Intelligent Robot Control

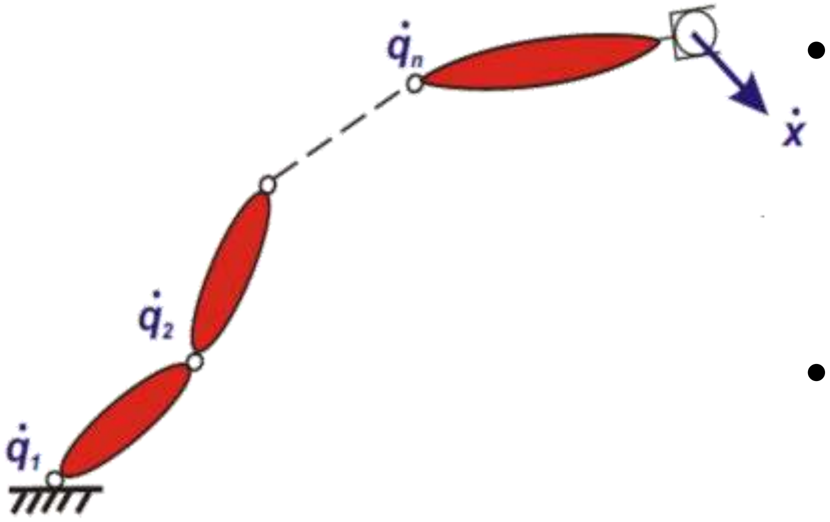
Lecture 3: Control of Redundant Robots

Tadej Petrič

tadej.petric@ijs.si

Institut
"Jožef Stefan"
Ljubljana, Slovenija

Robot manipulators



- Robot is seen as (open) kinematic chain of rigid bodies interconnected by (revolute or prismatic) joints.
- Parameterization:
  - Unambiguous and minimal characterization of the robot configuration
  - n = degrees of freedom (DOF)
  - n = robot joints (rotational or translational)
- Configuration on n-DOF robot is described by joint coordinates:

$$\boldsymbol{q} = [q_1, q_2, \ldots, q_n]^T \qquad q_i \in Q_i = [q_{i,min}, q_{i,max}]$$

- The configuration space *C* is the space where the joint variables *q* are defined:

$$\mathcal{C} : Q_1 \times Q_2 \times \cdots \times Q_n$$

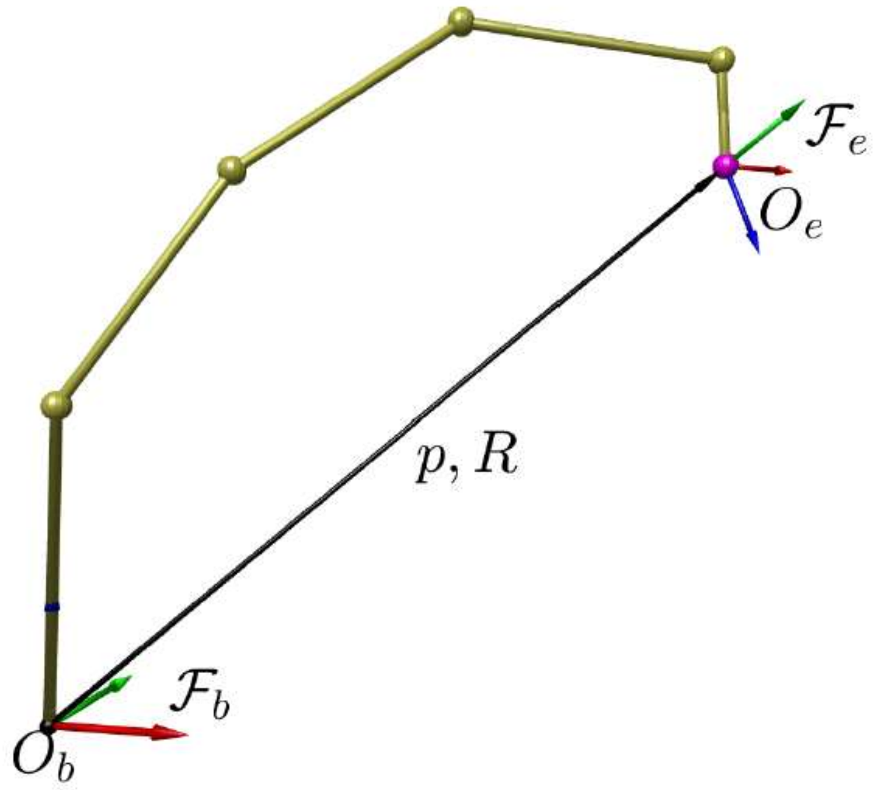
Institut "Jožef Stefan" Ljubljana, Slovenija

Robot pose

- End-effector position (operational point) in base coordinate frame:

$$\mathbf{T}_e = \begin{bmatrix} \mathrm{R}(q) & p(q) \\ \mathbf{0} & 1 \end{bmatrix}$$

- The operational space *O* is space, where the positions/orientations of the robot end-effector are defined (6-dimensional Cartesian space).
- Reachable workspace: is the set of all *p* where the robot can reach all positions with at least one orientation
- Dexterous workspace: is the set of all *p* where the robot can reach all positions with any feasible orientation

