Development of intelligent systems (RInS)

Robot manipulation

Danijel Skočaj
University of Ljubljana
Faculty of Computer and Information Science

Literature: Tadej Bajd (2006) Osnove robotike, poglavje 4
Anže Rezelj (2017) Razvoj nizkocenovnega lahkega robotskega manipulatorja

Academic year: 2022/23

Robot manipulator

- Industrial robot as defined by the standard ISO 8373:

> An automatically controlled, reprogrammable, multipurpose manipulator programmable in three or more axes, which can be either fixed in place or mobile for use in industrial automation applications.

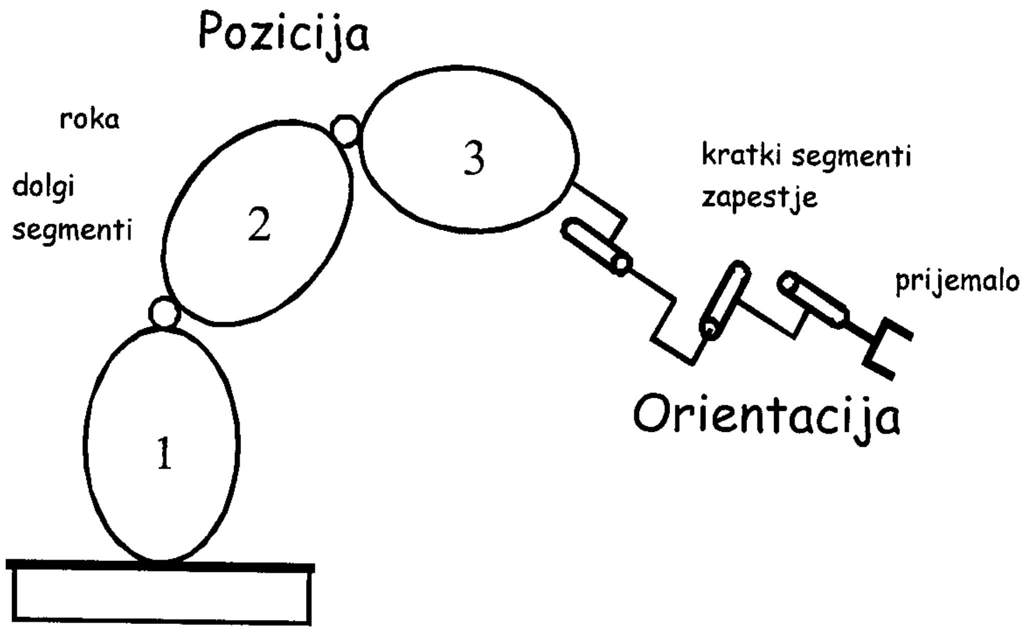



Characteristics

- Closed-loop control
  - Electrical or hydraulic motors
  - Sensors
    - Proprioceptive: rotation encoders, measurement od distance, speed
    - Exteroceptive: tactile sensors, robot vision
- Reprogrammable:
  - designed so that the programmed motions or auxiliary functions can be changed without physical alteration
- Multipurpose:
  - capable of being adapted to a different application with physical alteration
- Physical alteration:
  - alteration of the mechanical system
  - fixed or mobile robots
- Axis: direction used to specify the robot motion in a linear or rotary mode
  - 3 or more DOF

Robot manipulator

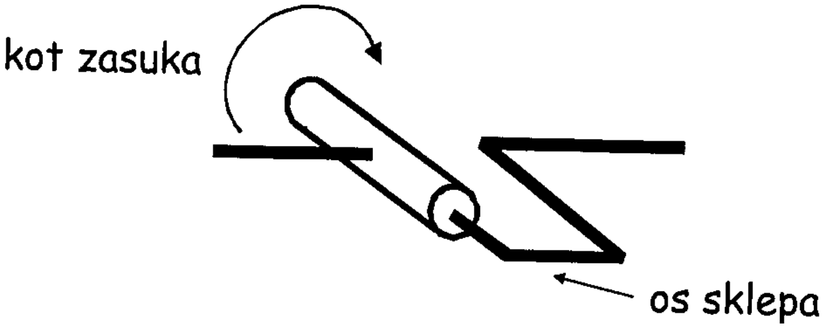
- Arm+wrist+end effector/gripper
- 6DOF – can put an object in an arbitrary pose
  - arm positions the object into the desired position
  - wrist rotates it into the desired orientation
  - gripper grasps the object



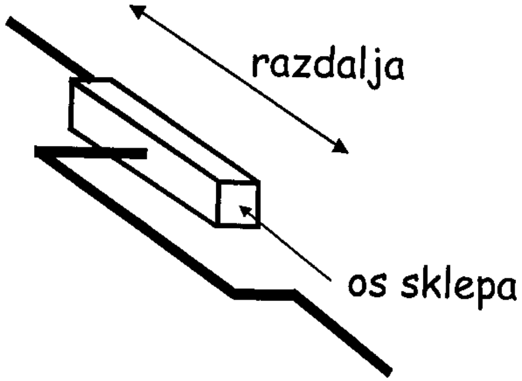
Robot arm

- Serial chain of three links
- connected by joints

- Revolute/rotational joint



- Prismatic/translational joint

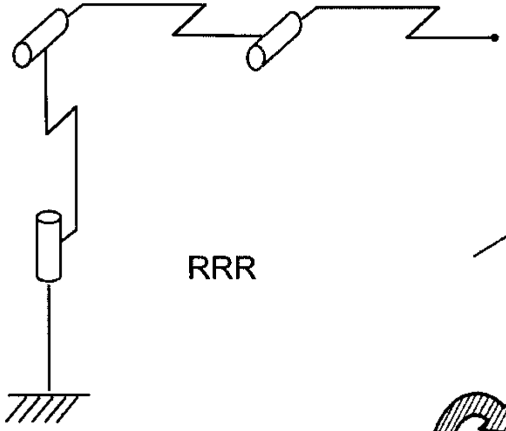


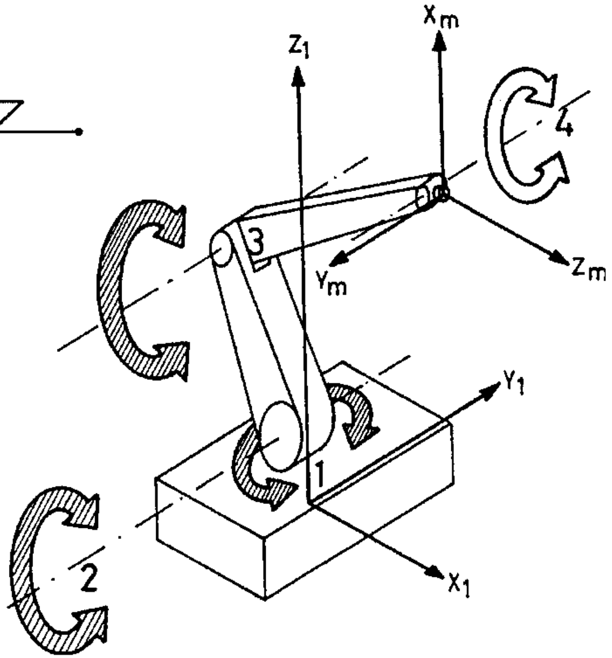
Robot arm types

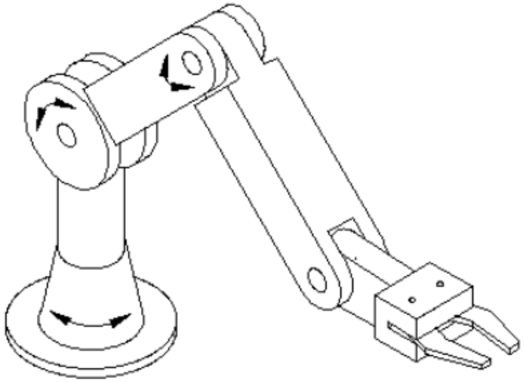
- Joints
  - Revolute/rotational
  - Prismatic/translational
- Axis of two neigbouring links
  - Parallel
  - Perpendicular
- 3DOF
- In practice typically five different arms:
  - Anthropomorphic
  - Spherical
  - SCARA
  - Cylindrical
  - Cartesian

