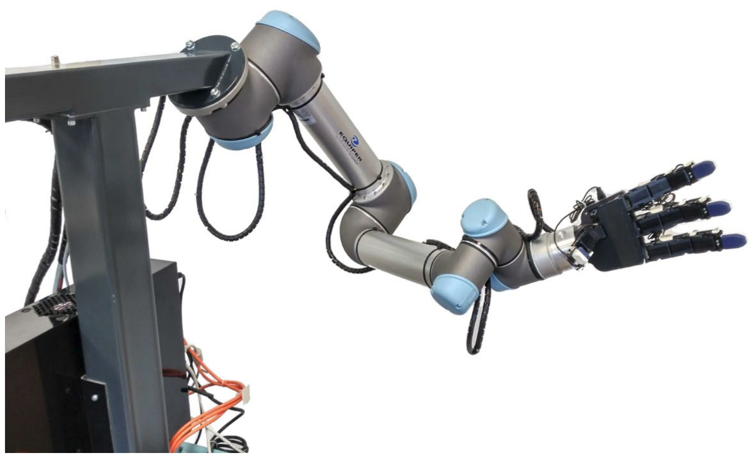
9th Conference on Applied Geometric Algebras in Computer Science and Engineering (AGACSE)

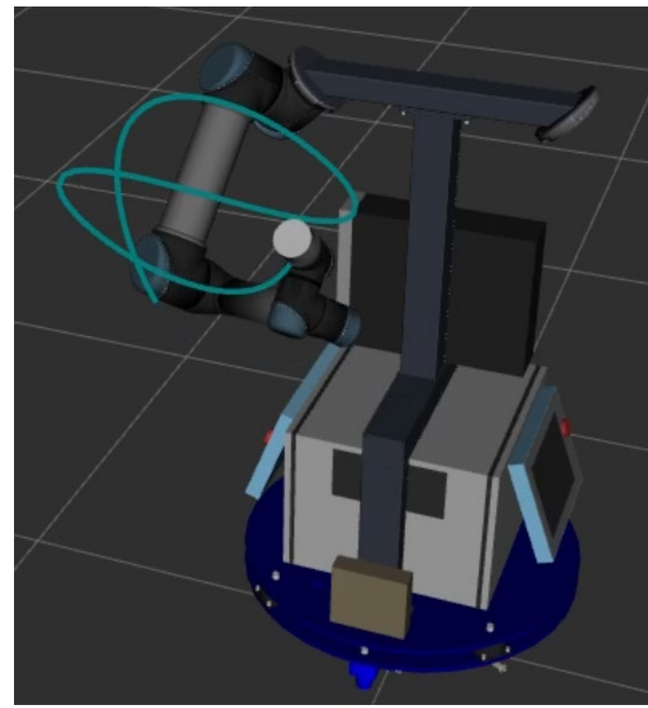
Closed-form inverse kinematics solutions for a class of serial robots without spherical wrist using conformal geometric algebra

Arnau Marzabal and Isiah Zaplana

Institute of Industrial and Control Engineering
Universitat Politècnica de Catalunya

isiah.zaplana@upc.edu





UPC
UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Institut d'Organització i Control
de Sistemes Industrials

What to expect from this talk? An outline

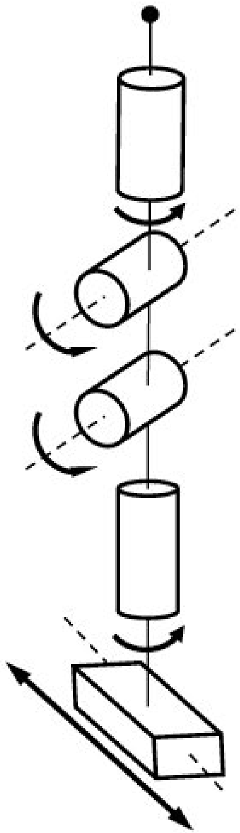
- The **formulation of the inverse kinematics (IK) problem** for serial robots and the traditional approaches to solve it (inc. some with GA).

- A novel **CGA-based method to solve the IK problem** for some classes of serial robots without spherical wrist.

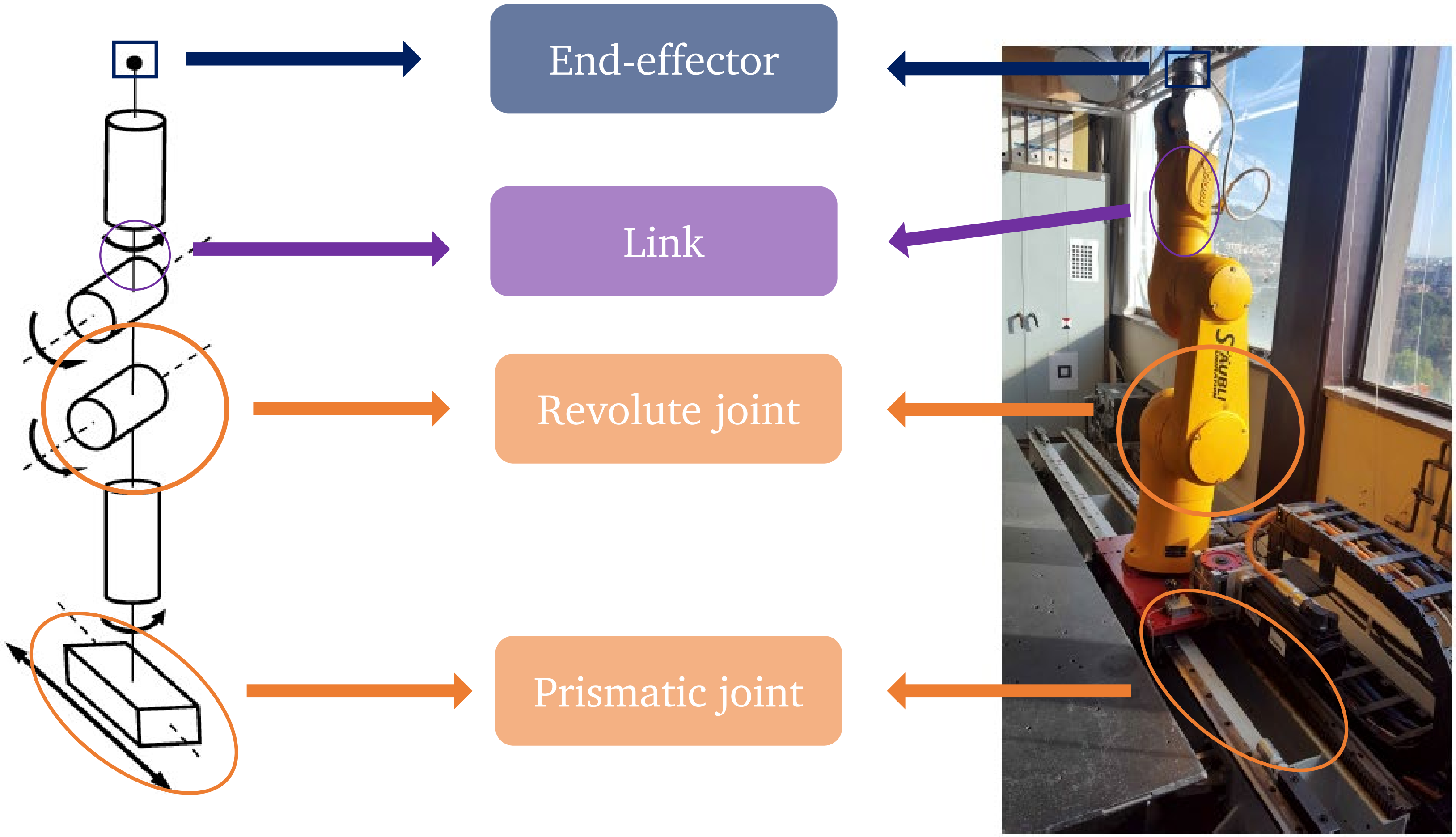
- **Implementation of the solution** using specific GA libraries (**inc. a new library**).

- **Validation** of the proposed solution **with a real robot**.