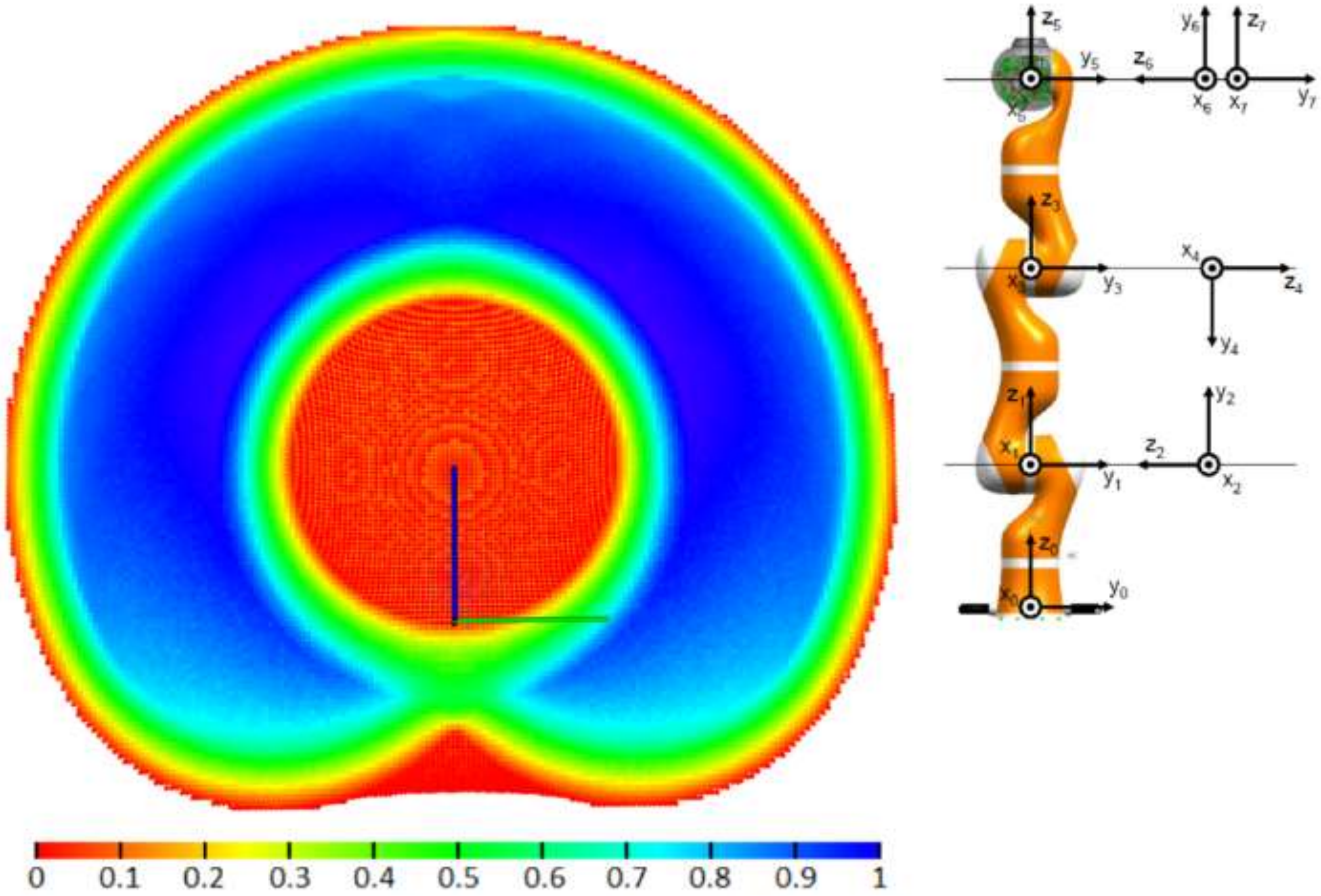


Institut
"Jožef Stefan"
Ljubljana, Slovenija

Working envelope of Kuka LWR

Cross-section of the workspace of a KUKA LBR arm in form of a Capability map.

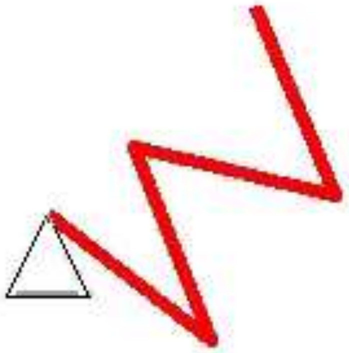
The HSV color scale encodes the dexterity of each region in the space; blue indicates areas with higher dexterity of the manipulator.



Source: Porges, Oliver, et al. "Reachability and dexterity: Analysis and applications for space robotics."

Redundant robots

- A robot is redundant if it has more degrees-of-freedom (DOF) than needed to accomplish a task.

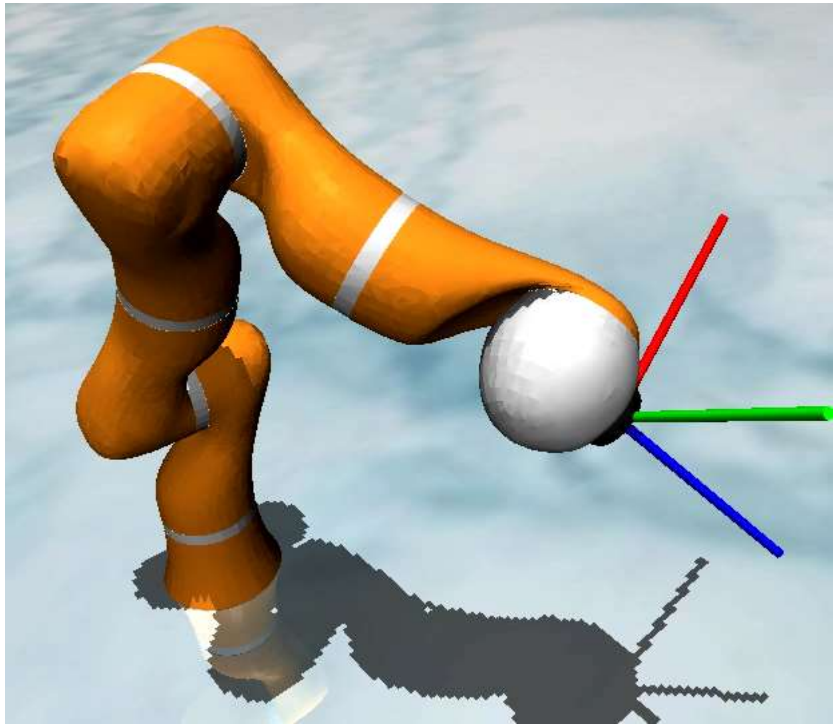


- Two types of redundancy can be identified:
  - Serial robots that have a joint-space dimension greater than their operational-space dimension are termed intrinsically redundant :

$$r_i = \dim(\mathcal{C}) - \dim(\mathcal{O})$$

  - A robot is termed functionally redundant if the task does not use all operational-space dimensions, $\mathcal{T} \subset \mathcal{O}$:

$$r_f = \dim(\mathcal{O}) - \dim(\mathcal{T})$$



Task space

- Task space $\mathcal{T} \subseteq \mathcal{O}$ is the space where the operation of robot is required.