Anthropomorphic robot arm

- Three rotational joints (RRR)
- Workspace: sphere-like
- Resembles a human arm

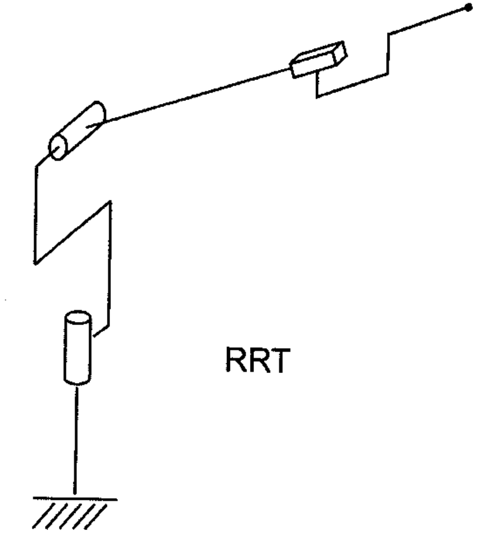


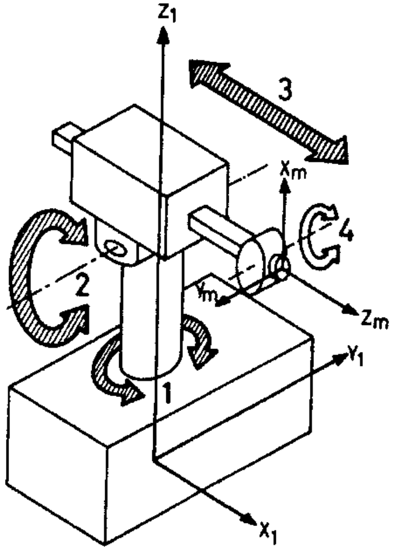


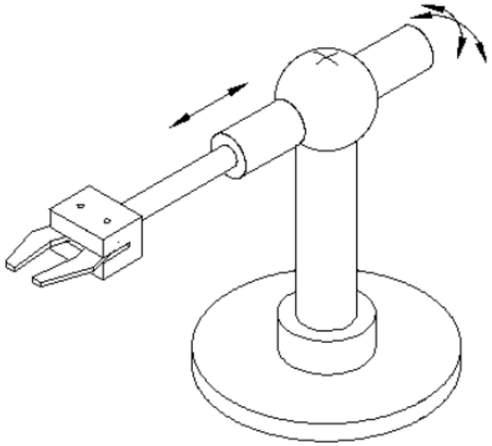


Spherical robot arm

- Two rotational, one translational joint (RRT)
- Workspace: sphere-like

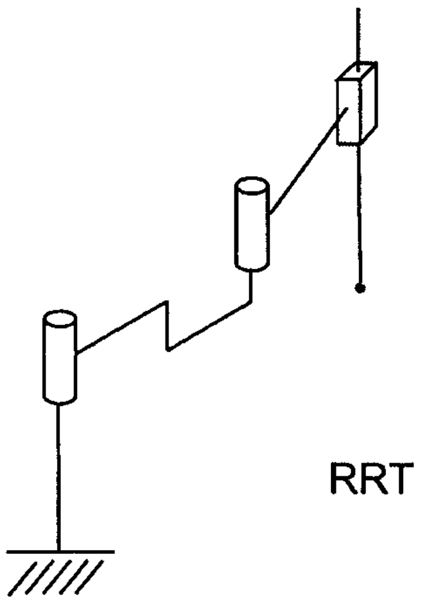


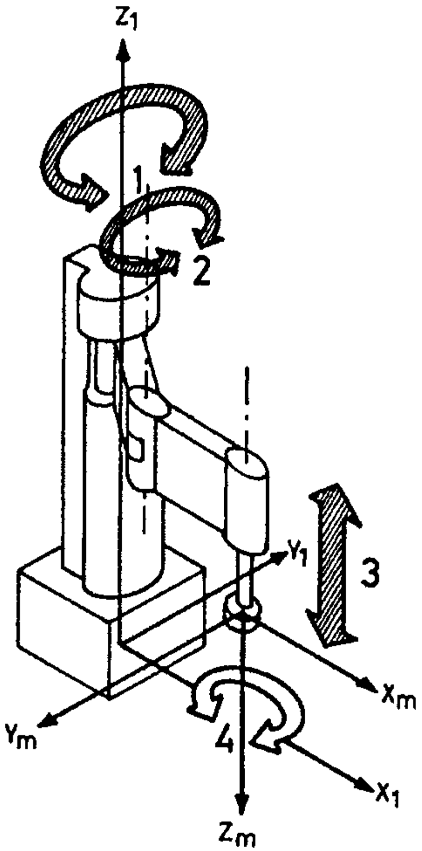


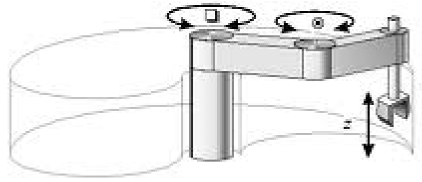


SCARA robot arm

- Selective Articulated Robot for Assembly
- Two rotational, one translational joint (RRT)
- Workspace: cylinder-like

