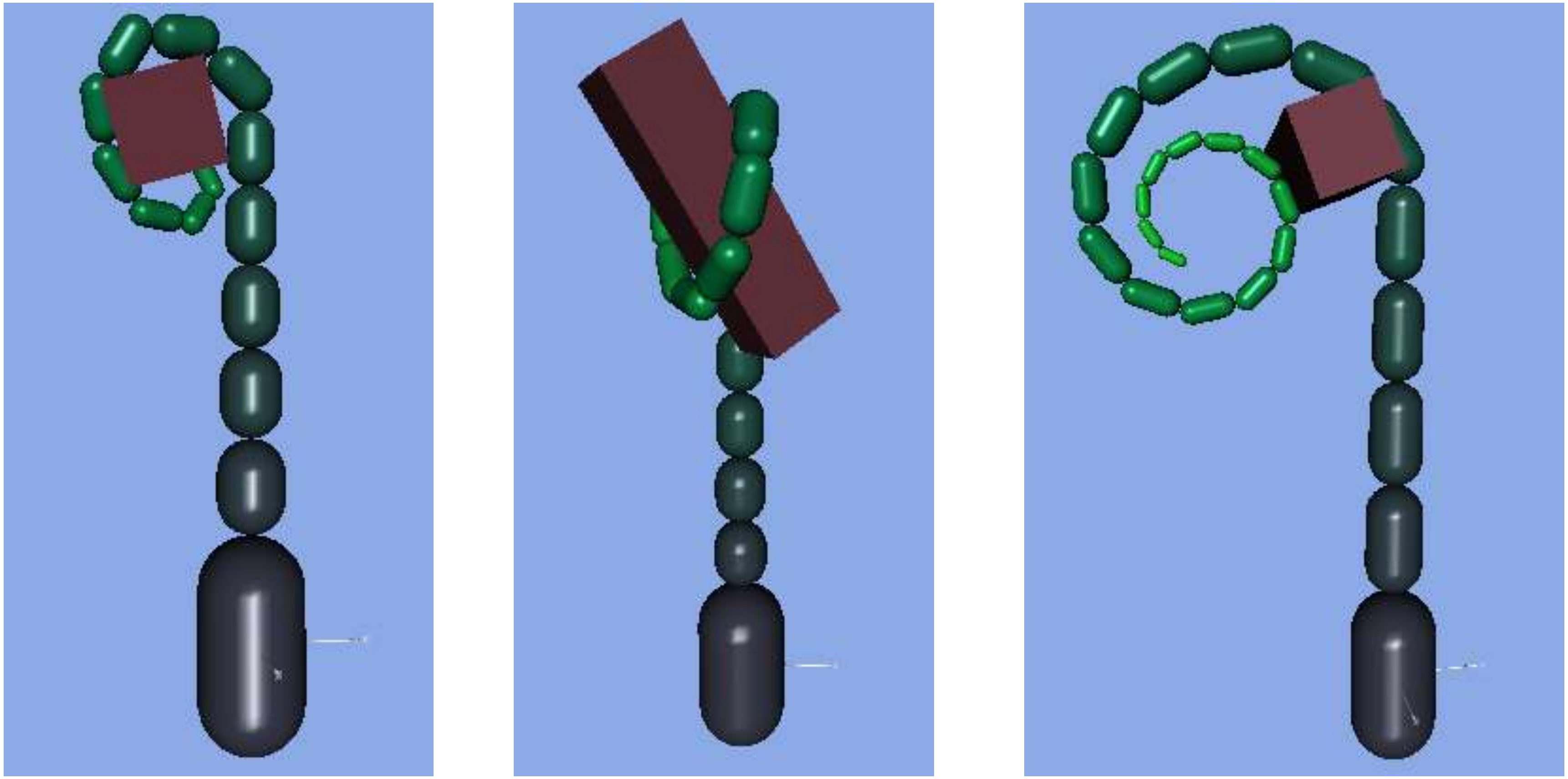
- 70 independent optimizations done
- Evaluation of 448.000 parameter configurations.
- Classification of the results using Growing Cell Structures.
- Divided the 70 results into 12 Classes.
- The three classes with the most results all represent a similar solution, where the arm does not wrap around the box but holds the box with the tip of the arm.
- These solutions additionally received the best average ratings.