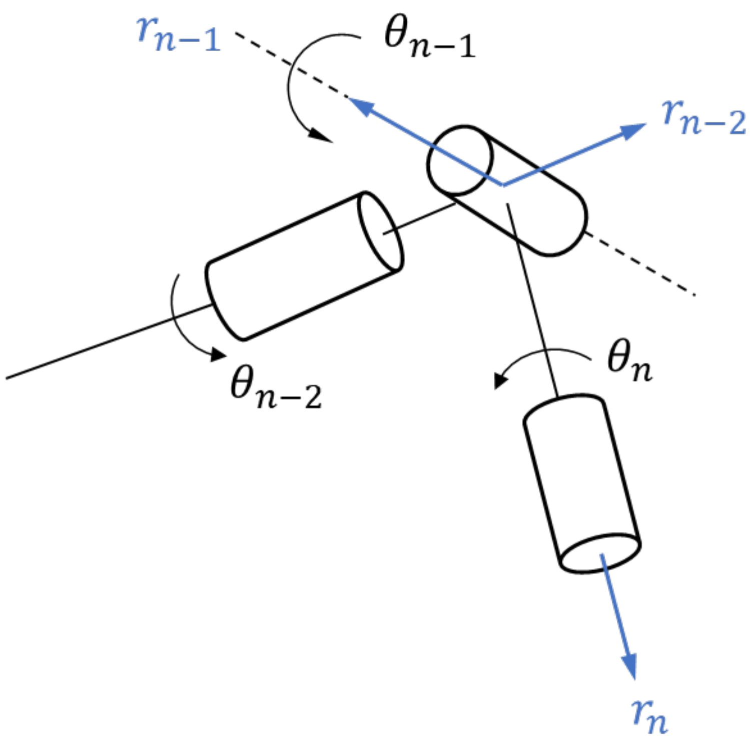
The actors → Serial robots



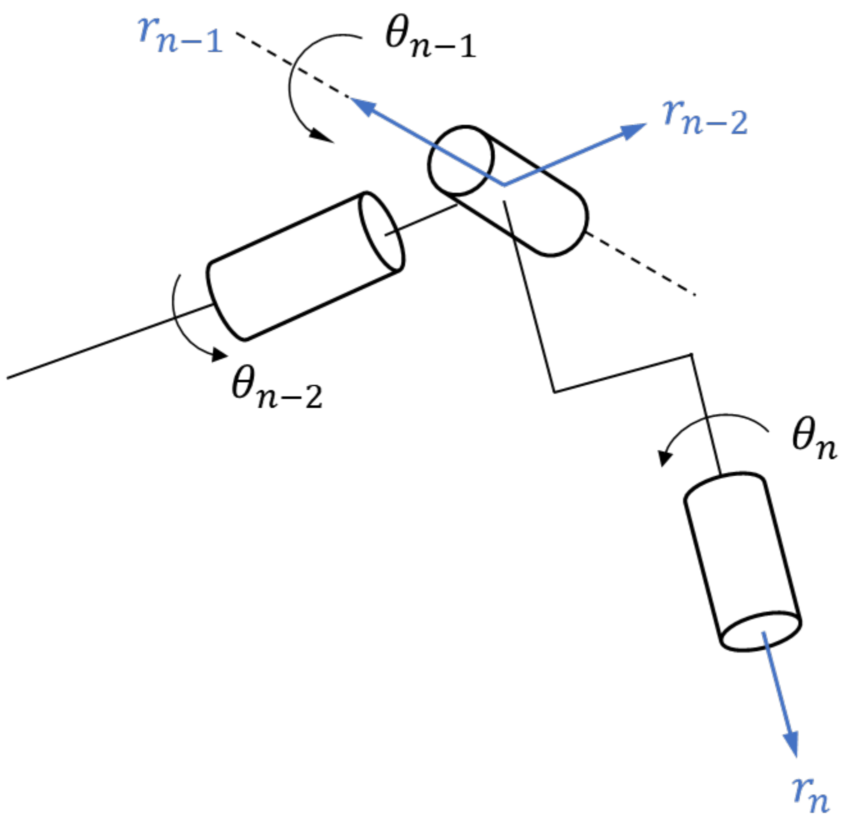
The actors → Serial robots



The actors → Serial robots



Spherical wrist



Non-spherical wrist

Serial robots – Some terms

- **Joint variable →** Rotation angle θ (revolute joints) or amount of displacement d (prismatic joints).
- **Configuration →** Vector $\boldsymbol{q} = (q_1, \ldots, q_n)$ with $q_i = \theta_i$ (i revolute joint) or $q_i = d_i$ (i prismatic joint).
- **Configuration space** $\mathcal{C}$ **→** (Vector, Topological) Space of all configurations.
- **Workspace** $\mathcal{W}$ **→** Space of all the positions and orientations of the end-effector.
- **Kinematic map →** Relation between the configuration space $\mathcal{C}$ and the workspace $\mathcal{W}$ of a given robot. The direct relation, $\mathcal{C} \to \mathcal{W}$ is the **forward kinematics**, while the inverse relation, $\mathcal{W} \to \mathcal{C}$ is the **inverse kinematics**.

Serial robots – Some terms

- **Joint variable** → Rotation angle θ (revolute joints) or amount of displacement d (prismatic joints).
- **Configuration** → Vector $\boldsymbol{q} = (q_1, \ldots, q_n)$ with $q_i = \theta_i$ (i revolute joint) or $q_i = d_i$ (i prismatic joint).
- **Configuration space** $\mathcal{C}$ → (Vector, Topological) Space of all configurations.
- **Workspace** $\mathcal{W}$ → Space of all the positions and orientations of the end-effector.
- **Kinematic map** → Relation between the configuration space $\mathcal{C}$ and the workspace $\mathcal{W}$ of a given robot. The direct relation, $\mathcal{C} \to \mathcal{W}$ is the **forward kinematics**, while the inverse relation, $\mathcal{W} \to \mathcal{C}$ is the **inverse kinematics**.

Serial robots – Some terms

- **Joint variable** → Rotation angle θ (revolute joints) or amount of displacement d (prismatic joints).
- **Configuration** → Vector $\boldsymbol{q} = (q_1, \ldots, q_n)$ with $q_i = \theta_i$ (i revolute joint) or $q_i = d_i$ (i prismatic joint).
- **Configuration space** $\mathcal{C}$ → Space of all configurations.
- **Workspace** $\mathcal{W}$ → Space of all the positions and orientations of the end-effector.
- **Kinematic map** → Relation between the configuration space $\mathcal{C}$ and the workspace $\mathcal{W}$ of a given robot. The direct relation, $\mathcal{C} \to \mathcal{W}$ is the **forward kinematics**, while the inverse relation, $\mathcal{W} \to \mathcal{C}$ is the **inverse kinematics**.

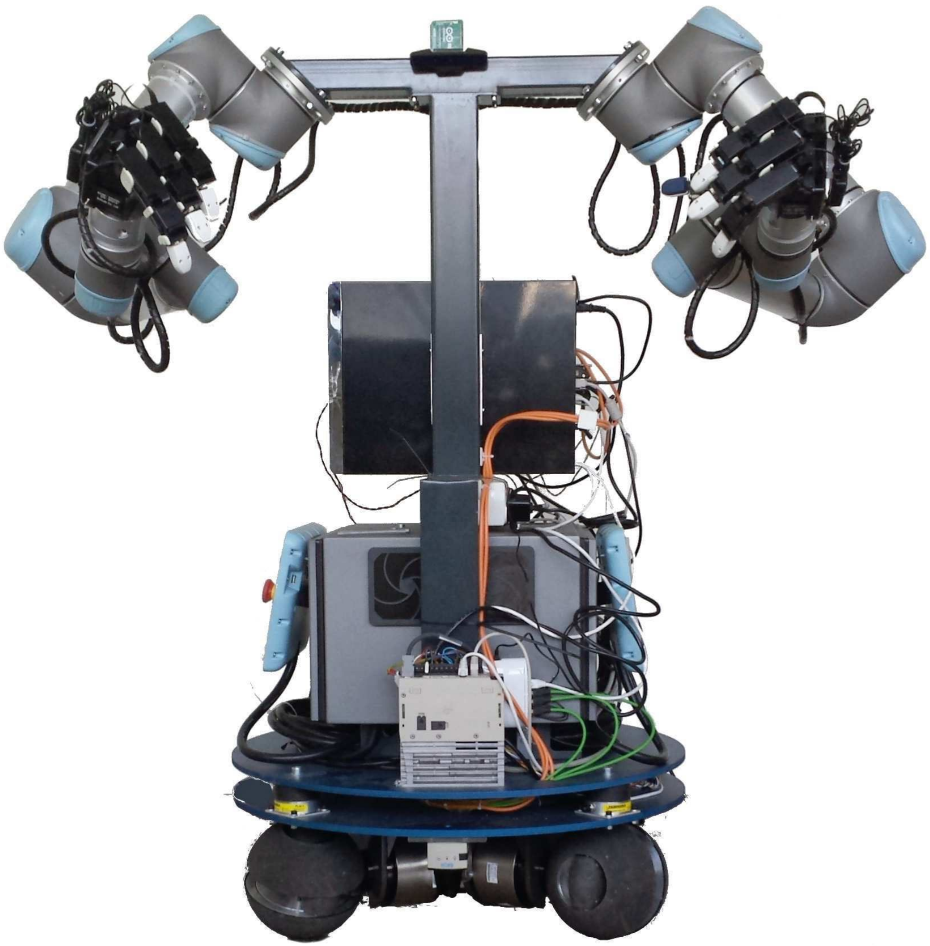
Serial robots – Some terms

- **Joint variable** → Rotation angle θ (revolute joints) or amount of displacement d (prismatic joints).
- **Configuration** → Vector $\boldsymbol{q} = (q_1, \dots, q_n)$ with $q_i = \theta_i$ (i revolute joint) or $q_i = d_i$ (i prismatic joint).
- **Configuration space** $\mathcal{C}$ → Space of all configurations.
- **Workspace** $\mathcal{W}$ → Space of all the positions and orientations of the end-effector.
- **Kinematic map** → Relation between the configuration space $\mathcal{C}$ and the workspace $\mathcal{W}$ of a given robot. The direct relation, $\mathcal{C} \rightarrow \mathcal{W}$ is the **forward kinematics**, while the inverse relation, $\mathcal{W} \rightarrow \mathcal{C}$ is the **inverse kinematics**.

Serial robots – Some terms

- **Joint variable** → Rotation angle θ (revolute joints) or amount of displacement d (prismatic joints).
- **Configuration** → Vector $\boldsymbol{q} = (q_1, \dots, q_n)$ with $q_i = \theta_i$ (i revolute joint) or $q_i = d_i$ (i prismatic joint).
- **Configuration space** $\mathcal{C}$ → Space of all configurations.
- **Workspace** $\mathcal{W}$ → Space of all the positions and orientations of the end-effector.
- **Kinematic map** → Relation between the configuration space $\mathcal{C}$ and the workspace $\mathcal{W}$ of a given robot. The direct relation, $\mathcal{C} \rightarrow \mathcal{W}$ is the **forward kinematics**, while the inverse relation, $\mathcal{W} \rightarrow \mathcal{C}$ is the **inverse kinematics**.