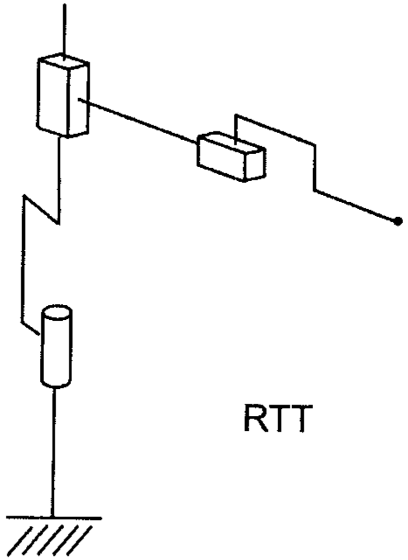


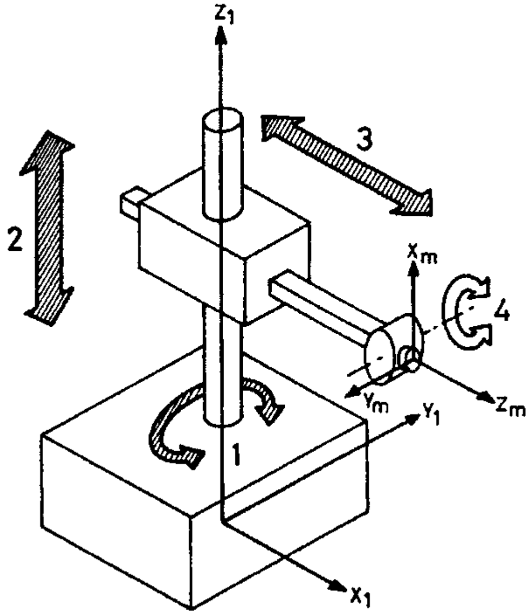


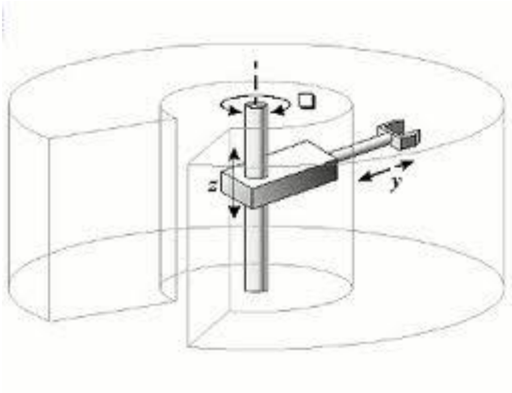


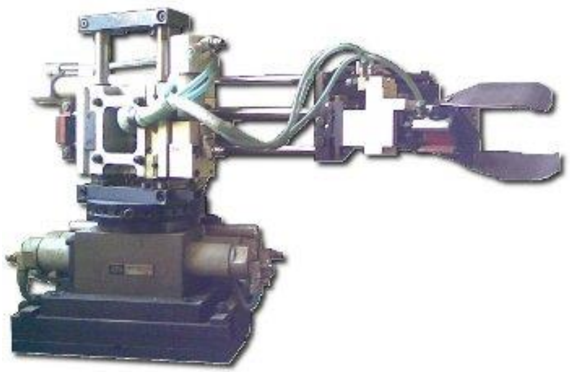
Cylindrical robot arm

- One rotational, two translational joints (RTT)
- Workspace: cylinder



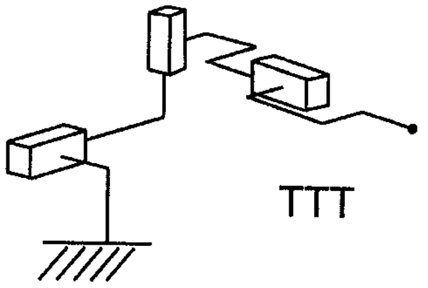


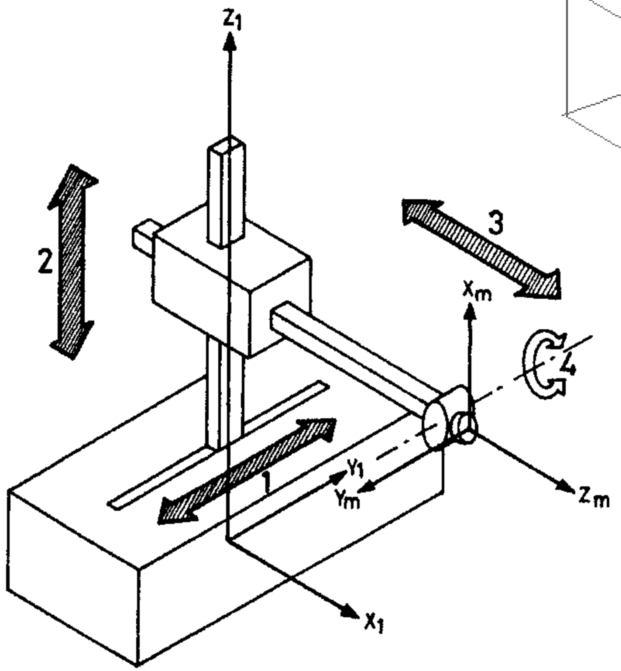


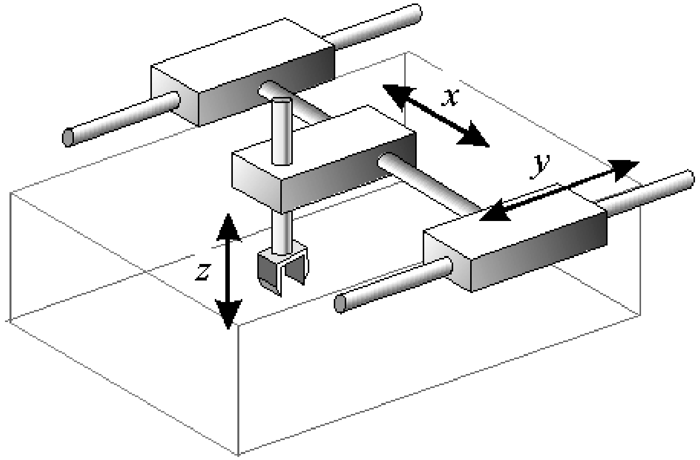


Cartesian robot arm

- Three translational joints (TTT)
- Workspace: cuboid

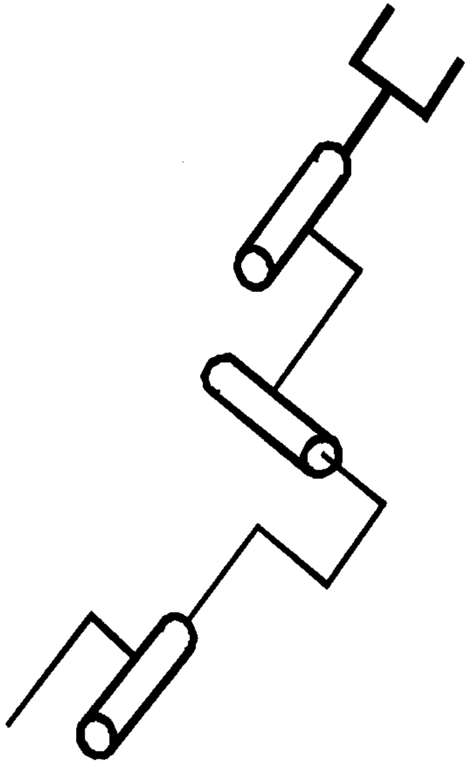






Robot wrist

- Rotates the object in an arbitrary orientation
- Three rotational joints (RRR)
  - Sometimes also one or two suffice
  - Links should be as short as possible



Robot end-effector

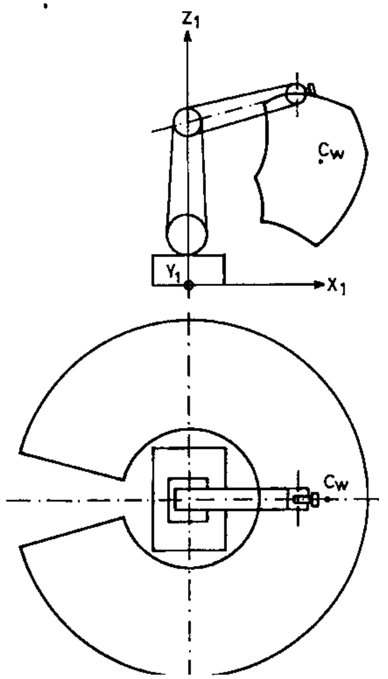
- The final link of the robot manipulator
  - Grippers with fingers
    - With two fingers
    - With more than two fingers
  - Other type of grippers
    - Vacuum
    - Magnetic
    - Perforation
  - Other tools as end-effectors
    - Welding gun
    - Spray painting gun