Arm optimization

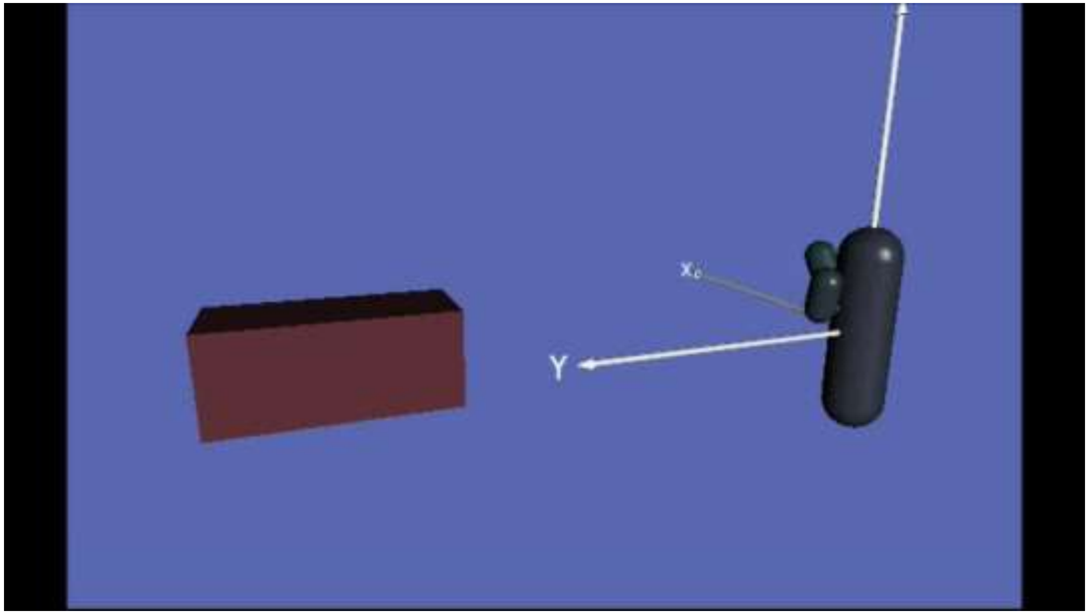
One of the best solutions for the arm optimization:

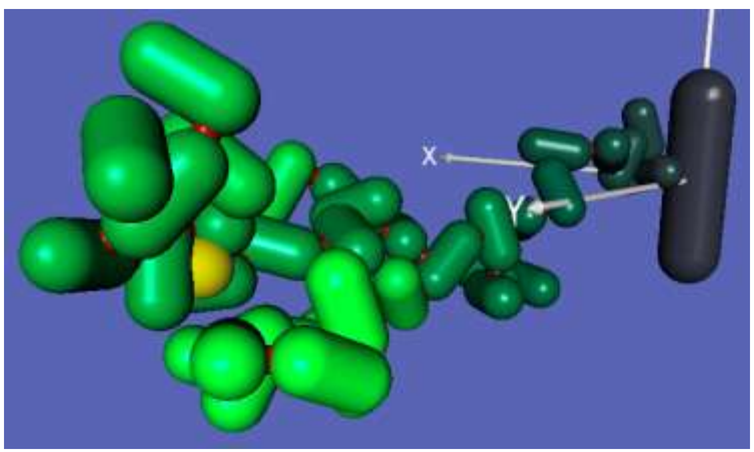


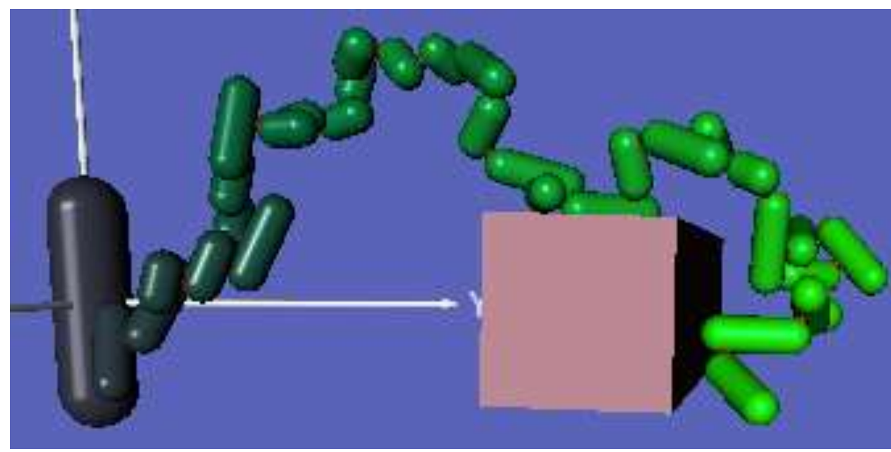
Other examples for solutions:



Structure optimization:





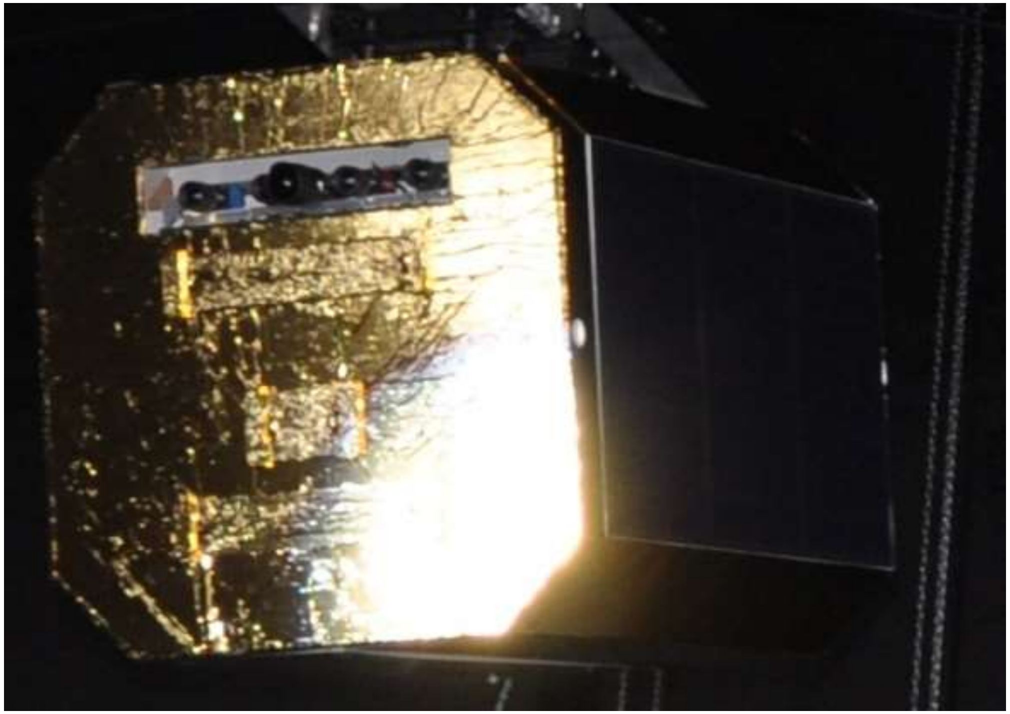


- **Questions so far?**

- **Camera systems and data preprocessing**

Camera systems and data preprocessing

- Overview camera
- Stereo camera System
- Data processing node
- Preprocessing
- Results



Camera systems and data preprocessing

- **Overview camera**
  - Used for monitoring
  - Prosilica GX 3300
    - 3296x2472 (8 MP) @ 17 FPS
    - Controlled like the stereo camera system
  - Nikkor Fisheye lens
    - 10,5 mm focal length
    - Opening angle of +/-55 °