- DOFs needed for some common tasks:
  - m = 2
    - pointing in space
    - positioning in plane
  - m = 3
    - orientation in space
    - positioning and orientation in plane
  - m = 5
    - positioning and pointing in space
  - m = 6
    - positioning and orientation in space

Kinematics of redundant robot

- Forward kinematics of redundant robots is given in the form

$$\dot{\boldsymbol{x}}_{(m\times 1)} = \mathbf{J}_{(m\times n)}\dot{\boldsymbol{q}}_{(n\times 1)} \qquad m < n$$

- Inverse kinematics problem is now to solve this set of equations. The system is under constraint and can be solved by choosing some additional constraints. The solution is of the form:

$$\dot{q} = \mathbf{J}^{\dagger}\dot{x}$$

- Where $\mathbf{J}^{\dagger}$ is some generalized inverse of **J**, e.g. any matrix satisfying

$$\mathbf{J}\mathbf{J}^{\dagger}\mathbf{J} = \mathbf{J}$$

- Note: Generalized inverse always exists.

Least-norm solution

- Consider: $y = \mathbf{A}x$
- Where $x \in \mathbb{R}^n,\ y \in \mathbb{R}^m,\ \mathbf{A} \in \mathbb{R}^{m \times n}$, and m < n (We have more variables than equations).
- Optimization problem:

$$\min \ \|y\| \qquad \text{subject to: } \ \mathbf{A}y = x$$

- Assume **A** has full row rank, $\mathcal{R}(\mathbf{A}) = m$. Then the solution has form:

$$\{x \mid \mathbf{A}x = y\} = \{x_p + z \mid z \in \mathcal{N}(\mathbf{A})\}$$

- One particular solution is

$$x_p = \mathbf{A}^T (\mathbf{A}\,\mathbf{A}^T)^{-1} y$$

Pseudo-inverse

- Most commonly used pseudo-inverses are Moore-Penrose pseudo-inverse (minimal joint velocities)

$$\mathbf{J}^{\dagger} = \mathbf{J}^{T}(\mathbf{J}\mathbf{J}^{T})^{-1}$$

- or weighted pseudo-inverse, where W is a weighting matrix:

$$\mathbf{J}^{\dagger} = \mathbf{W}^{-1}\mathbf{J}^{T}(\mathbf{J}\mathbf{W}^{-1}\mathbf{J}^{T})^{-1}$$

- Special case when **W = H** → dynamic consistent pseudoinverse.

- When the robot approaches a singular configuration this solution becomes inefficient, very high joint velocities are required even for small task space velocities in the directions which become unfeasible in the singularity.

Damped least-squares inverse

- From the mathematical point of view, the Jacobian **J** becomes near singular configuration ill-conditioned (some singular values of **J** become very small).
- Solution is the damped least-squares inverse (DLS)

$$\mathbf{J}^* = \mathbf{W}^{-1}\mathbf{J}^T(\mathbf{J}\mathbf{W}^{-1}\mathbf{J}^T + \lambda^2\mathbf{I})^{-1}$$

- where λ is the damping factor. The additional damping term $\lambda^2\mathbf{I}$ decreases the task space accuracy in favor of feasible joint velocities.
- $\mathbf{J}^*$ does not fulfill all Moore-Penrose conditions, e.g. $\mathbf{J}\mathbf{J}^*\mathbf{J} \neq \mathbf{J}$. Hence, the DLS inverse should not be used in the calculation of the null-space projectors.

General inverse kinematics solution

- The question is now, how to incorporate any constraints in the general solution given in the form

$$\dot{q} = \mathbf{J}^{\dagger}\dot{x} + (\mathbf{I}_n - \mathbf{J}^{\dagger}\mathbf{J})\dot{\varphi}$$

- where $\mathbf{I}_n$ is identity matrix and $\dot{\varphi}$ an arbitrary vector.
- The term $\mathbf{J}^{\dagger}\dot{x}$ represents the particular solution which satisfies the main task and any "rigid" constraints depending on the selected generalized inverse.
- To find a suitable generalize inverse $\mathbf{J}^{\dagger}$ we specify some performance criterion. By finding the optimum of this criterion we get the desired generalized inverse.

General inverse kinematics solution . . .

$$\dot{q} = \mathbf{J}^{\dagger}\dot{x} + (\mathbf{I}_n - \mathbf{J}^{\dagger}\mathbf{J})\dot{\varphi}$$

- The term $(\mathbf{I}_n - \mathbf{J}^{\dagger}\mathbf{J})\dot{\varphi}$ is the homogenous solution and serves to purely reconfigure the robot arm without affecting the task.
- The homogenous solution is typically used to achieve some additional goals, i.e. different joint velocities $\dot{q}$ can be obtained, which result in the same end-effector velocity $\dot{x}$. Typically, it is used for
  - obstacle avoidance
  - some kind of optimization
  - singularity avoidance
  - joint limits avoidance
  - pose control
  - . . .

Institut
"Jožef Stefan"
Ljubljana, Slovenija

Some performance measures

- Manipulability
  - A commonly used measure is the manipulability measure defined as

$$w = \sqrt{\sigma_1 \sigma_2 \cdots \sigma_m} = \sqrt{\det(\mathbf{J}\mathbf{J}^T)}$$

- Condition number
  - The condition number ρ is the ratio between the maximal and the minimal singular value of **J**.

$$\rho = \sqrt{\frac{\sigma_{\max}}{\sigma_{\min}}} \qquad \nabla\rho = \frac{1}{2\rho}\frac{\sigma_{\min}\nabla\sigma_{\max} - \sigma_{\max}\nabla\sigma_{\min}}{\sigma_{\min}^2}$$