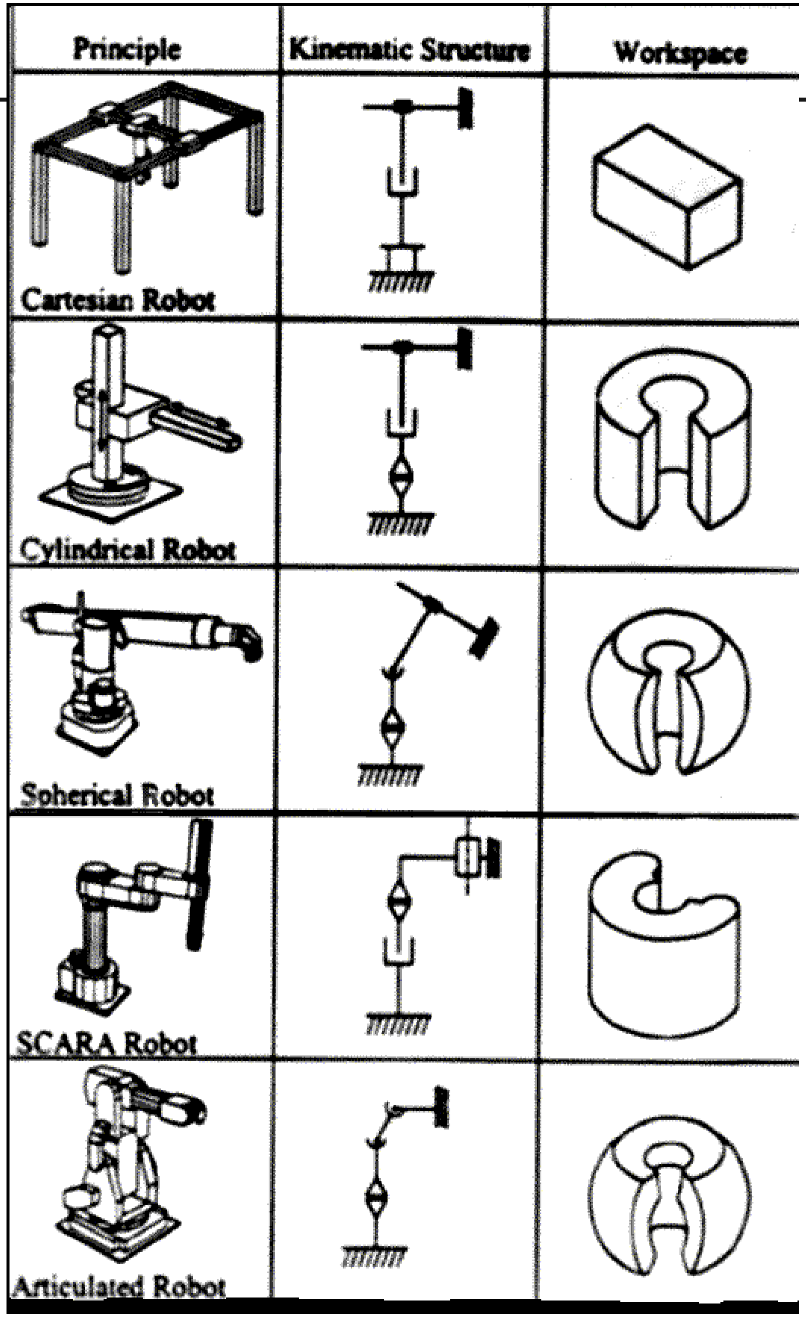





Robot workspace

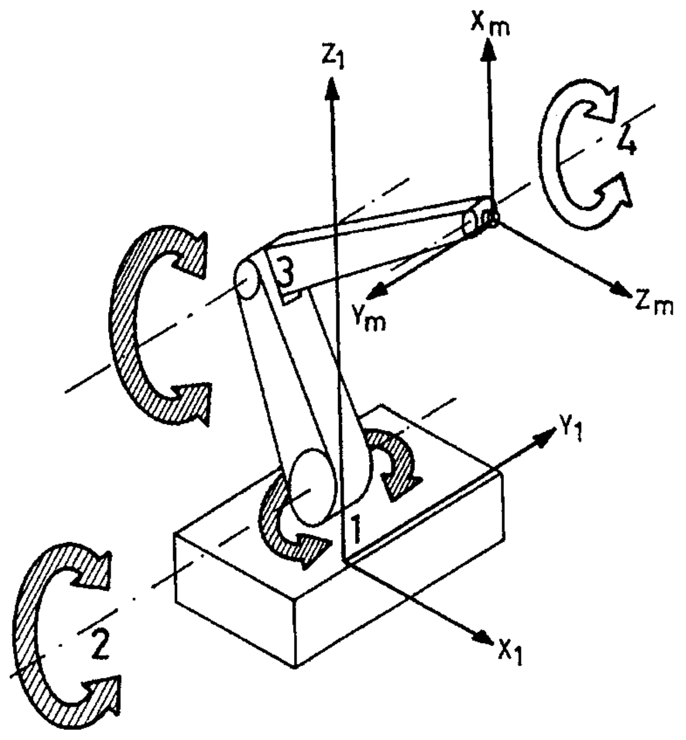
- Reachable workspace
  - The end-effector can reach every point in this space
- Dexterous workspace
  - The end-effector can reach every point in tis space from the arbitrary orientation





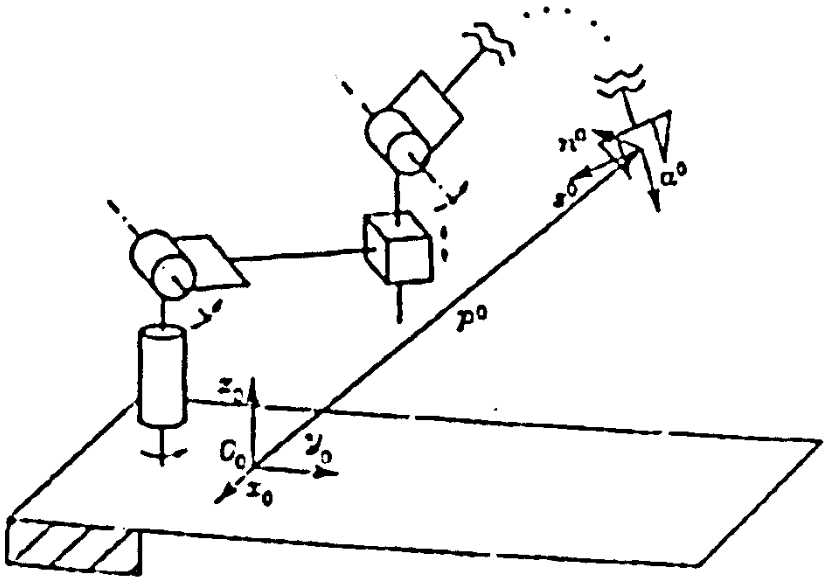
Kinematics

- Base coordinate frame $[X_1,Y_1,Z_1]$
  - Usually also world coordinate frame
  - Used for defining of the robotic task
- End-effector reference frame $[X_m,Y_m,Z_m]$
- End-effector position
  - Vector between the origins of the coordinate frames
- Object orientation
  - Three angles
- Internal robot coordinates / joint variables
  - Joint states (angles, translations)
  - Uniquely describe the pose of the robot
- Direct kinematics
  - Determine the external robot coordinates from the internal coordinates
- Inverse kinematics
  - Determine the internal robot coordinates from the external coordinates



Geometrical robot model

- Robot manipulator = a serial chain of segments connected by joints
- Every joint can be either rotational or translational
  - 1DOF – 1 internal coordinate