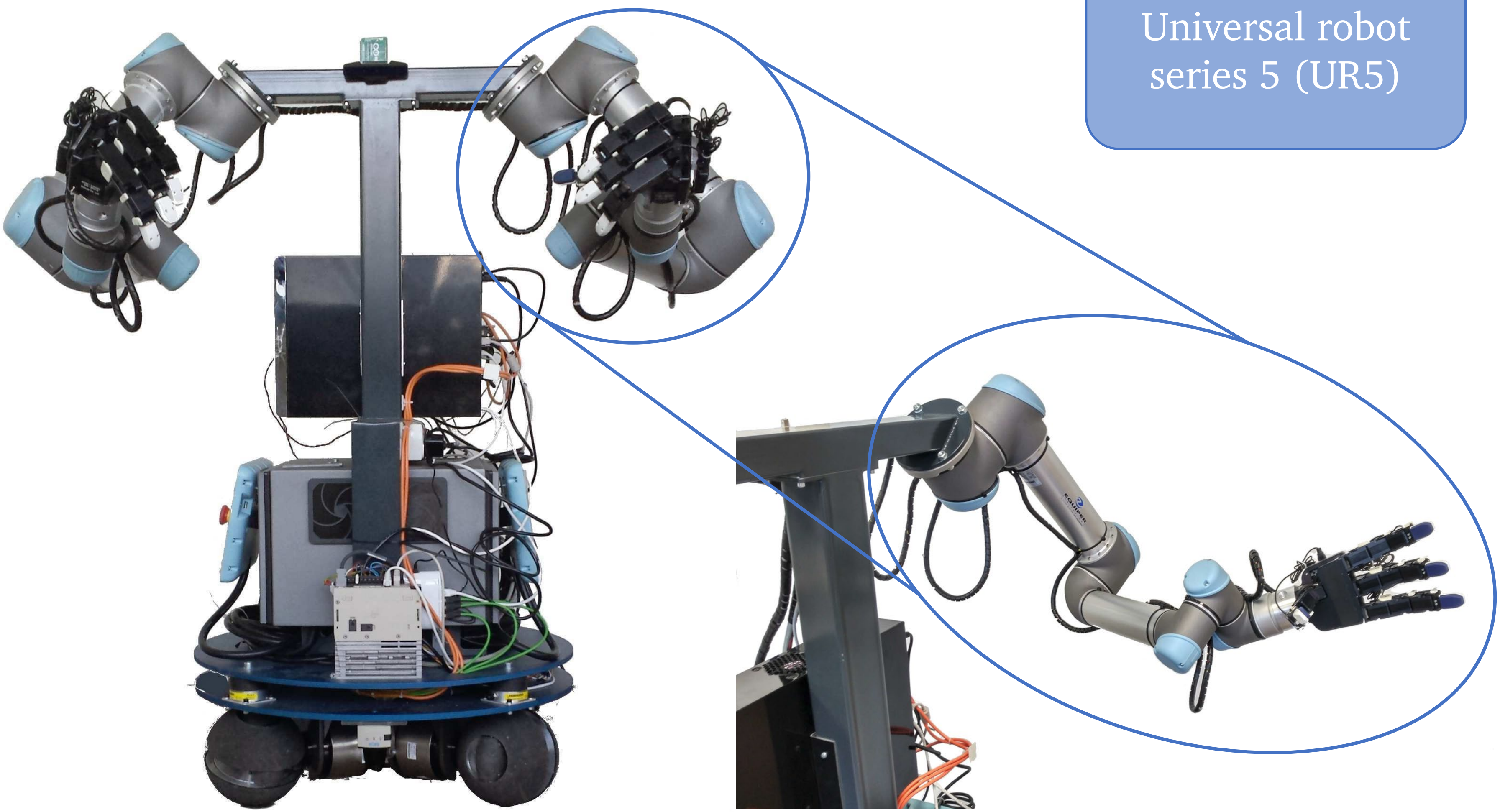
The main character



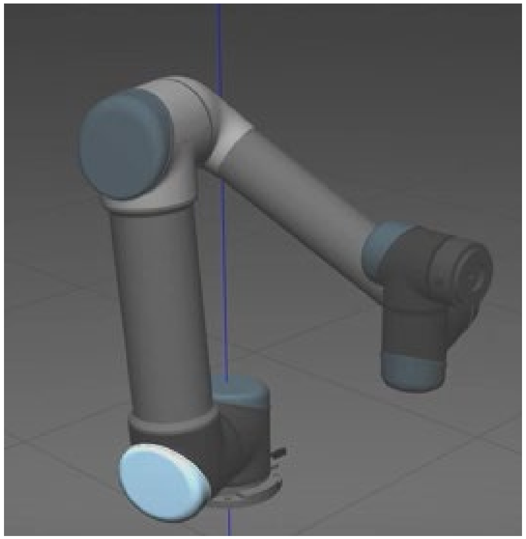
Mobile Anthropomorphic Dual-Arm Robot (MADAR)

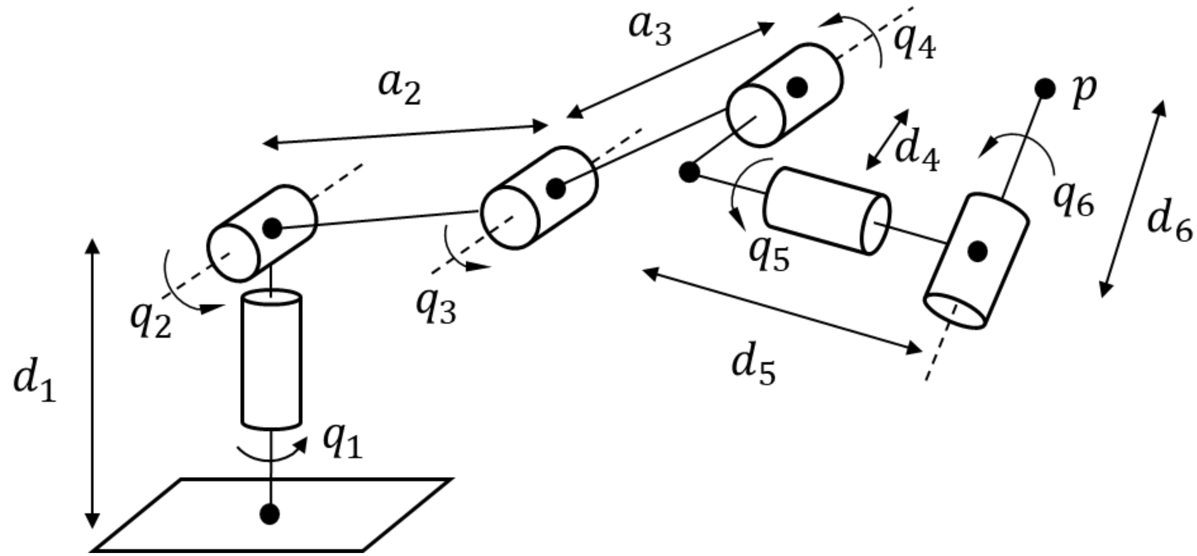
Serial robots – Use case

Universal robot series 5 (UR5)