- Gravity torques norm
  - Considering only the gravity, the performance measure ***p*** representing the weighted norm of joint torques can be expressed as

$$p = \boldsymbol{g}(\boldsymbol{q})^T \mathbf{W} \boldsymbol{g}(\boldsymbol{q}) \qquad \nabla p(\boldsymbol{q}) = 2\left(\frac{\partial \boldsymbol{g}(\boldsymbol{q})}{\partial \boldsymbol{q}}\right)^T \mathbf{W} \boldsymbol{g}(\boldsymbol{q})$$

Institut
"Jožef Stefan"
Ljubljana, Slovenija

Control at the kinematic level

- For velocity control the following kinematic controller can be used:

$$\dot{q}_c = \mathbf{J}^{+}\dot{\boldsymbol{x}}_c + \mathbf{N}\dot{\varphi} \qquad \mathbf{N} = (\mathbf{I} - \mathbf{J}^{+}\mathbf{J})$$

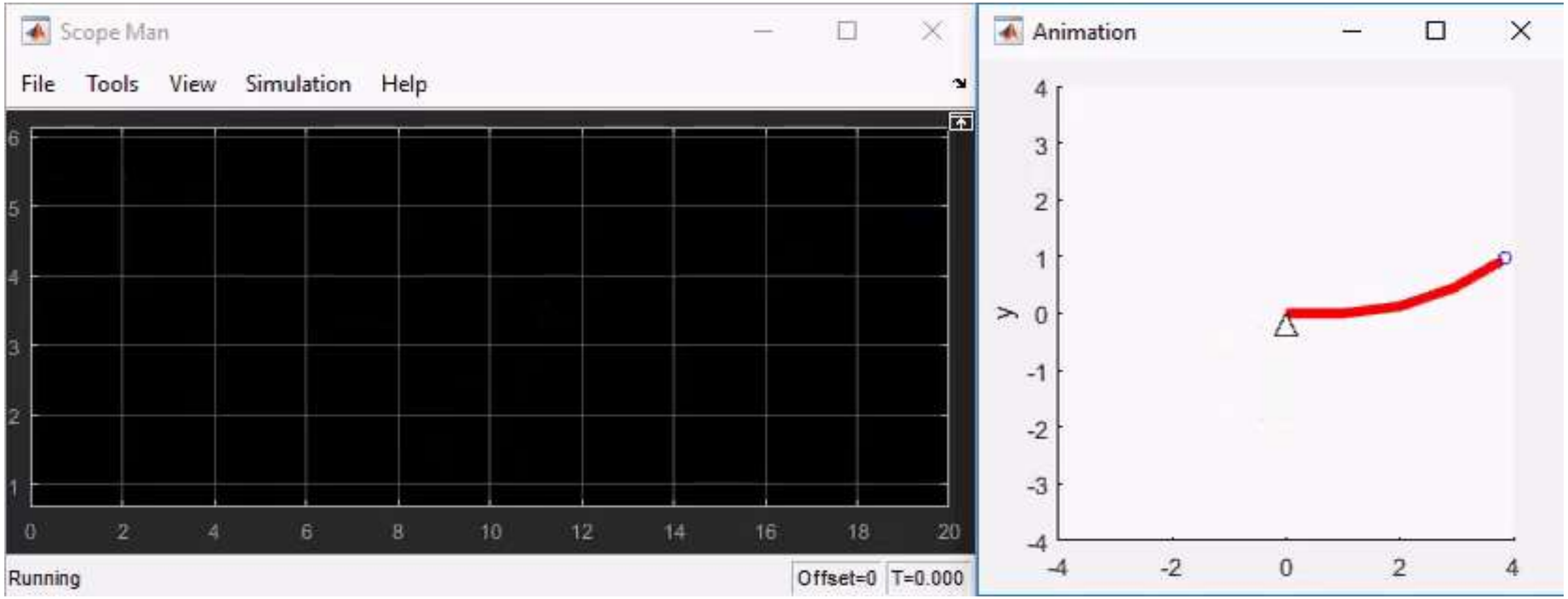
- Primary task: end-effector position $\dot{\boldsymbol{x}}_c$:

$$\dot{\boldsymbol{x}}_c = \dot{\boldsymbol{x}}_E + \mathbf{K}_p(\boldsymbol{x}_E - \boldsymbol{x})$$

- Secondary tasks: we use joint velocities $\dot{\varphi}$ (self motion).

Example: Kinematic control - Planar 4R

- Task space: PTP motion, kinematic control using Moore-Penrose pseudoinverse.
- Null-space: Optimization of robot pose by maximizing manipulability (avoiding singular configurations)



Institut
"Jožef Stefan"
Ljubljana, Slovenija

Control at the dynamic level

- Using the acceleration formulation we can use the following dynamic controller

$$\boldsymbol{\tau} = \mathbf{H}(\bar{\mathbf{J}}(\ddot{\boldsymbol{x}}_c - \dot{\mathbf{J}}\dot{\boldsymbol{q}}) + \bar{\mathbf{N}}(\boldsymbol{\phi} + \dot{\bar{\mathbf{J}}}\dot{\boldsymbol{x}}) + \mathbf{h} + \mathbf{g}$$

- Primary task: end-effector acceleration (position)

$$\ddot{\boldsymbol{x}}_c = \ddot{\boldsymbol{x}}_d + \mathbf{K}_v\dot{\boldsymbol{e}} + \mathbf{K}_p\boldsymbol{e}$$

- Secondary tasks: we use joint velocities $\dot{\boldsymbol{\varphi}}$ (self motion)

$$\boldsymbol{\phi} = \ddot{\boldsymbol{\varphi}} + \mathbf{K}_n\bar{\mathbf{N}}(\dot{\boldsymbol{\varphi}} - \dot{\boldsymbol{q}})$$

- By selecting proper controller parameters the following dynamic properties can be achieved