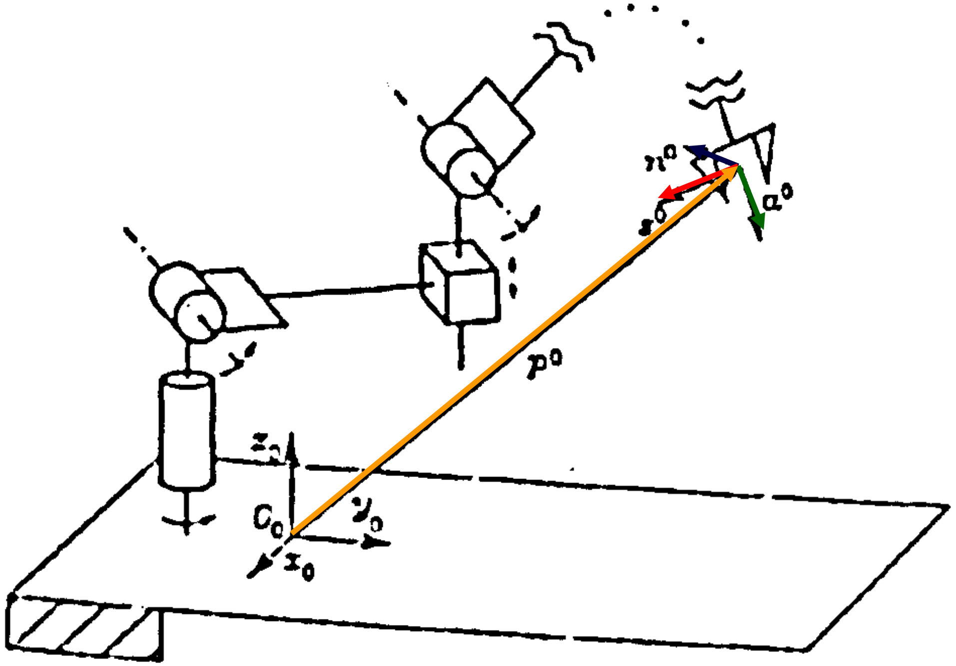
- Geometrical robot model describes
  - the pose of the last segment of the robot (end-effector) expressed in the reference (base) frame
  - depending on the current internal coordinates

Geometrical robot model

- Geometrical model can be expressed by a homogenuous transformation:

$$\mathbf{T}^o(\mathbf{q}) = \begin{bmatrix} \mathbf{n}^o(\mathbf{q}) & \mathbf{s}^o(\mathbf{q}) & \mathbf{a}^o(\mathbf{q}) & \mathbf{p}^o(\mathbf{q}) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

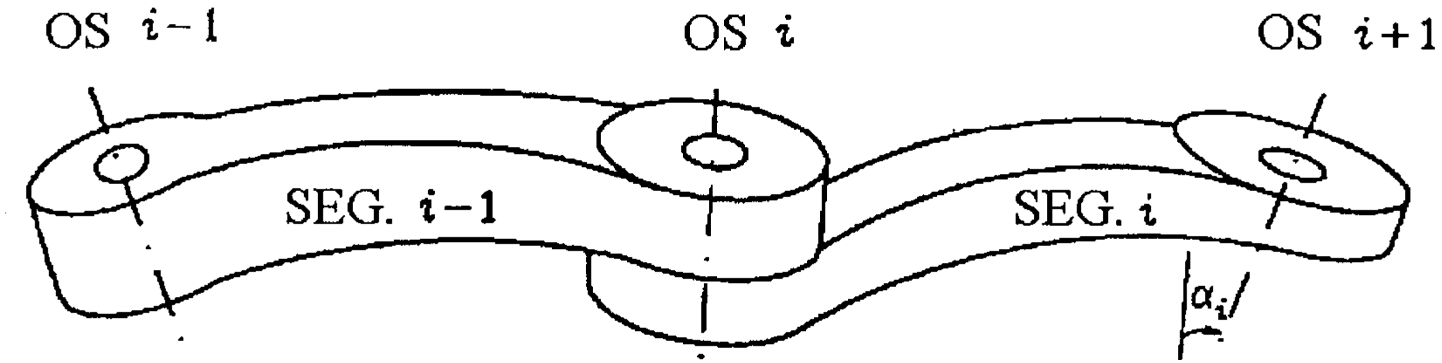
  - **p** : position of the end effector in the reference coordinate frame
  - **n**, **s**, **a** : unit vectors of the end-effector coordinate frame:
    - **a**: approach
    - **s**: sliding
    - **n**: normal
  - **q** : vector of internal coordinates



Poses of segments

- Every joint connects two neighbouring segments/links
  - Determine the transformation between them
  - Recursively build the full model for the entire robot
- Coordinate frames can be arbitrarily attached to the individual segments

- Denavit – Hartenberg rules simplify computation of the geometrical robot model
  - Determine the pose of the i-th c.f. with respect to the pose of the (i-1)-th c.f.
  - Axis i connects segments (i-1) and i