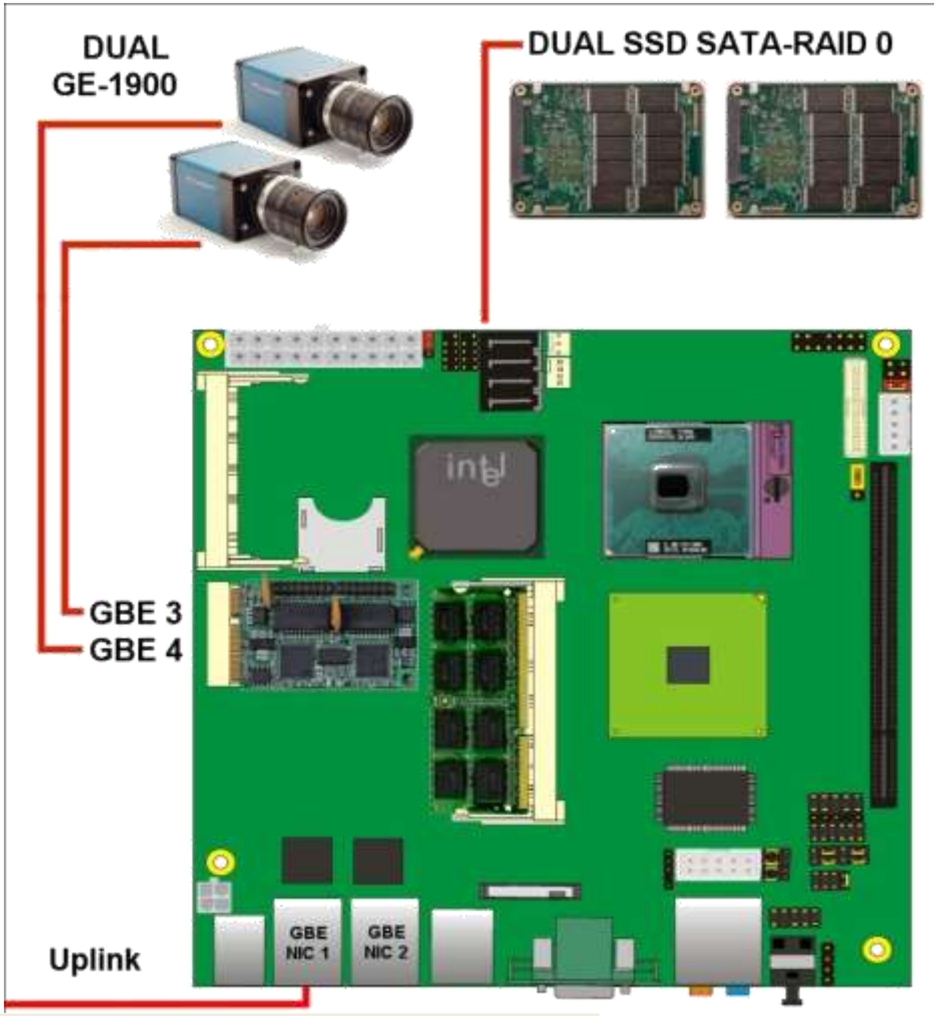
Camera systems and data preprocessing

- **Stereo camera system**
  - Based on the DEOS specification
  - 2 cameras, classic stereo configuration
  - 36,4 cm baseline
  - Image area 0.5 m - 17 m
  - GE1900 Gigabit Ethernet cameras
    - Lens with 12.5 mm focal length
    - Opening Angle of + / -17 °
    - Native resolution of 1920x1080 (HDTV)
    - Used resolution 1024x1024 (ROI)
    - Up to 30 frames / sec at full resolution
    - monochrome
    - High sensitivity
    - Hardware-synchronized

Camera systems and data preprocessing

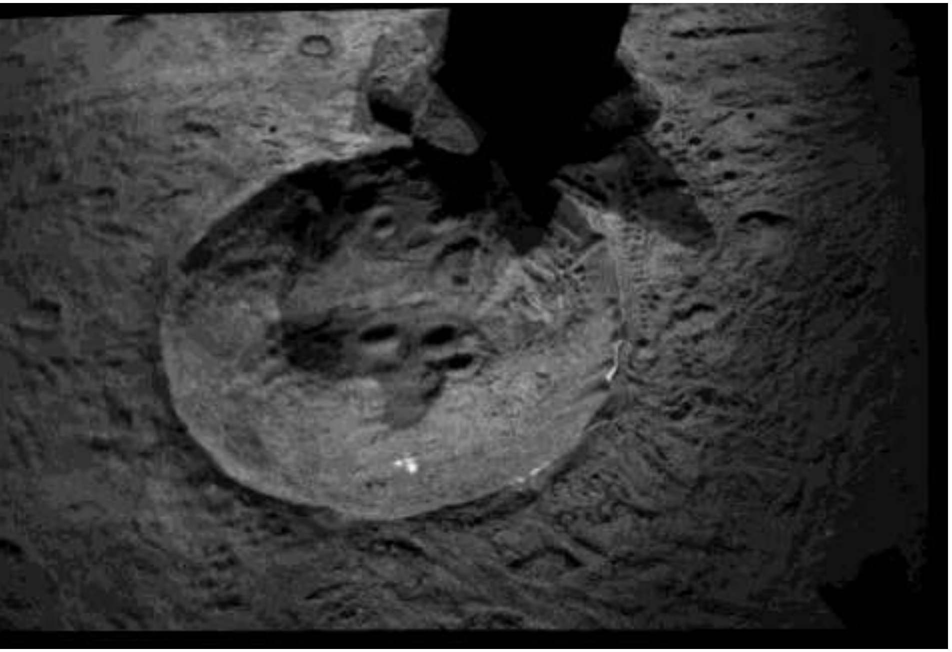
❒ Image processing node

- ➢ Camera system with dedicated data lines
- ➢ Uplink line for transmitting the pre-processed data
- ➢ Compact, high-performance PC
  - Four processor cores
  - SSD drives
  - Consistent x64 architecture
  - A total of approximately 50 W power
  - 17x17x5 cm space usage