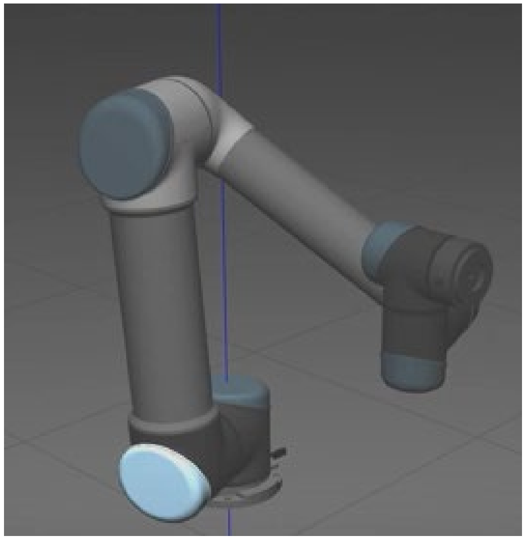


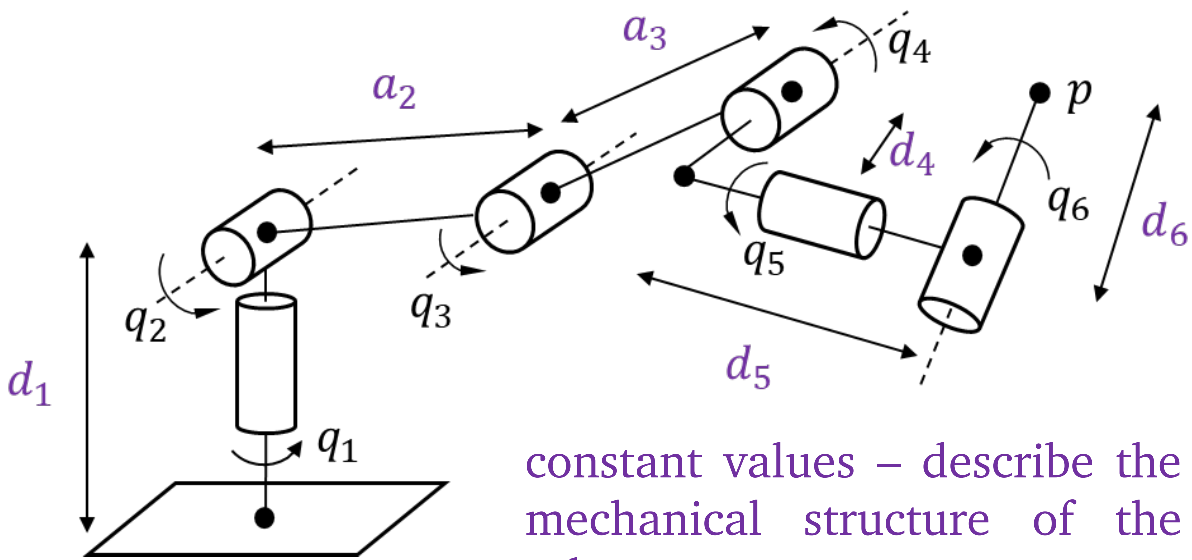
Serial robots – Use case



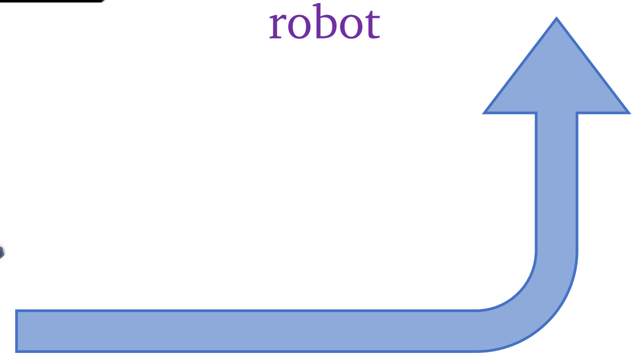


Serial robots – Use case

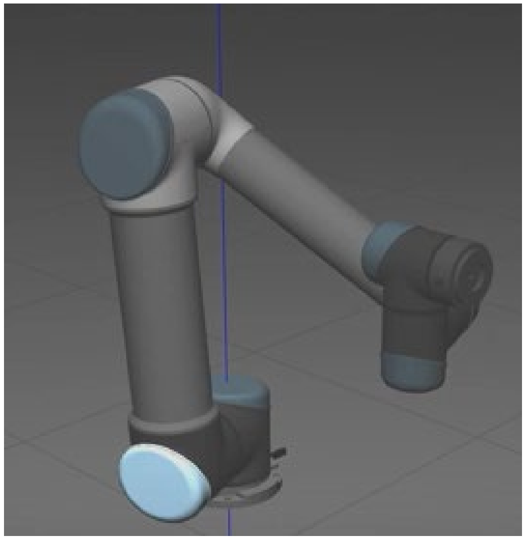


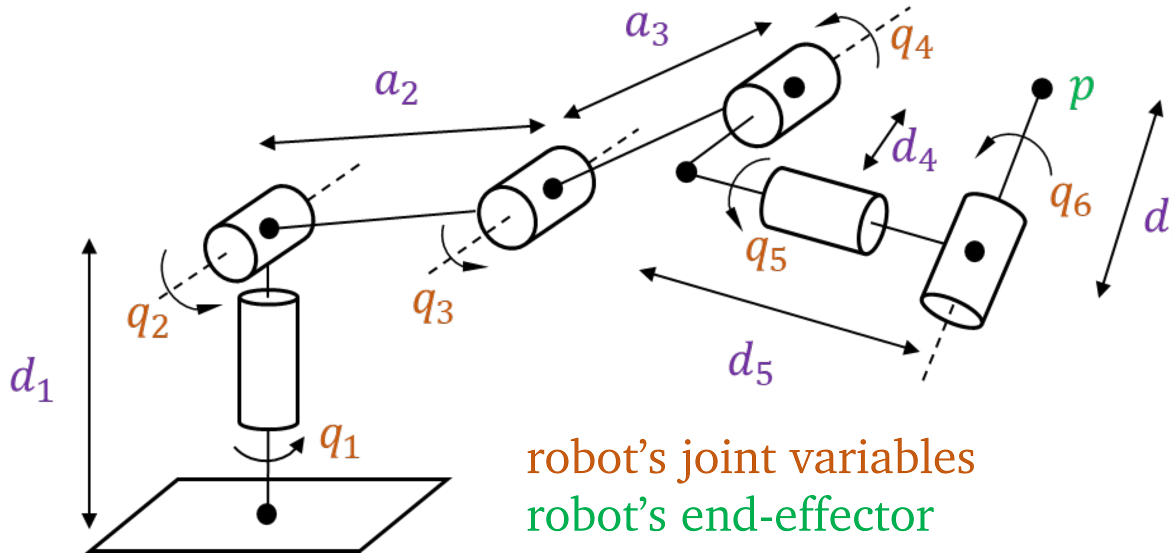


constant values – describe the mechanical structure of the robot



Serial robots – Use case





robot's joint variables

robot's end-effector

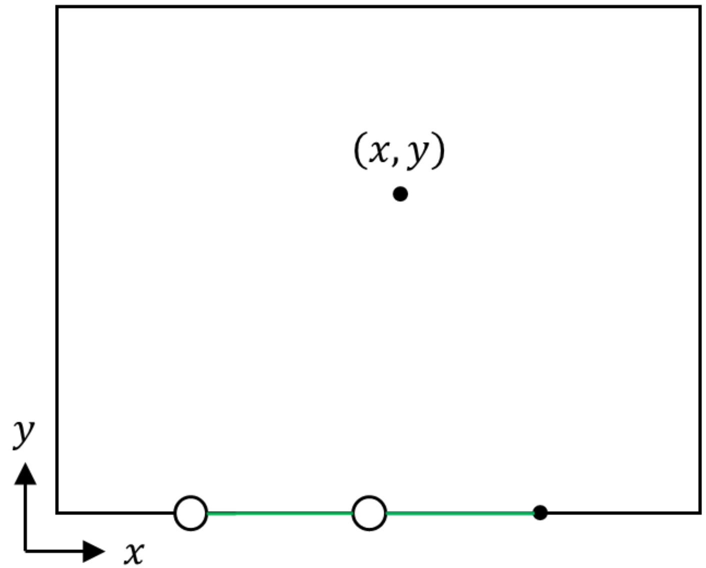
Inverse kinematics of a serial robot

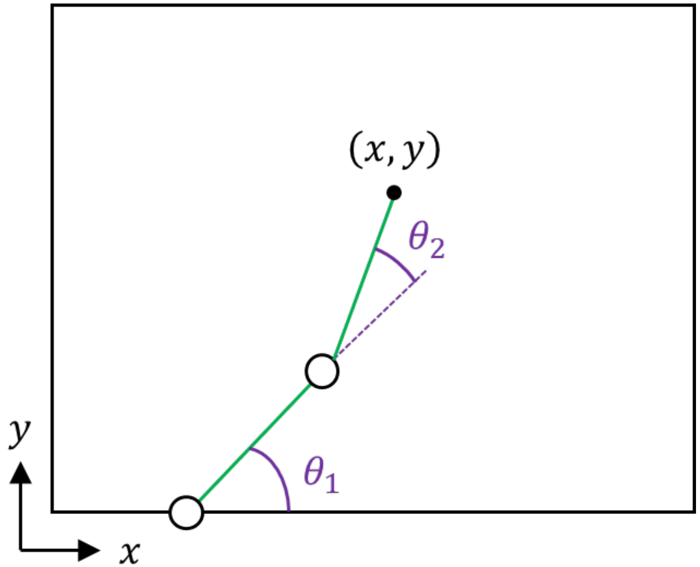
Reminder → The inverse kinematics is the relation between the workspace of the robot $\mathcal{W}$ with its configuration space $\mathcal{C}$, $\mathcal{W} \rightarrow \mathcal{C}$.

Inverse kinematics of a serial robot

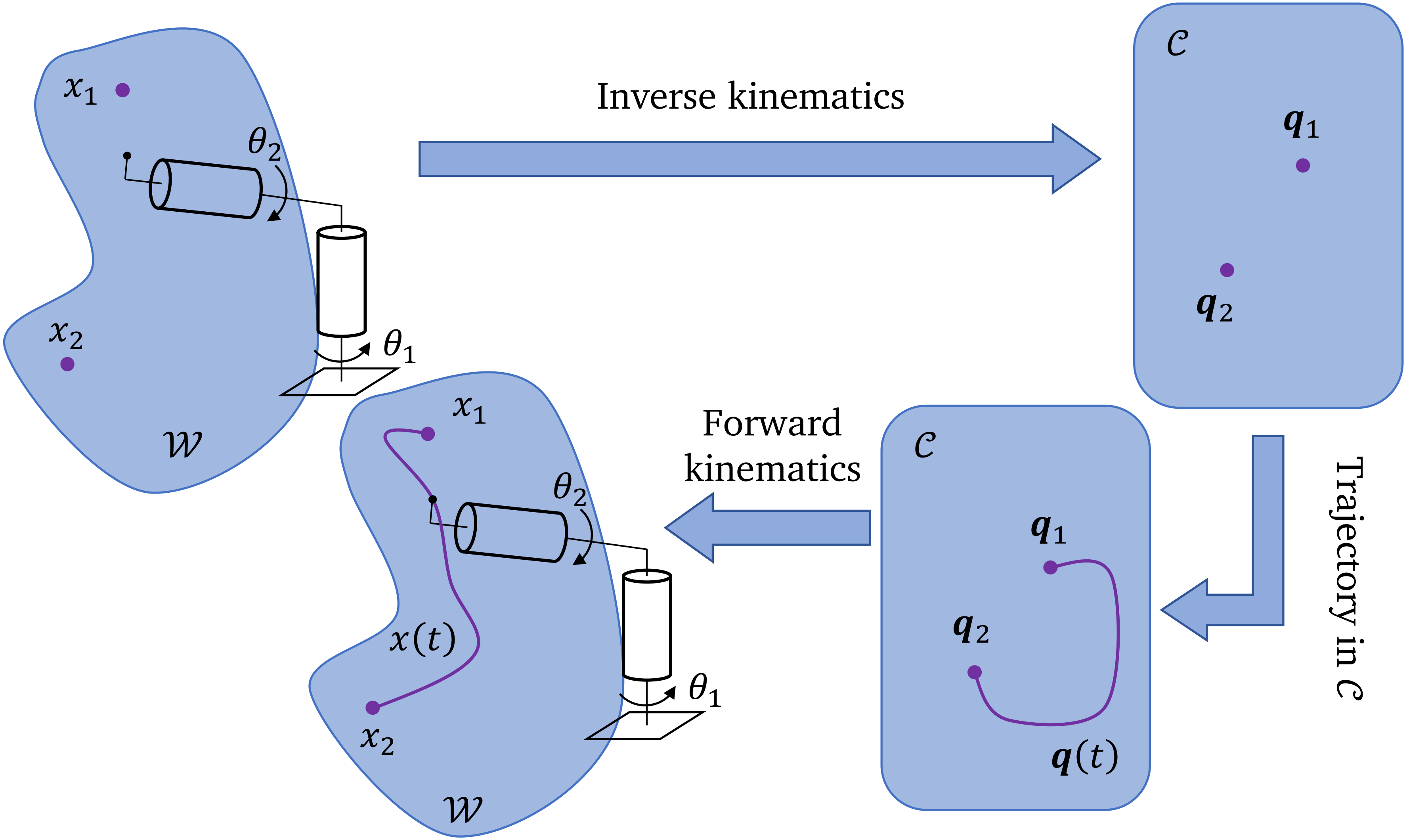
Reminder → The inverse kinematics is the relation between the workspace of the robot $\mathcal{W}$ with its configuration space $\mathcal{C}$, $\mathcal{W} \rightarrow \mathcal{C}$.

Inverse kinematics of a robot → Kinematic problem consisting of determining the configuration or configurations of the robot that result in the end-effector having a predetermined position and orientation.





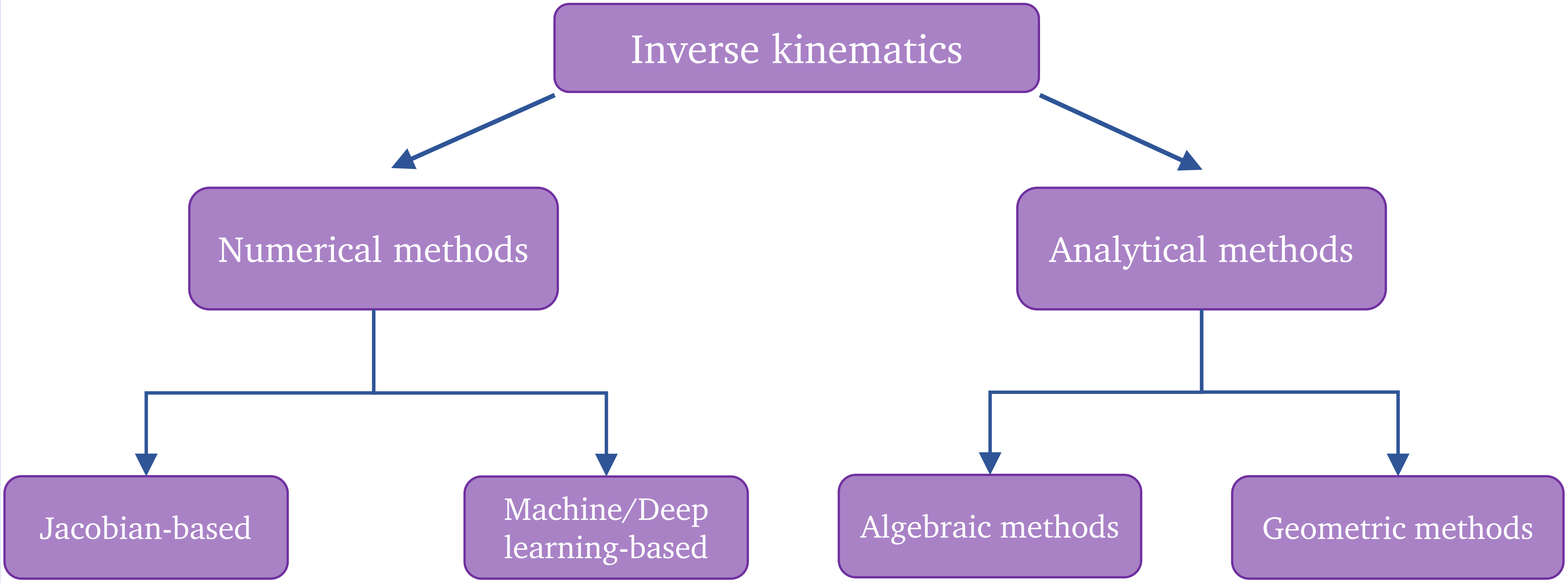
Why inverse kinematics is so important?