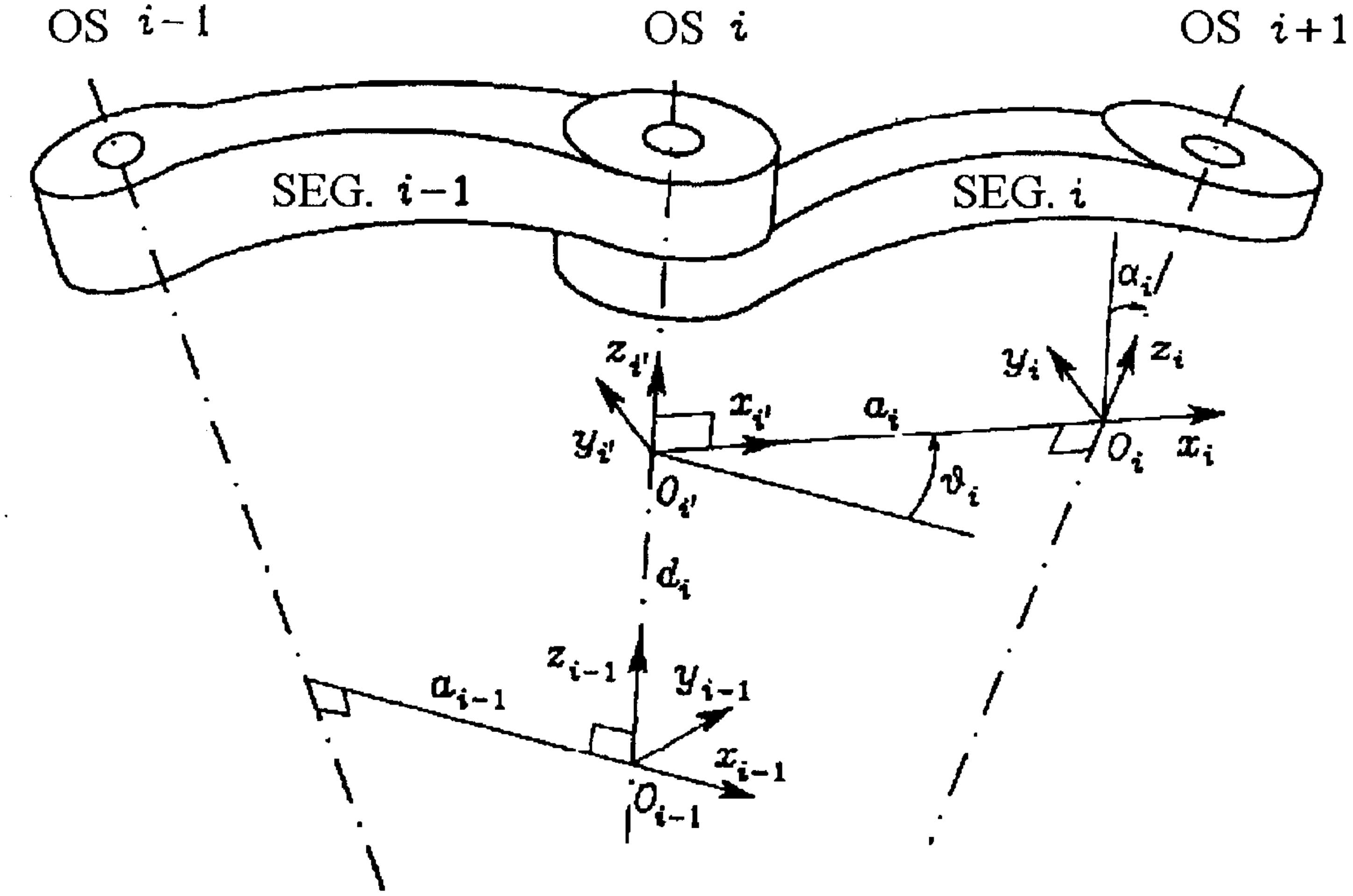
Denavit – Hartenberg rules

- Describe the coordinate frame of the i-th segment (having the joint i+1):

1. Define the axis z_i through the axis of the joint i+1
2. Find the common normal, perpendicular to the axes z_{i-1} and z_i
   - Position the origin of O_i into the intersection of the axis z_i with the common normal
   - Position the origin of $O_{i`}$ into the intersection of the axis z_{i-1} with the common normal
   - If the axes are parallel, position the origin anywhere
3. Position the axis x_i on a common normal in a way, that it is oriented from the joint i towards the joint i+1
   - If the axis z_{i-1} and z_i intersect, orient the axis x_i perpendicular to the plane defined by the axes z_{i-1} in z_i
4. Determine the axis y_i in a way that gives the right-handed c.f.

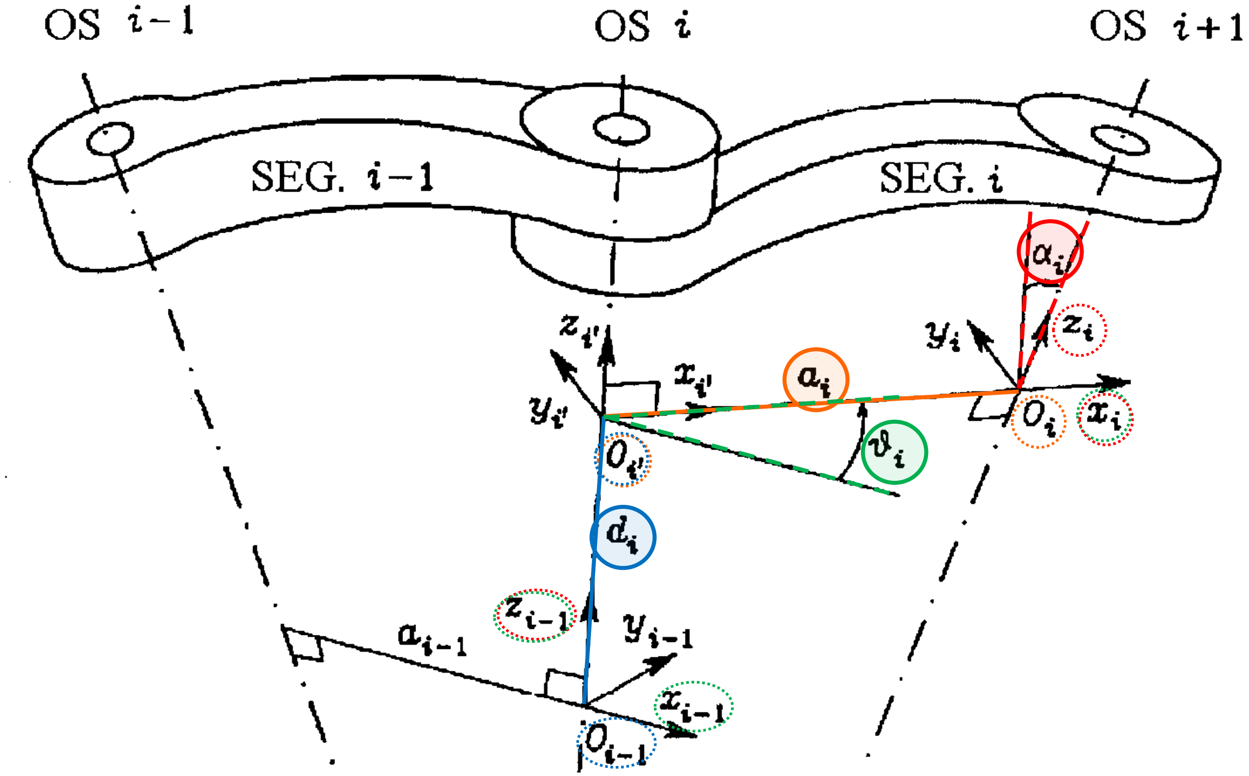
- Similarly we also describe (have already described) the coordinate frame of the segment (i-1)
  - The origin O_{i-1} is determined by the intersection of the common normal of the axes i-1 and i
  - The axis z_{i-1} is oriented along the i-th axis
  - x_{i-1} is oriented along the common normal and directed from the joint i-1 towards the joint i