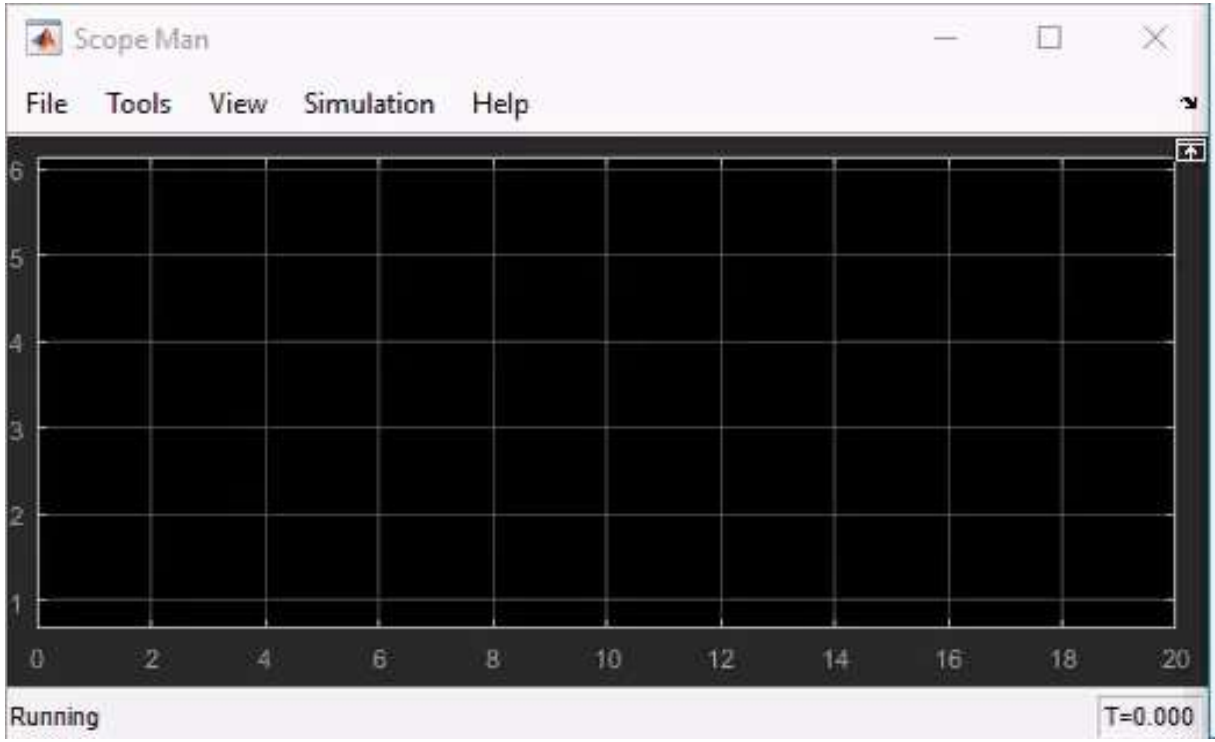
$$\boldsymbol{\Lambda}\ddot{\boldsymbol{e}} + \boldsymbol{\Lambda}\mathbf{K}_v\dot{\boldsymbol{e}} + \boldsymbol{\Lambda}\mathbf{K}_p\boldsymbol{e} = -\boldsymbol{F}$$

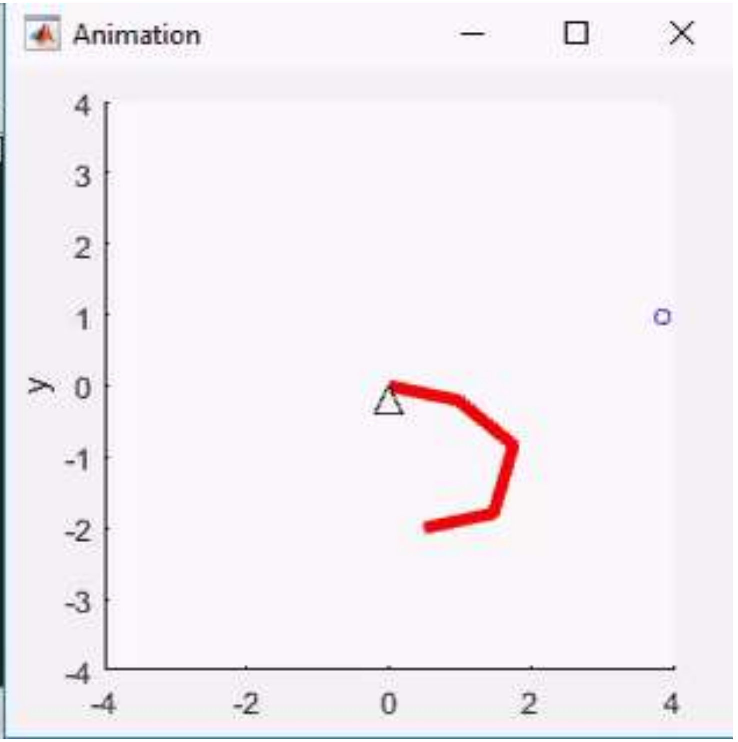
$$\mathbf{H}_n\ddot{\boldsymbol{e}}_n + \mathbf{H}_n\mathbf{K}_n\dot{\boldsymbol{e}}_n = -\bar{\mathbf{N}}^T\boldsymbol{\tau}_F$$

- Effective inertia matrix in $\mathbf{N}$: $\mathbf{H}_n = \bar{\mathbf{N}}^T\mathbf{H}\bar{\mathbf{N}} = \mathbf{H} - \mathbf{J}^T\boldsymbol{\Lambda}\mathbf{J}$

Example: Dynamic control - Planar 4R

- Task space: PTP motion, dynamic control using inertia-weighted pseudoinverse.
- Null-space: Optimization of robot pose by maximizing manipulability (avoiding singular configurations)