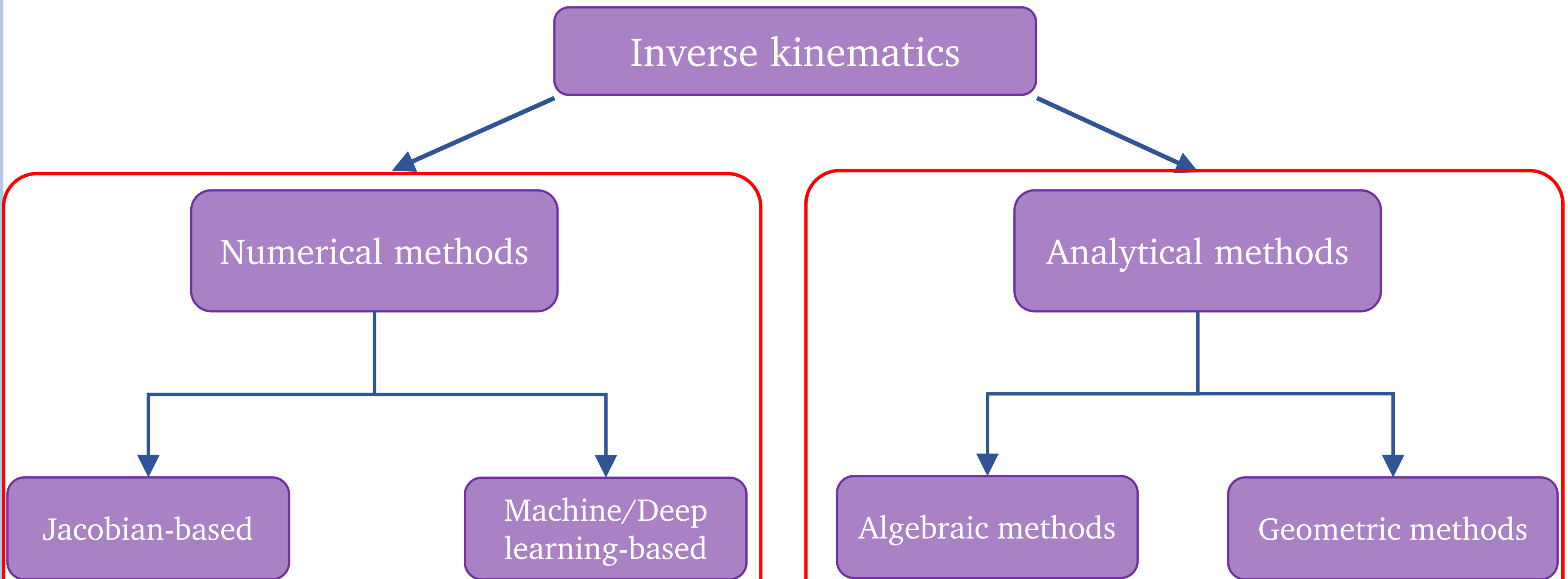
Why inverse kinematics is so important?

It rules robot's motion!

Inverse kinematics of a serial robot



Inverse kinematics of a serial robot



Numerical methods

- Only approximate one solution.
- Suffer from local minima and numerical instabilities.

Analytical methods

- Provide the set of all solutions.
- Difficult to compute for robots with a complex mechanical structure.

What is already known

- D. Pieper → Serial robots with a spherical wrist always have closed-form solution; IK can be divided into the inverse position and inverse orientation subproblems.
- R. Paul → Method to obtain the closed-form solutions of arbitrary serial robots; in practice only works for simple robotic structures.
- GA-based methods → Method to obtain closed-form solutions of serial robots with a spherical wrist; reduce the computation of the joint variables to computing a pre-assigned Euclidean point for each joint (by means of defining and manipulating different geometric entities in CGA).
  - D. Hildenbrand, J. Zamora, and E. Bayro-Corrochano.
  - A. Kleppe, and O. Egeland.
  - I. Zaplana, H. Hadfield, and J. Lasenby.

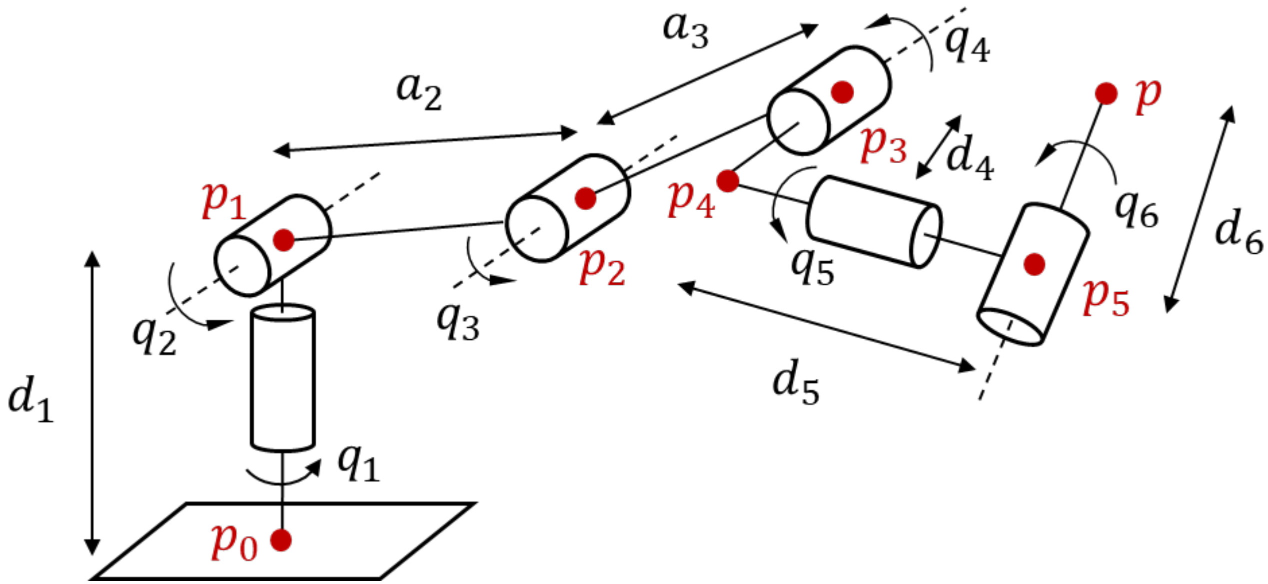
Some CGA notation to understand each other well

- e_0 and e_∞ → Null basis vectors associated with the origin and the point at the infinity.
- p → Null vector representation of the Euclidean point $\boldsymbol{p}$.
- $R(\theta, B) = \cos(\theta/2) - \sin(\theta/2)\, B$ → Rotor encoding a rotation by an angle θ in the rotation plane represented by the bivector B.
- $T(d, \boldsymbol{v}) = 1 - d\frac{ve_\infty}{2}$ → Rotor encoding a translation by an amount d along the direction of the vector $\boldsymbol{v}$.
- G → Geometric entity with inner representation O, i.e., $G = \{\boldsymbol{x} \mid x \cdot O = 0\}$, and outer representation O^*, i.e., $G = \{\boldsymbol{x} \mid x \wedge O^* = 0\}$.
- $G_1 \vee G_2 = (O_1 \wedge O_2)^*$ → Intersection between the geometric entities G_1 and G_2 (with inner representations O_1 and O_2).

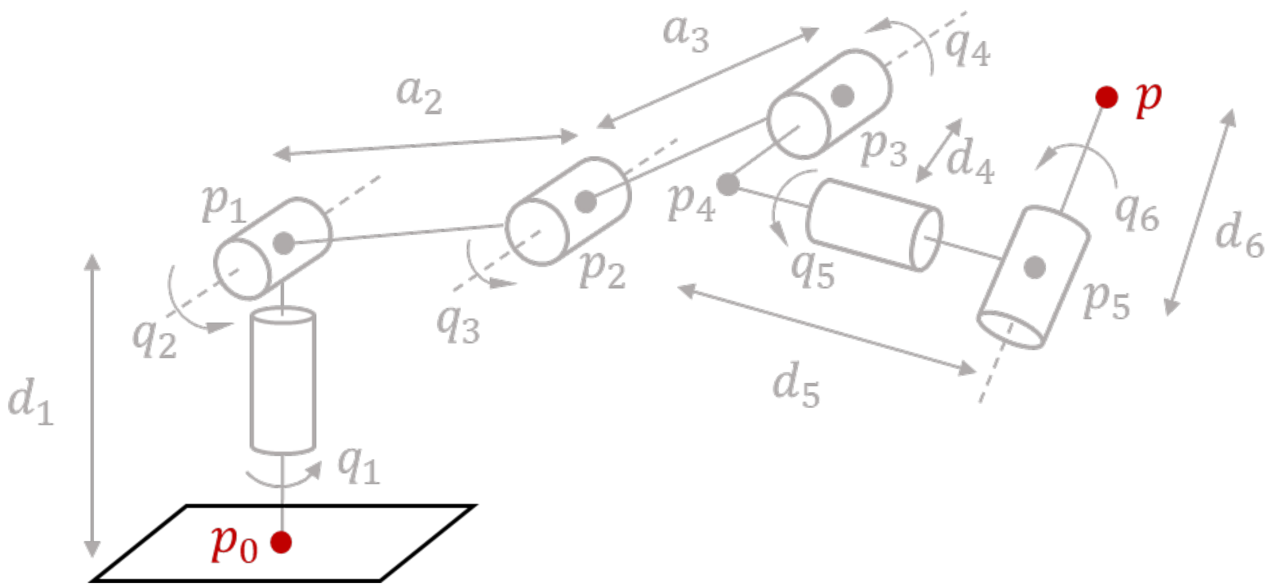
Some CGA notation to understand each other well

	Inner representation (O)	Outer representation (O^*)
Pair of points $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$		$b^* = p_1 \wedge p_2$
Line passing through points $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$, and with direction vector $\boldsymbol{v}$	$\ell = \boldsymbol{v}e_{\mathbf{123}} - (\boldsymbol{p}_1 \wedge \boldsymbol{v})e_{123}e_\infty$	$\ell^* = p_1 \wedge p_2 \wedge e_\infty$
Plane passing through points $\boldsymbol{p}_1, \boldsymbol{p}_2$, and $\boldsymbol{p}_3$, and with with normal vector $\boldsymbol{n}$ and orthogonal distance to the origin δ	$\pi = \boldsymbol{n} - \delta e_\infty$	$\pi^* = p_1 \wedge p_2 \wedge p_3 \wedge e_\infty$
Sphere passing through points $\boldsymbol{p}_1, \boldsymbol{p}_2$, $\boldsymbol{p}_3$, and $\boldsymbol{p}_4$, and with centre $\boldsymbol{c}$ and radius r	$s = c - \frac{1}{2}r^2\, e_\infty$	$s^* = p_1 \wedge p_2 \wedge p_3 \wedge p_4$