Graphical illustration of DH parameters

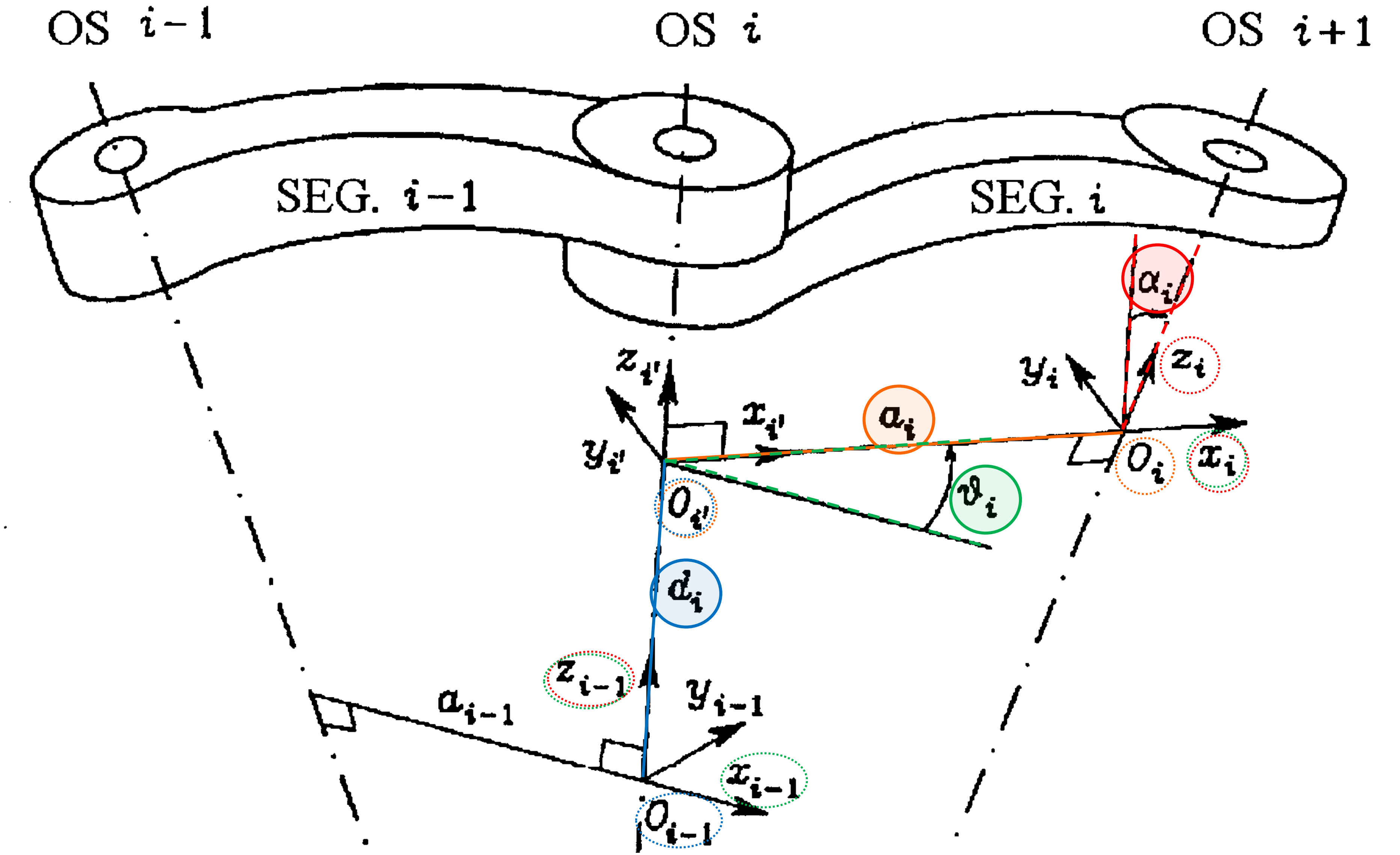


Denavit – Hartenberg parameters

- The pose of i-th c.f. with respect to (i-1)-th c.f. is determined by 4 parameters:
  1. a_i – distance between O_i and $O_{i'}$ along x_i
  2. d_i – distance between O_{i-1} and $O_{i'}$ along z_{i-1}
  3. α_i – angle between z_{i-1} and z_i around x_i
  4. θ_i – angle between x_{i-1} and x_i around z_{i-1}



Denavit – Hartenberg parameters

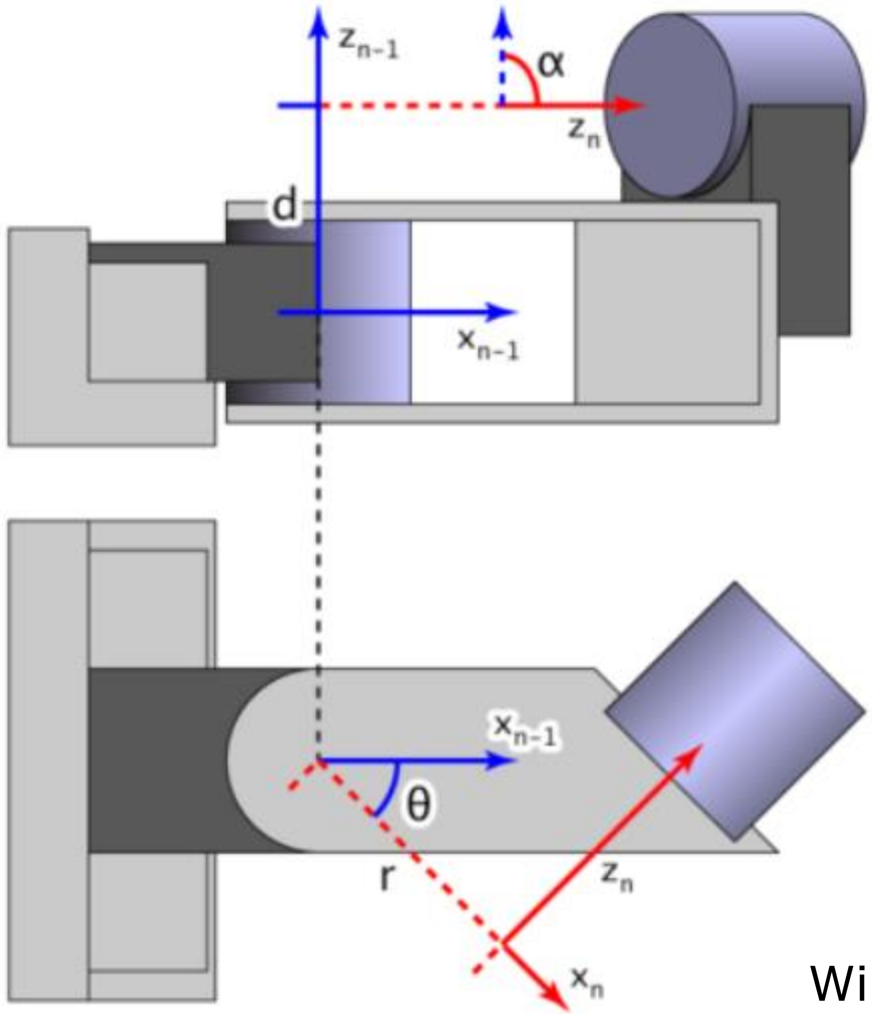


Denavit – Hartenberg parameters

- a_i and α_i are always constant
  - They depend on the geometry of the robot, the links between the joints, etc.
  - They do not change during the operation of the robot
- One of the two remaining parameters is a variable
  - Θ_i, if the i-th joint is rotational
  - d_i, if the i-th joint is translational