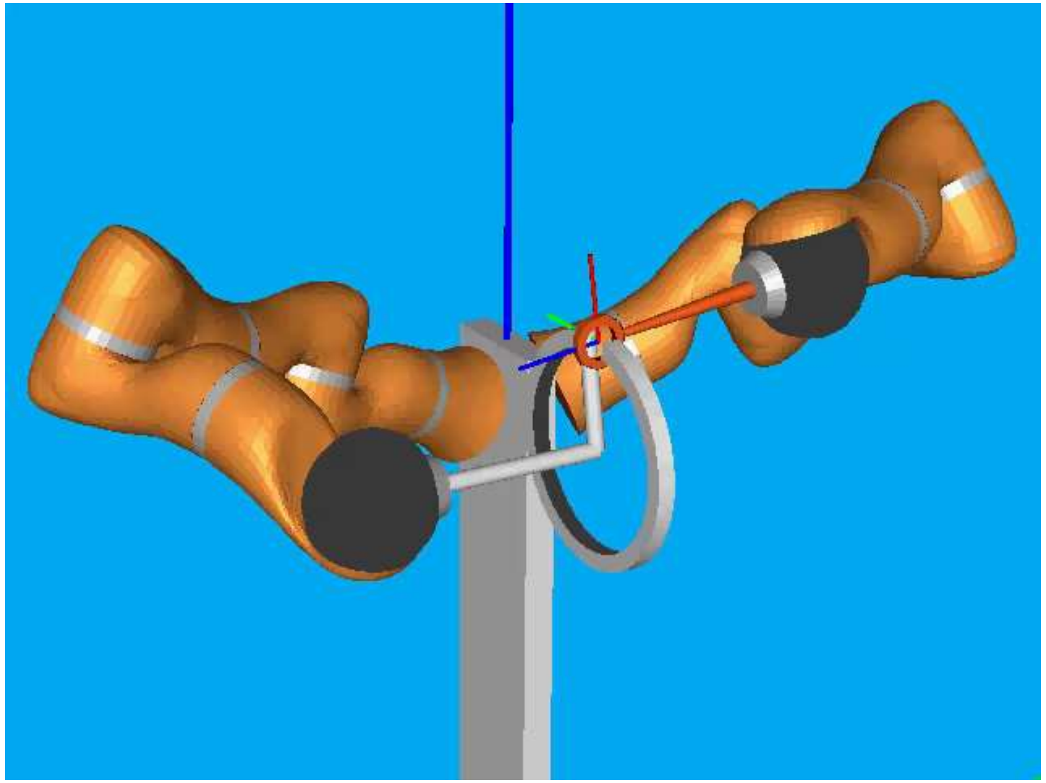


Institut
"Jožef Stefan"
Ljubljana, Slovenija

Control of functional redundant robots

- Some task do not require controlled motion in all spatial directions.
- Example: The motion of the ring is free around the hoop, so we can remove rotation around the x-axis from the task control.

$$\dot{x} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$



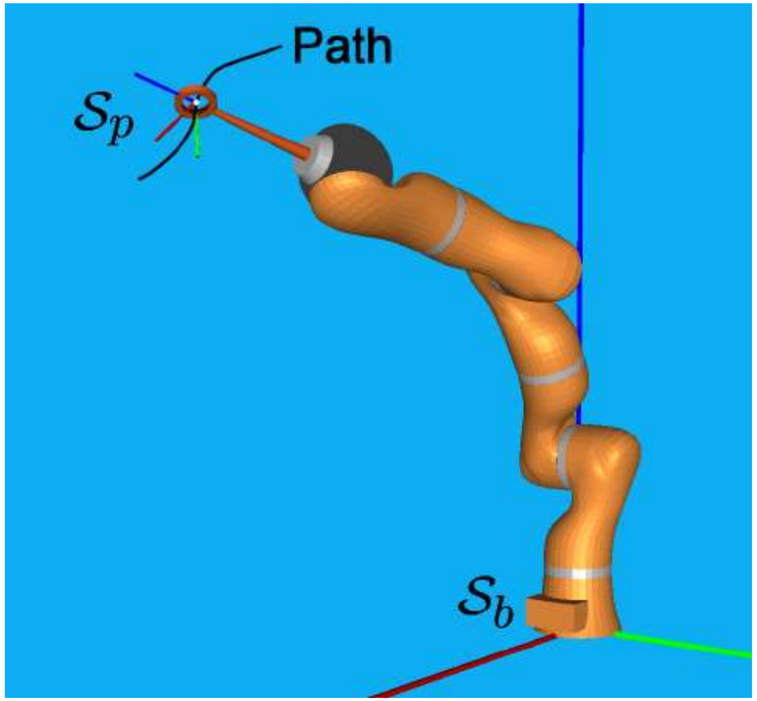
- The problem is when the ring is moved along the hoop and the control is not adequate anymore.

Institut
"Jožef Stefan"
Ljubljana, Slovenija

Control of functional redundant robots . . .

- To exploit the available functional redundancy it is necessary to and a task frame where the redundant DOFs are rows of the Jacobian matrix.
- If the task frame is changing along the task path, we have to consider this in the control.
- Mapping between the path frame $\mathcal{S}_p$ and base frame $\mathcal{S}_b$ (only rotation)

$$\widetilde{\mathbf{R}}_t = \begin{bmatrix} \mathbf{R}_t & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{R}_t \end{bmatrix}$$



- We map the control into the workspace which is anchored on the path

$$\dot{q}_c = (\widetilde{\mathbf{R}}_t^T \mathbf{J})^{\#} \left(\widetilde{\mathbf{R}}_t^T \begin{bmatrix} \mathbf{K}_p e_p + \dot{\boldsymbol{p}}_d \\ \mathbf{K}_o e_o + \boldsymbol{\omega}_d \end{bmatrix} \right) + (\mathbf{I} - (\widetilde{\mathbf{R}}_t^T \mathbf{J})^{\#} \widetilde{\mathbf{R}}_t^T \mathbf{J}) \dot{q}_n \, ,$$

Institut
"Jožef Stefan"
Ljubljana, Slovenija

Control of functional redundant robots . . .