Solution strategy

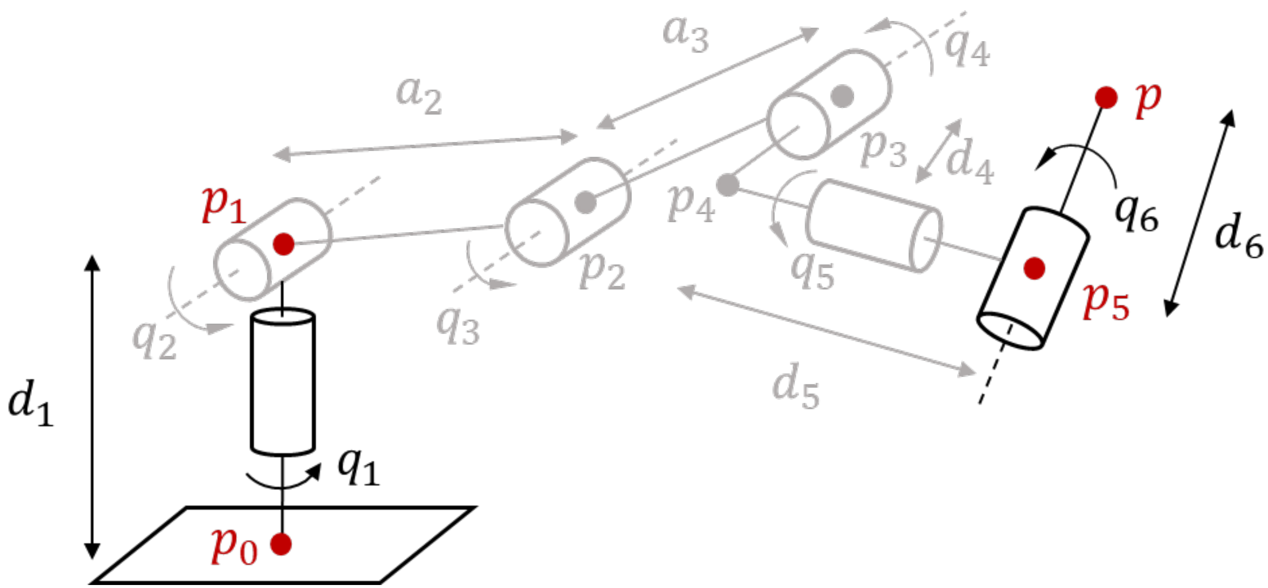


Solution strategy



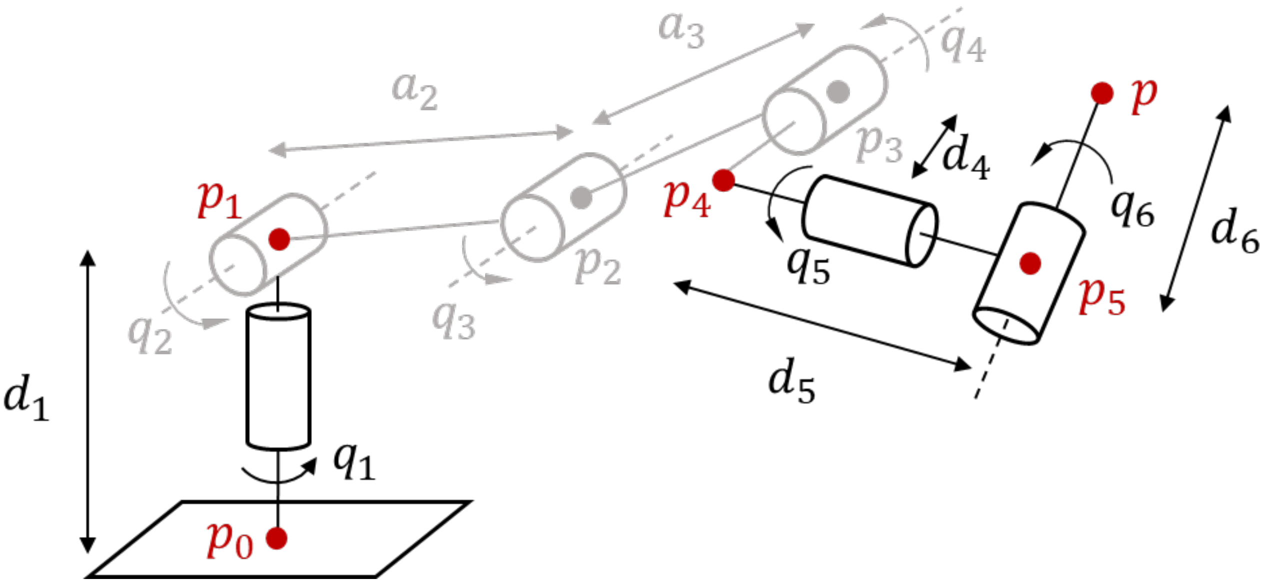
- Input information:
  - Desired position and orientation, represented by the null vector p and the Euclidean vector $\boldsymbol{z}$.
  - Reference frame, associated with the base of the robot, represented by the null vector p_0 and Euclidean vectors $\boldsymbol{x}_0, \boldsymbol{y}_0$ and $\boldsymbol{z}_0$.

Solution strategy



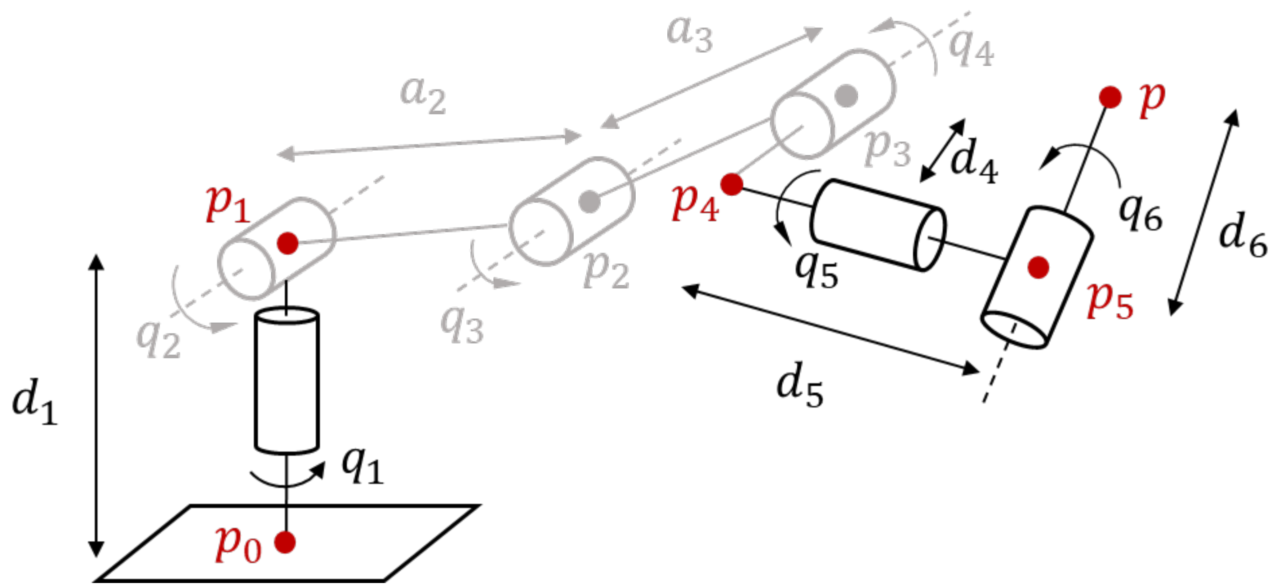
- First step → Computation of points p_1 and p_5:
  - $p_5 = T(-d_6, \boldsymbol{z})p\tilde{T}(-d_6, \boldsymbol{z})$ where $T(-d_6, \boldsymbol{z}) = 1 + d_6 \frac{\boldsymbol{z}e_\infty}{2}$
  - $p_1 = T(d_1, \boldsymbol{z}_0)p_0\tilde{T}(d_1, \boldsymbol{z}_0)$ where $T(d_1, \boldsymbol{z}_0) = 1 - d_1 \frac{\boldsymbol{z}_0 e_\infty}{2}$

Solution strategy



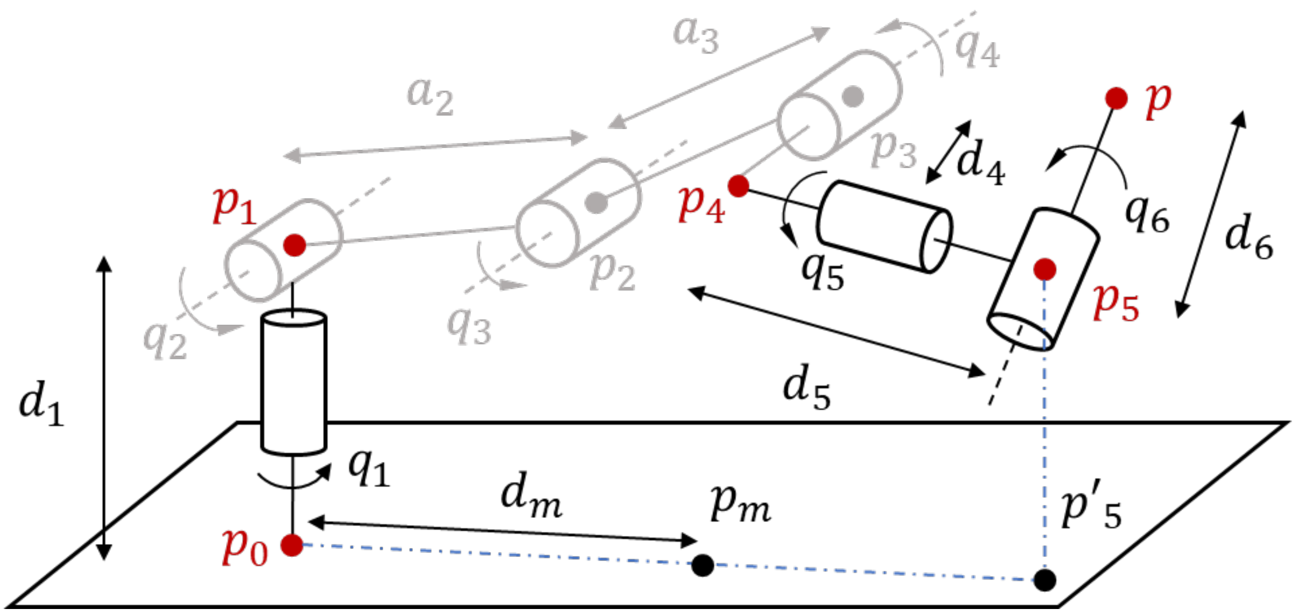
- Second step → Computation of point p_4, which lies on the intersection between:
  - A sphere centred at $\boldsymbol{p}_5$ and with radius d_5 → S_5.
  - A plane with normal vector $\boldsymbol{n}_5 = \boldsymbol{p} - \boldsymbol{p}_5$ containing $\boldsymbol{p}_5$ → Π_5.
  - A vertical plane that contains $\boldsymbol{p}_5$ with orthogonal distance to the origin $\delta = d_4$ → Π_4.