Camera systems and data preprocessing

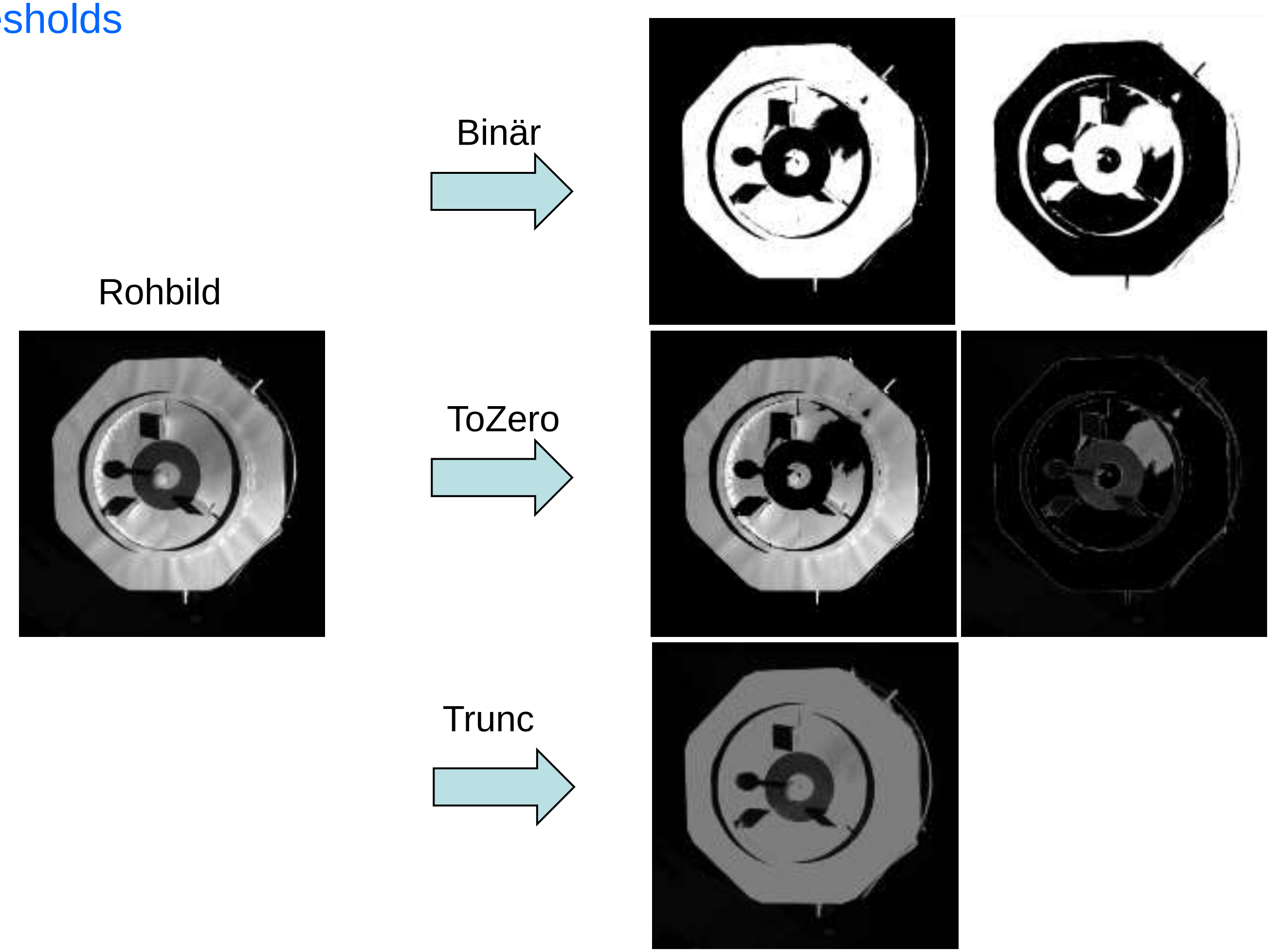
- **Preprocessing steps**
  - Acquiring image
  - Distortion correction
  - Rectification