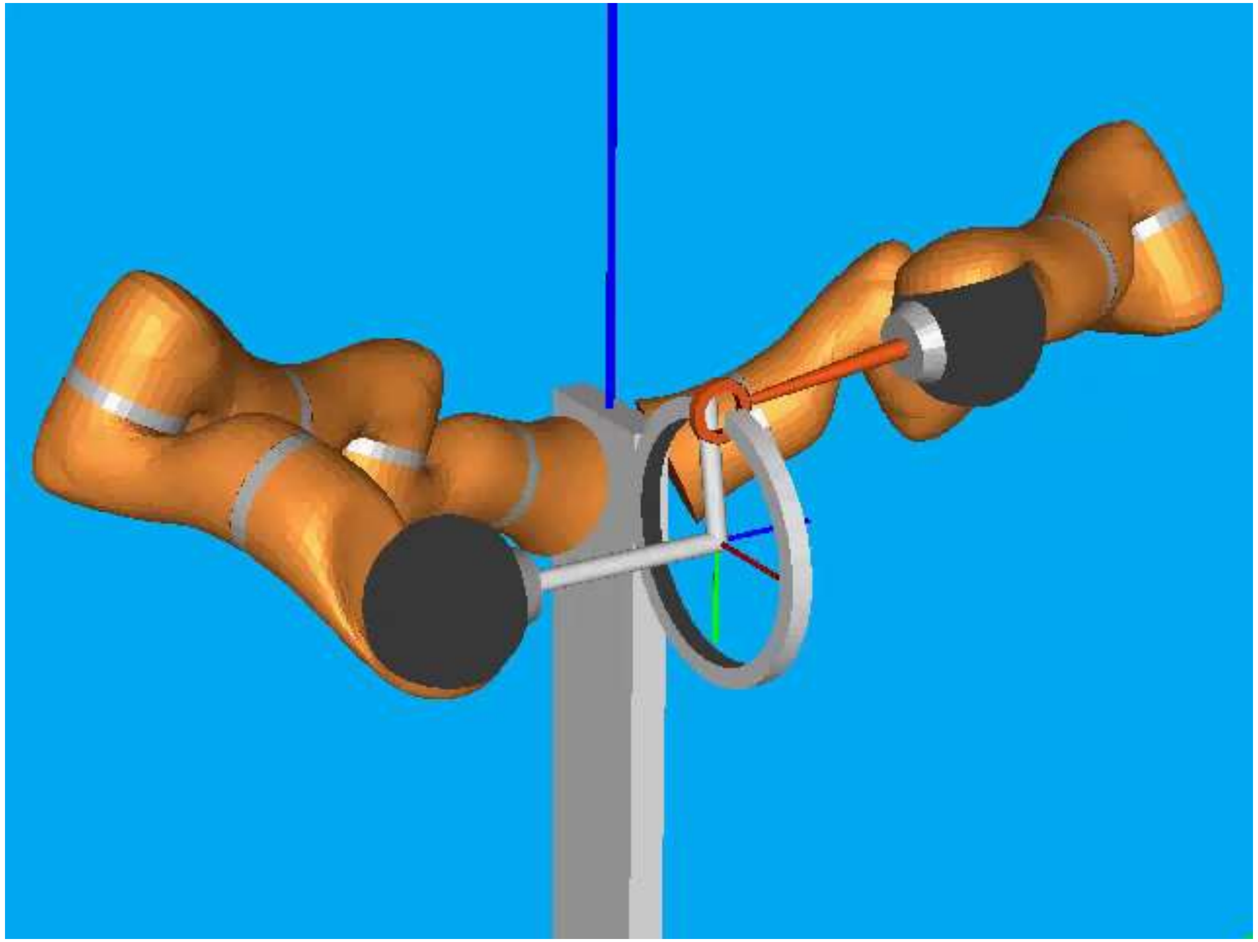
$$\dot{\boldsymbol{q}}_c = \mathbf{J}^{\dagger} \begin{bmatrix} \mathbf{K}_p \boldsymbol{e}_p + \dot{\boldsymbol{p}}_d \\ \mathbf{K}_o \boldsymbol{e}_o + \boldsymbol{\omega}_d \end{bmatrix}$$

- Let assume that for the tasks the linear motion in direction of y-axis and orientation around z-axis is not important.

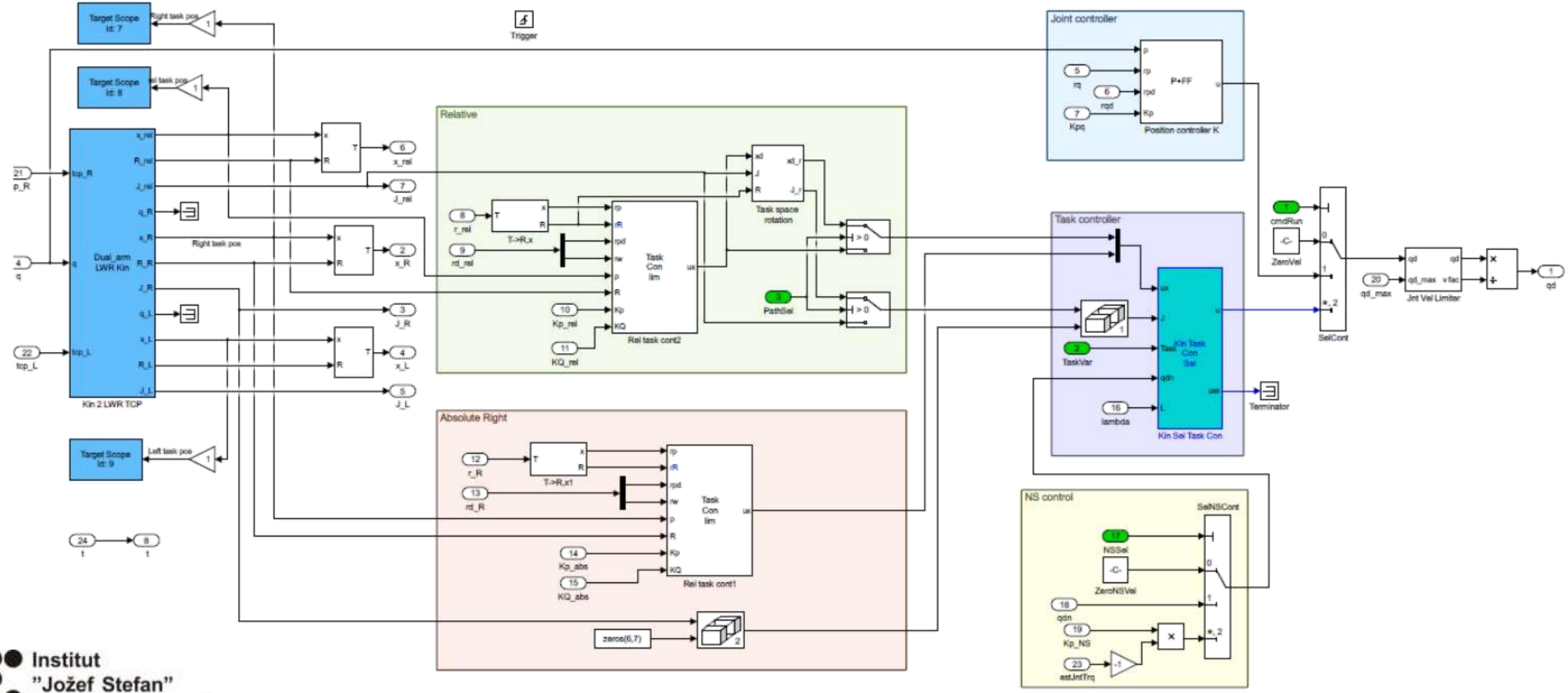
$$\dot{\boldsymbol{q}}_c = \begin{bmatrix} J_{11} & \cdots & J_{1n} \\ J_{21} & \cdots & J_{2n} \\ J_{31} & \cdots & J_{3n} \\ J_{41} & \cdots & J_{4n} \\ J_{51} & \cdots & J_{5n} \\ J_{61} & \cdots & J_{6n} \end{bmatrix}^{\#} \begin{bmatrix} \mathbf{K}_p \begin{bmatrix} e_{p,x} \\ e_{p,y} \\ e_{p,x} \end{bmatrix} + \begin{bmatrix} \dot{p}_{d,x} \\ \dot{p}_{d,y} \\ \dot{p}_{d,z} \end{bmatrix} \\ \\ \mathbf{K}_o \begin{bmatrix} e_{o,x} \\ e_{o,y} \\ e_{o,z} \end{bmatrix} + \begin{bmatrix} \dot{\omega}_{d,x} \\ \dot{\omega}_{d,y} \\ \dot{\omega}_{d,z} \end{bmatrix} \end{bmatrix}$$