Solution strategy



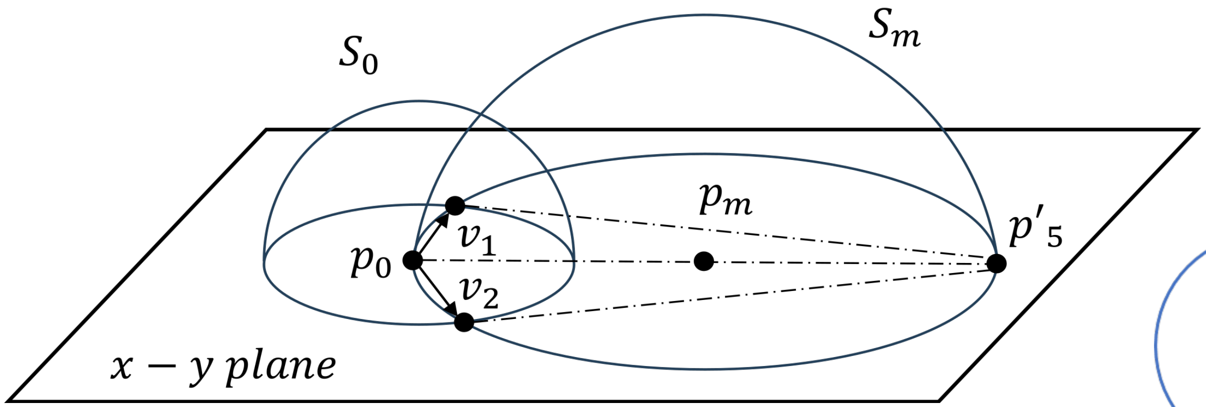
- Inner representations of S_5 and Π_5:
  - $s_5 = p_5 - \frac{1}{2} d_5^2 e_\infty$
  - $\pi_5 = \boldsymbol{n}_5 - \boldsymbol{p}_5 \cdot \boldsymbol{n}_5 e_\infty$

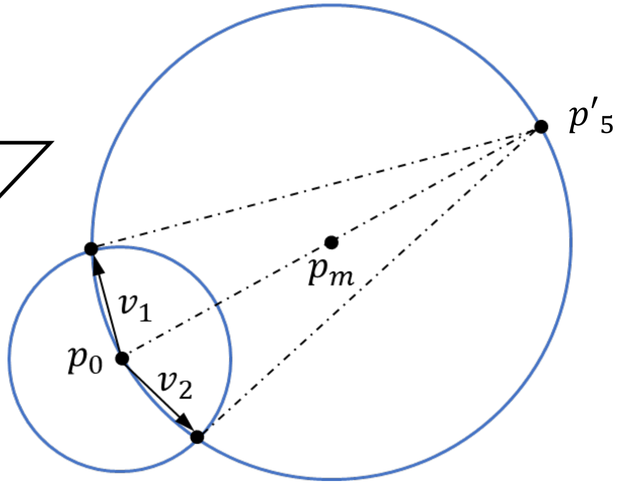
Solution strategy



- To compute the inner representation of plane Π_4, point $\boldsymbol{p}_5$ is first projected onto the $x - y$ plane of the reference frame and:
  - $\boldsymbol{p}_m = \frac{\boldsymbol{p'}_5}{2}$ → Middle point between the origin of the reference frame and the projected point $\boldsymbol{p}'_5$.
  - $d_m = |\boldsymbol{p}_m|$.

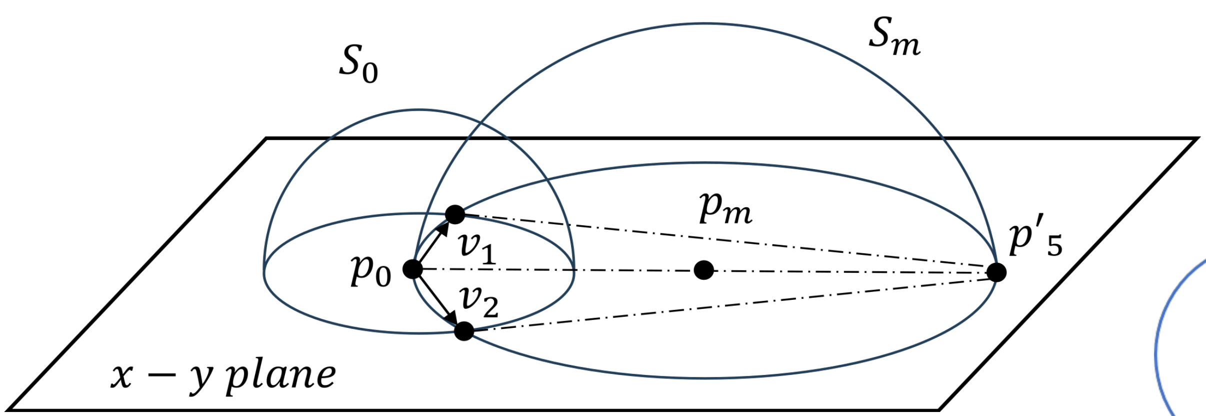
Solution strategy

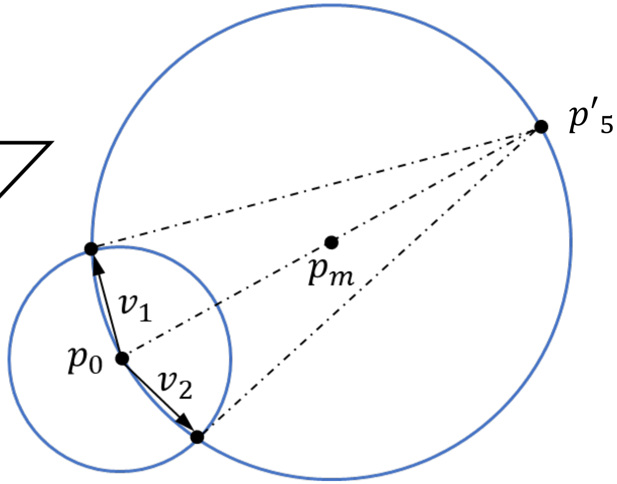




- Now, the inner representations of two spheres and a plane are computed:
  - $s_m = p_m - \frac{1}{2} d_m^2 e_\infty$
  - $s_0 = p_0 - \frac{1}{2} d_4^2 e_\infty$
  - $\pi_{xy} = \boldsymbol{z}_0$

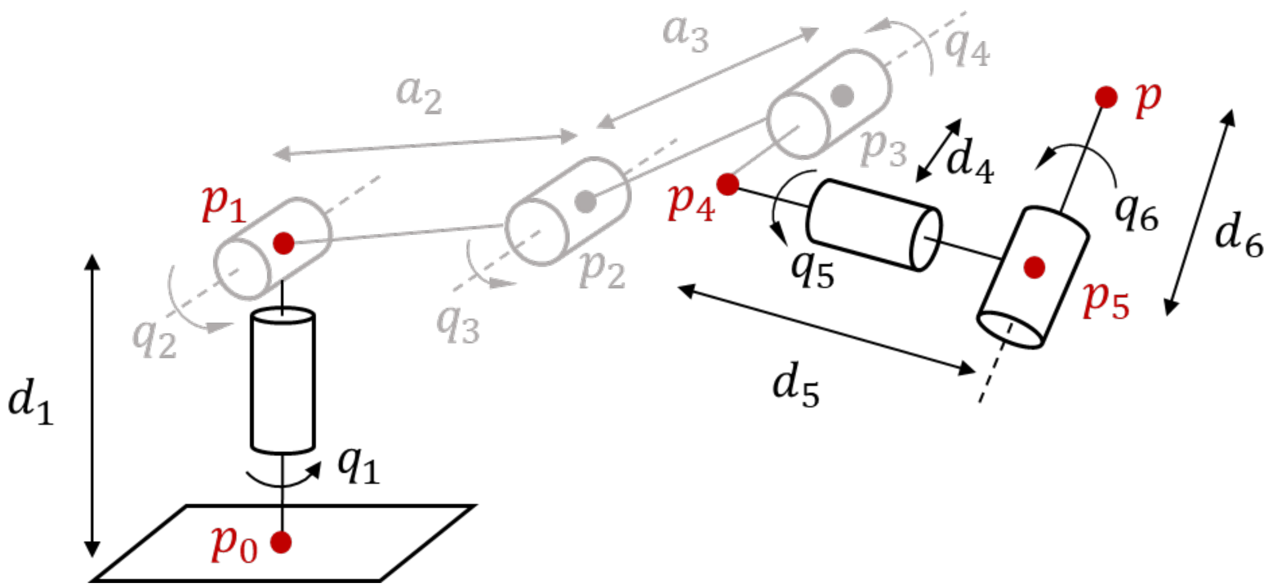
Solution strategy





- The intersection of these three entities, given by $(s_m \wedge s_0 \wedge \pi_{xy})^*$, is a bi-vector representing a pair points, which define two vectors, $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$.
- The inner representation of two different planes Π_4 can be defined as:
  - $\pi_4 = \boldsymbol{v}_1 - d_4 e_\infty$
  - $\pi_4 = \boldsymbol{v}_2 - d_4 e_\infty$

Solution strategy

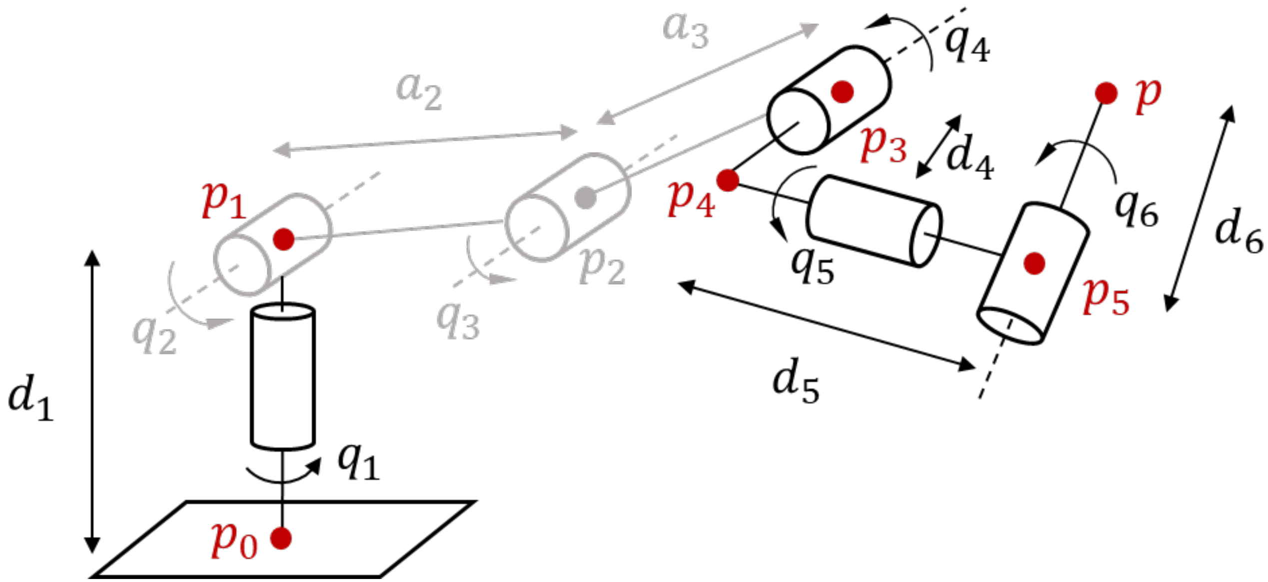


- Finally, p_4 is extracted from the bivector computed as the intersection of S_5, Π_5 and any of the two planes Π_4:

$$(s_5 \wedge \pi_5 \wedge \pi_4)^*$$

- This gives rise to up to four different points p_4, each of which leads to a distinct, yet valid, solution.