Camera systems and data preprocessing

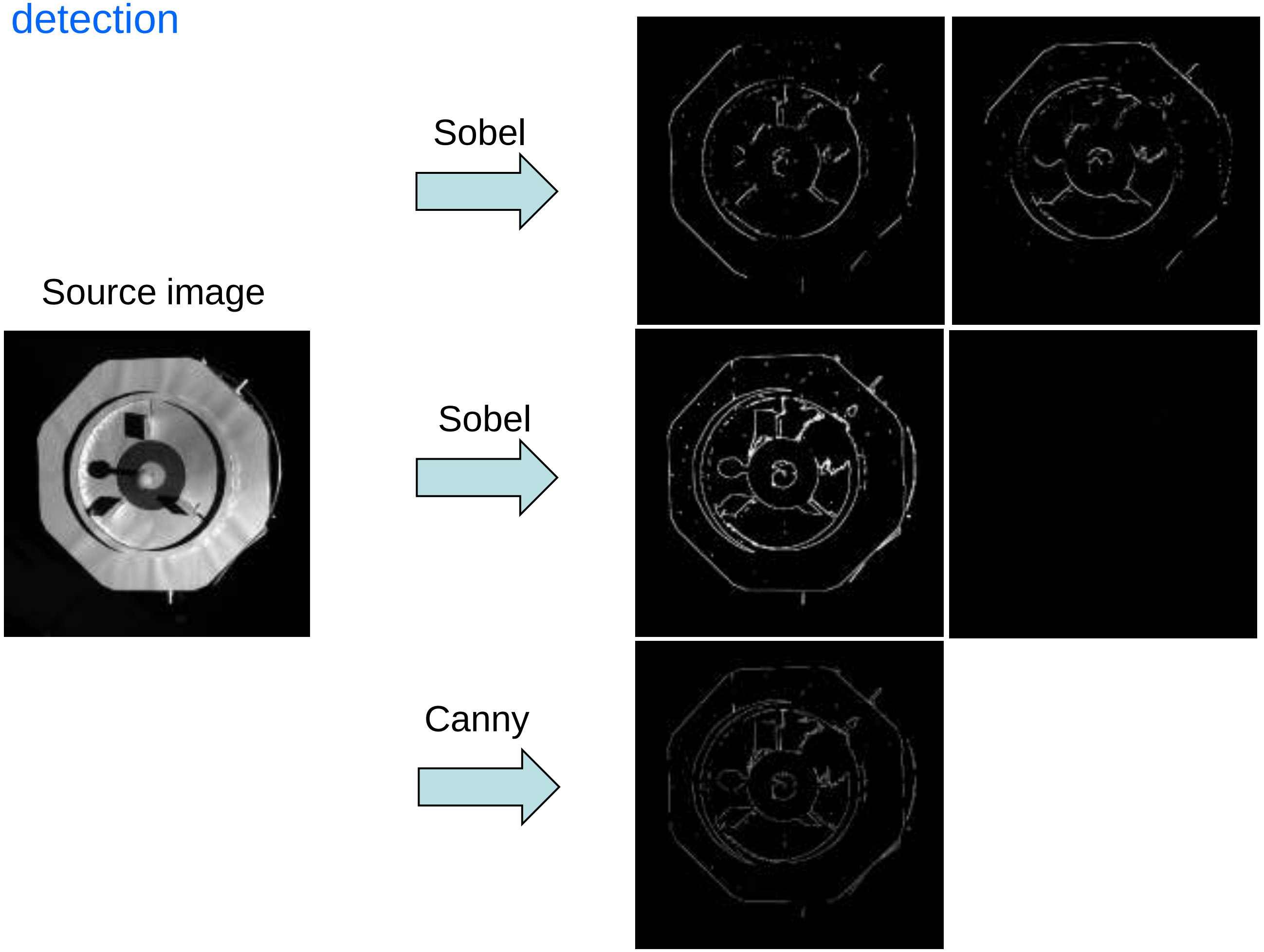
❐ Preprocessing steps

- Thresholds



Preprocessing steps

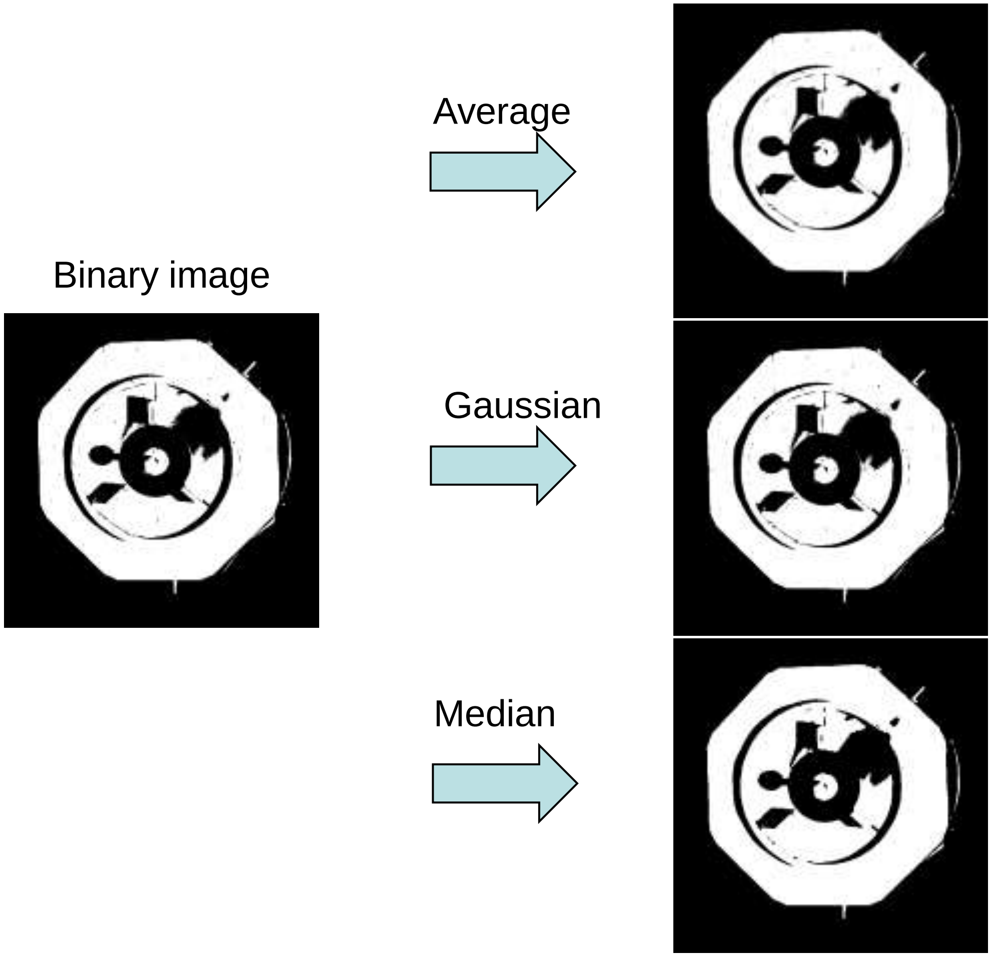
- Edge detection



Camera systems and data preprocessing

❒ Preprocessing steps

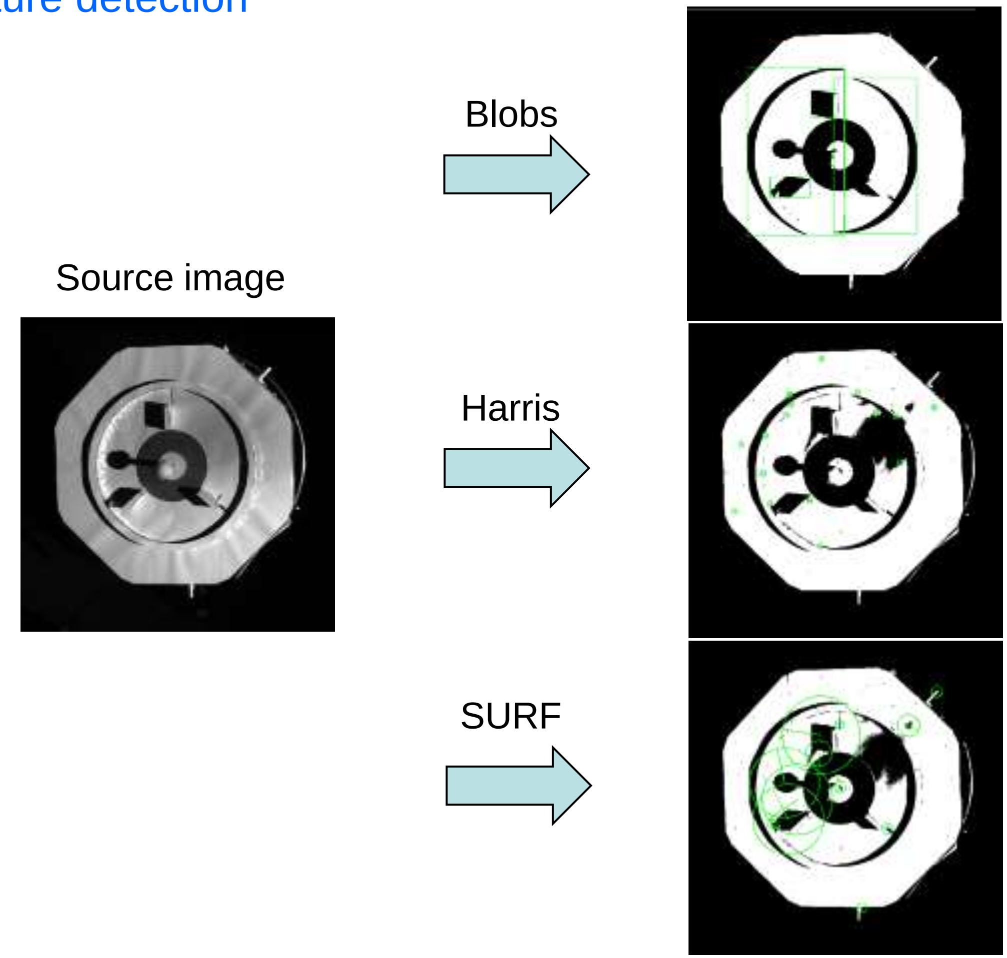