Control in the path space

- Free DOF is the rotation around path (rotation axis is in the direction of y-axis of path space - y-axis of the $\mathcal{S}_p$ connected to the path).

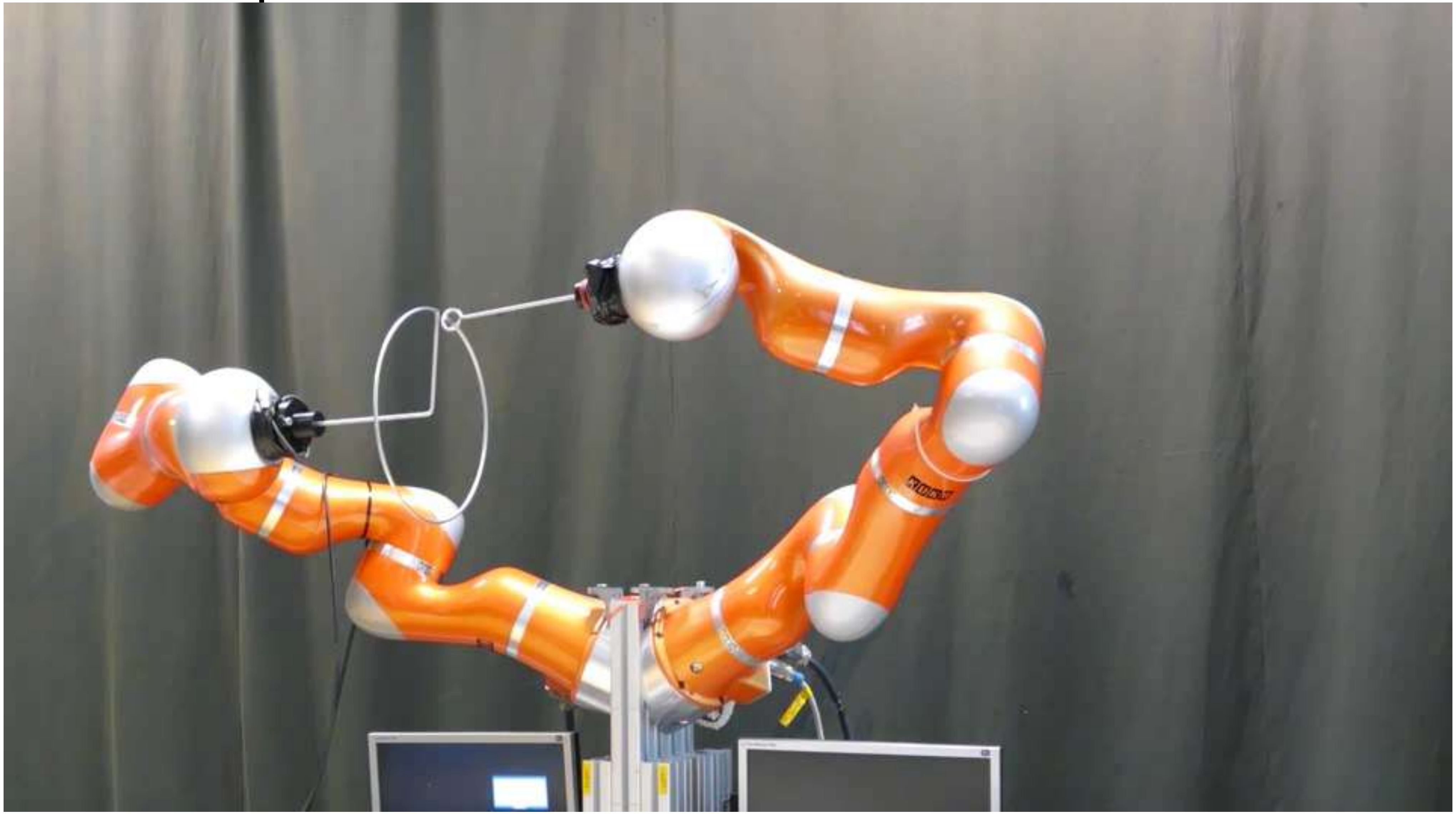


Kuka dual-arm controller



Institut
"Jožef Stefan"
Ljubljana, Slovenija

LWR hoop - Obstacle avoidance



Institut
"Jožef Stefan"
Ljubljana, Slovenija