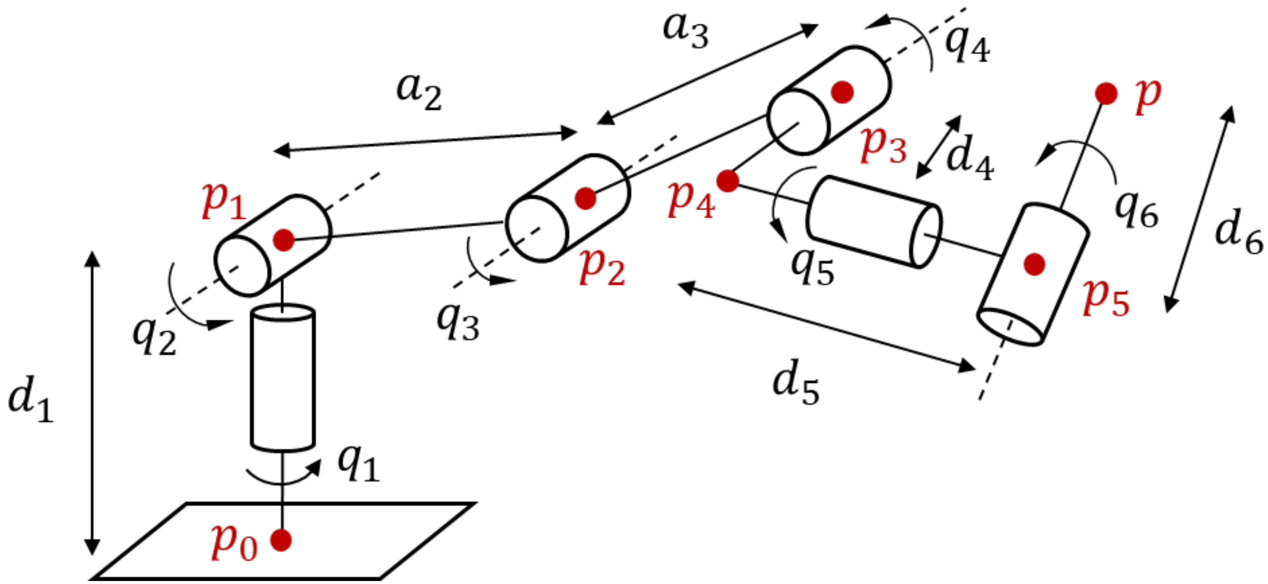
Solution strategy



- Third step → Computation of point p_3:
  - Two different points p_3, one with vector $\boldsymbol{v}_1$ and another with vector $\boldsymbol{v}_2$, can be computed as $p_3 = T(-d_4, \boldsymbol{v}_i)p_4\tilde{T}(-d_4, \boldsymbol{v}_i)$ where:

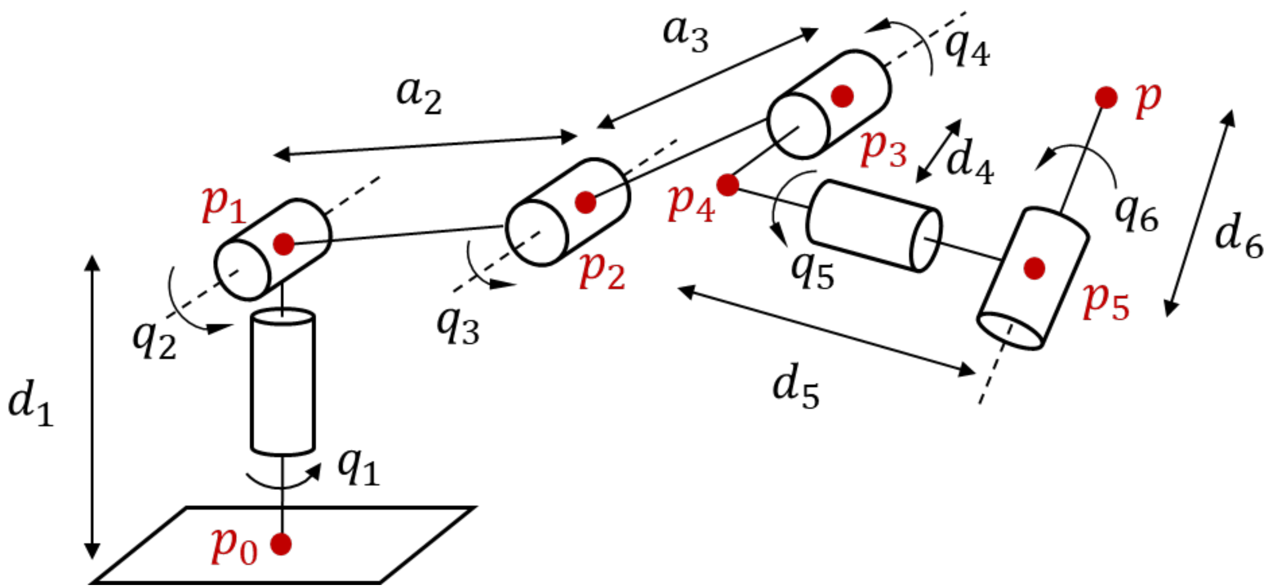
$$T(-d_4, \boldsymbol{v}_i) = 1 + d_4 \frac{\boldsymbol{v}_i e_\infty}{2} \quad (i = 1,2)$$

Solution strategy



- Fourth step → Computation of point p_2, which lies on the intersection between:
  - A sphere centred at p_1 and with radius a_2 → S_1.
  - A sphere centred at p_3 and with radius a_3 → S_3.
  - A plane passing by p_0, p_1 and p_3 → Π.

Solution strategy



- Inner representation of S_1, S_3 and Π:
  - $s_1 = p_1 - \frac{1}{2}a_2^2 e_\infty$
  - $s_2 = p_3 - \frac{1}{2}a_3^2 e_\infty$
  - $\pi = (p_0 \wedge p_1 \wedge p_3 \wedge e_\infty)^*$

Implementation, simulation and real execution

- The obtained geometric expressions have been implemented numerically using the Python library `clifford` and the Matlab library `Symbolic and User-friendly GA routines (SUGAR)`.

- `SUGAR` is a open-source Matlab toolbox recently developed that allows symbolic computations, therefore facilitating the calculation of closed-form solutions to these kind of geometric-oriented problems.

`clifford`'s Git repository

`SUGAR`'s Git repository

Implementation, simulation and real execution

Lissajous curve for position, continuous rotation around x axis of the reference frame for orientation

Parameterization of both position and orientation along the input curve

CGA-based IK!

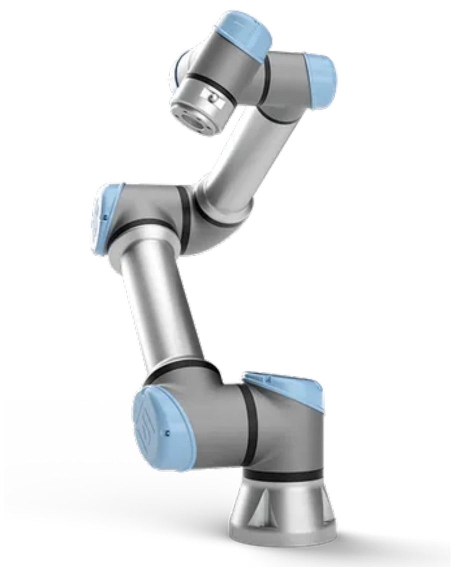
Trajectory in $\mathcal{C}$

Execution of the Robotic Operating System (ROS) action `follow _joint_trajectory`

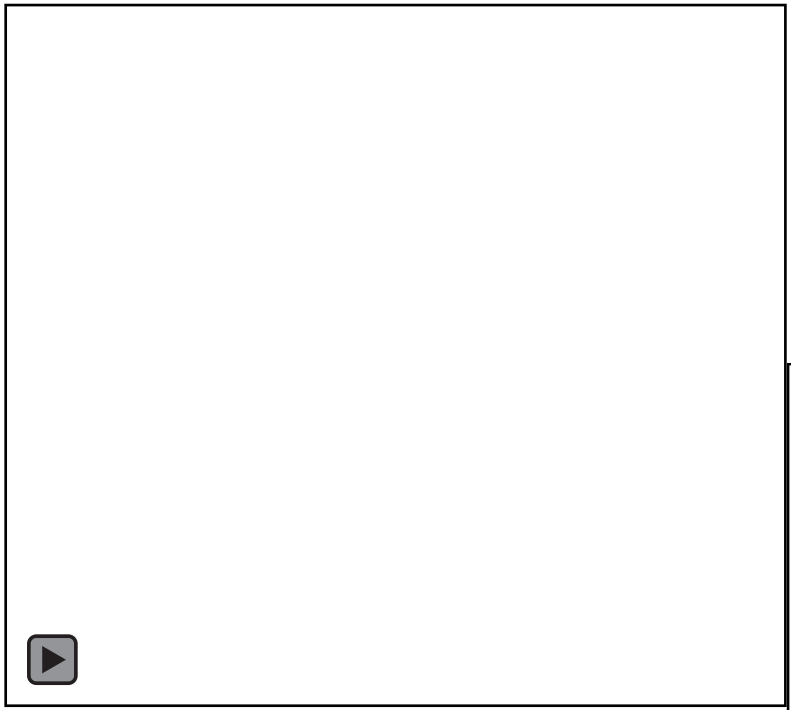
Simulation in Gazebo

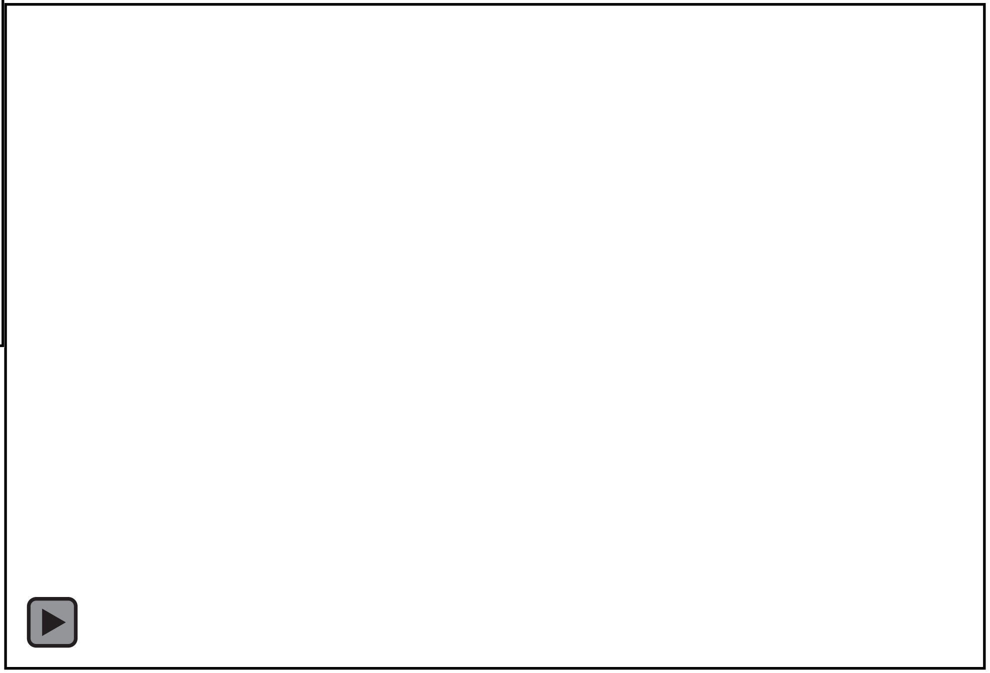
Execution in the real robot

Trajectory in $\mathcal{W}$



Implementation, simulation and real execution





And now, what?

GA-based numerical algorithm for position control.

Applications to the inverse kinematics of complex robotic structures, inc. redundant robots without a spherical wrist.

CGA-based formulation of the Paden-Kahan subproblems and extensions.

Thanks for your attention!