- ➢ Filter



Camera systems and data preprocessing

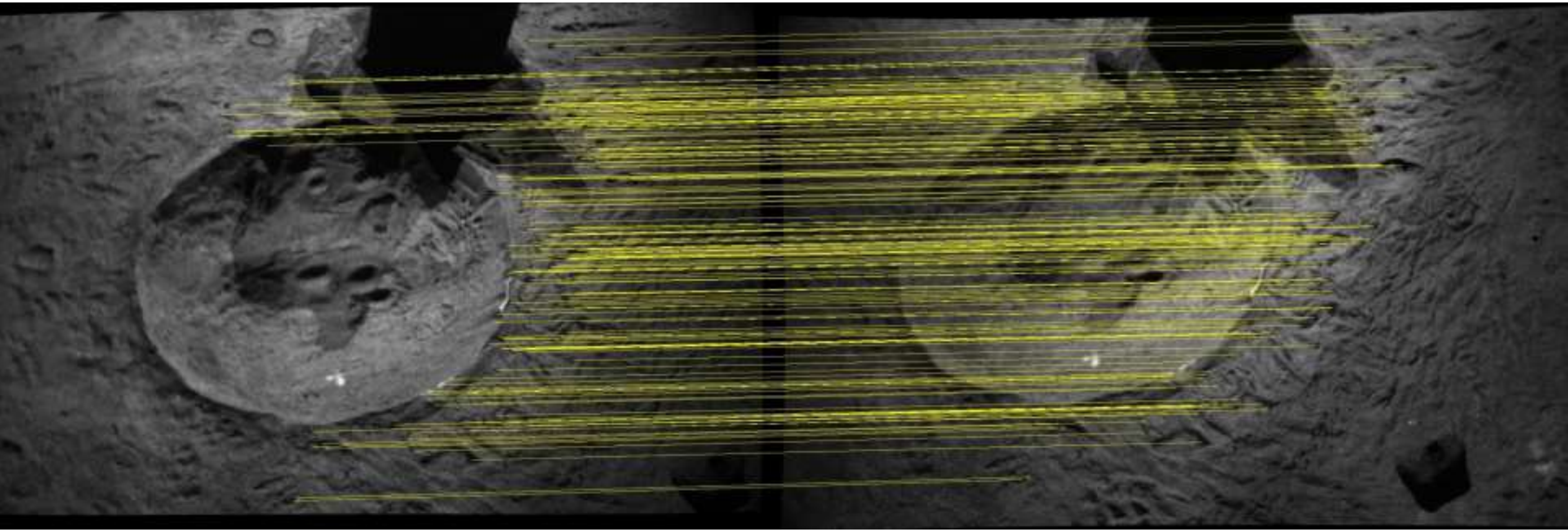
❐ Preprocessing steps

➢ Feature detection

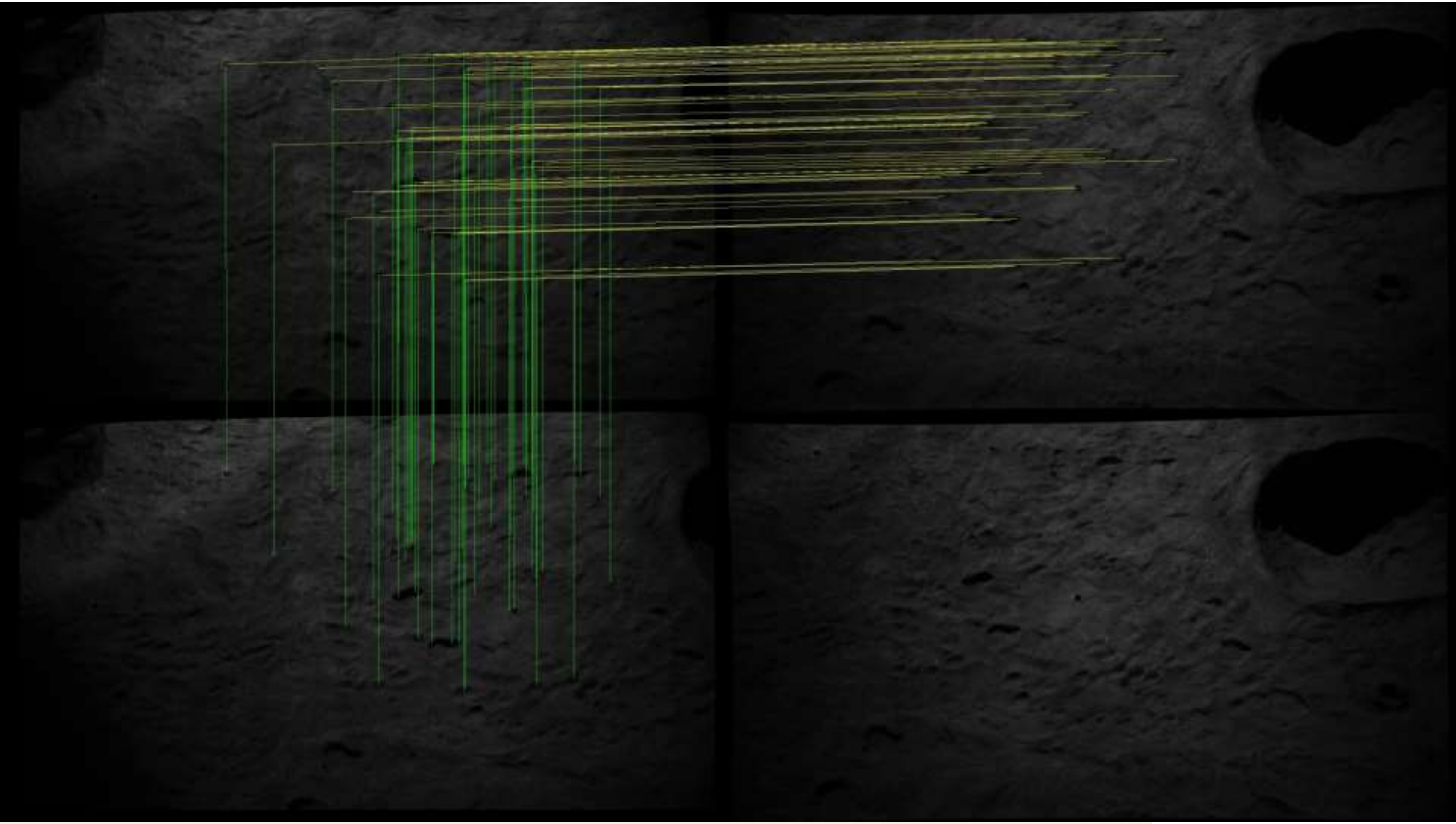


Camera systems and data preprocessing

- **Preprocessing steps**
  - 3D point clouds
    - Partial disparity
    - Triangulation
    - 200 ms computing time



Camera systems and data preprocessing



- **Thanks for your attention**
- **Questions?**

Thank you!

DFKI Bremen & Universität Bremen
Robotics Innovations Center
Director: Prof. Dr. Frank Kirchner
www.dfki.de/robotics
robotics@dfki.de

Universität Bremen