Denavit – Hartenberg parameters

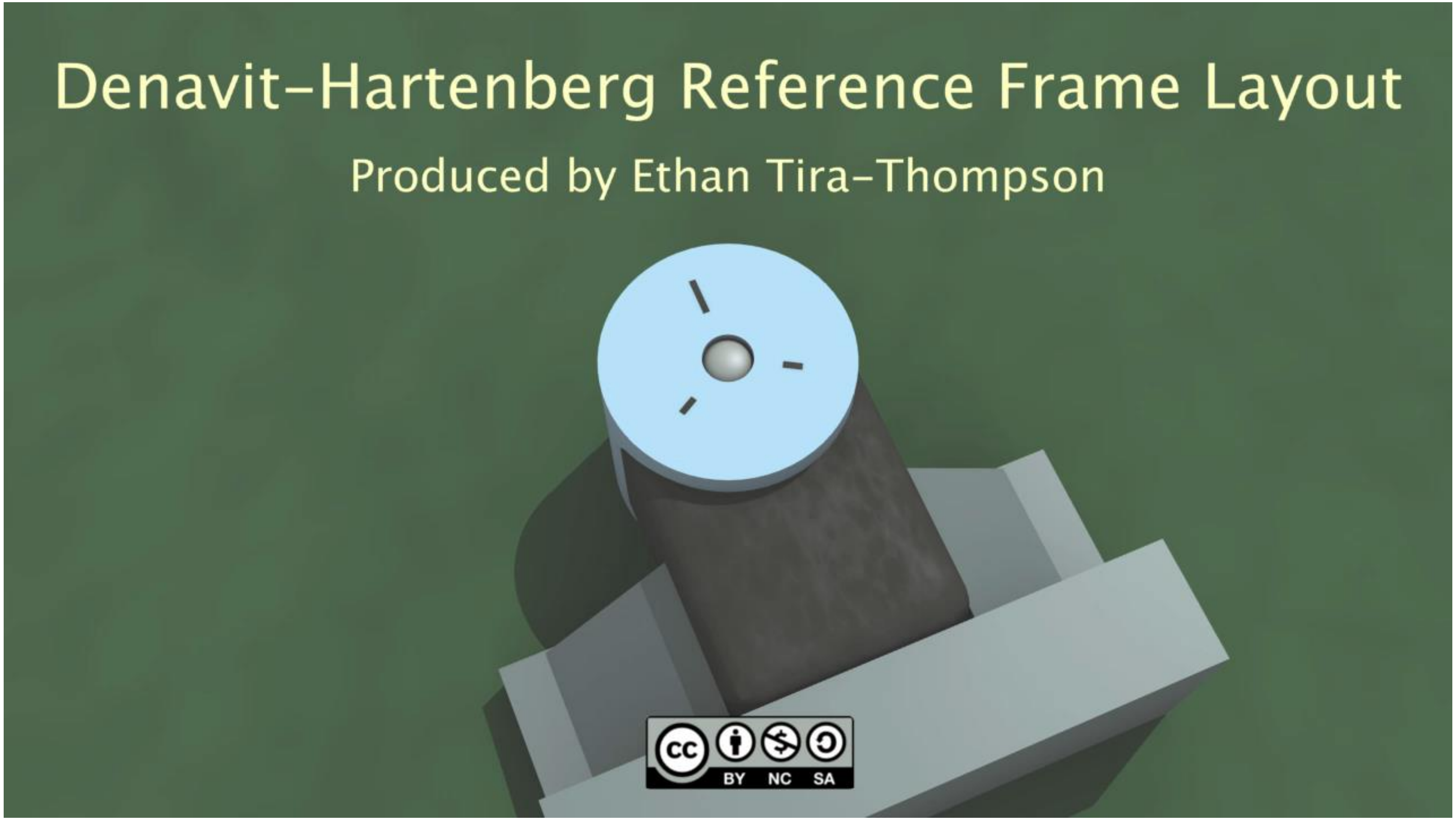
- Illustration
  - r=a
  - n=i



Wikipedia

Denavit – Hartenberg parameters

- Video:



http://en.wikipedia.org/wiki/Denavit-Hartenberg_Parameters

Exceptions

- Some exceptions in certain situations can be used to simplify the process:
  - Axis z_i and z_{i-1} are parallel -> $d_i=0$
  - Axis z_i and z_{i-1} intersect -> O_i is in the intersection
  - In case of the base (0-th) segment: only the axis z_0 is defined
    -> put the origin of O_0 in the first joint
    -> align x_0 and x_1
  - In case of end-effector (n-th c.f.):
    Only axis x_n is defined; it is perpendicular to z_{n-1}
    -> z_n should be parallel to z_{n-1}
  - In case of translational joint:
    -> orient the axis z_{i-1} in the direction of translation
    -> position O_{i-1} at the beginning of translation

Denavit – Hartenberg transformation

- Transformation between the i-th and (i-1)-th c.f.:

1. Take the c.f. O_{i-1} attached to the segment (i-1)
2. Translate it for d_i and rotate it for Θ_i along and around z_{i-1}, to align it with the c.f. $O_{i`}$
3. Translate c.f. $O_{i`}$ for $a_i$ in rotate it for $\alpha_i$ along and around $x_{i`}$, to align it with the c.f. O_i
4. DH transformation is obtained by postmultiplication of both transformation matrices
   - Function of a single variable:
     - Θ_i for rotational joint
     - d_i for translational joint

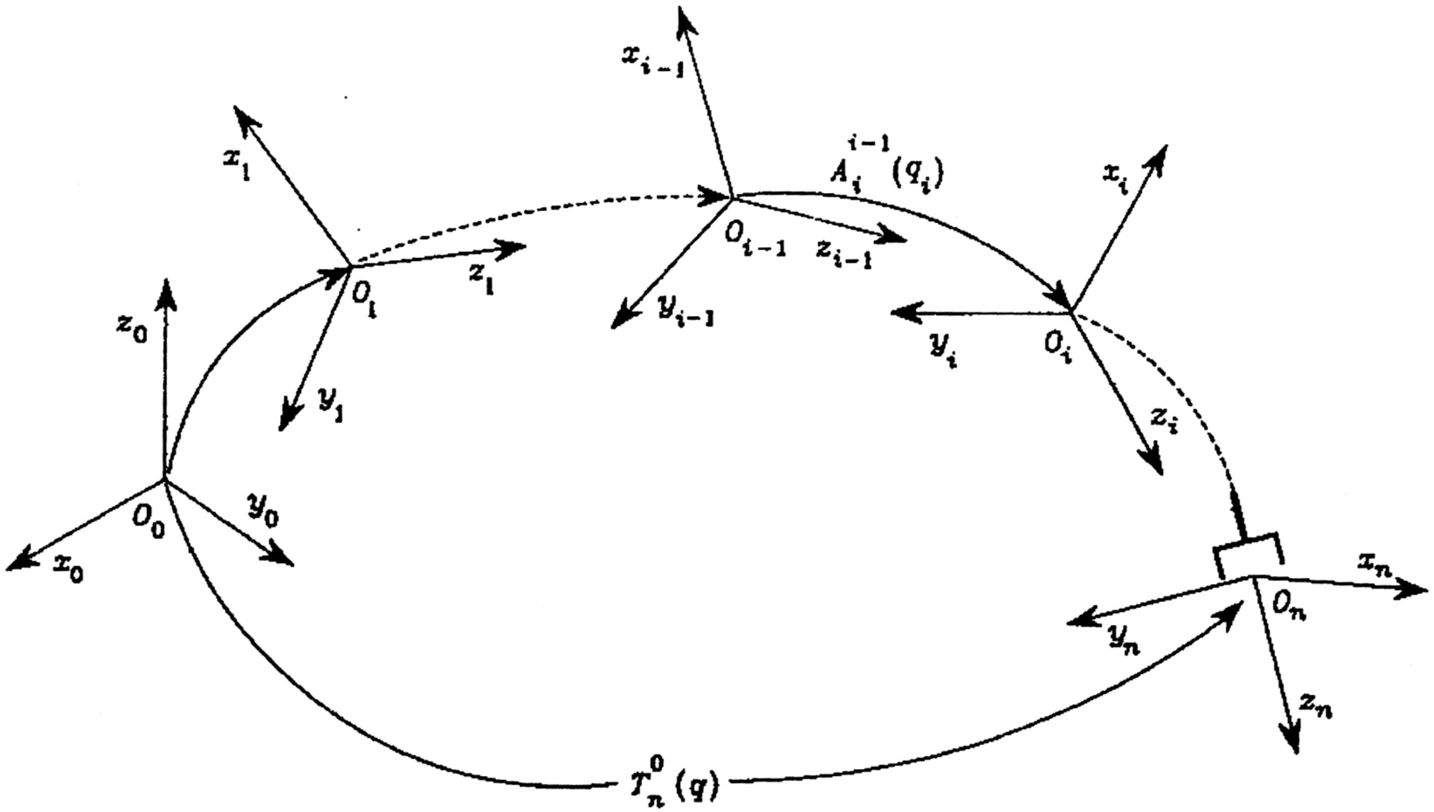
$$\mathbf{A}_{i'}^{i-1} = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}_{i}^{i'} = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}_{i}^{i-1}(q_i) = \mathbf{A}_{i'}^{i-1} \cdot \mathbf{A}_{i}^{i'} = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

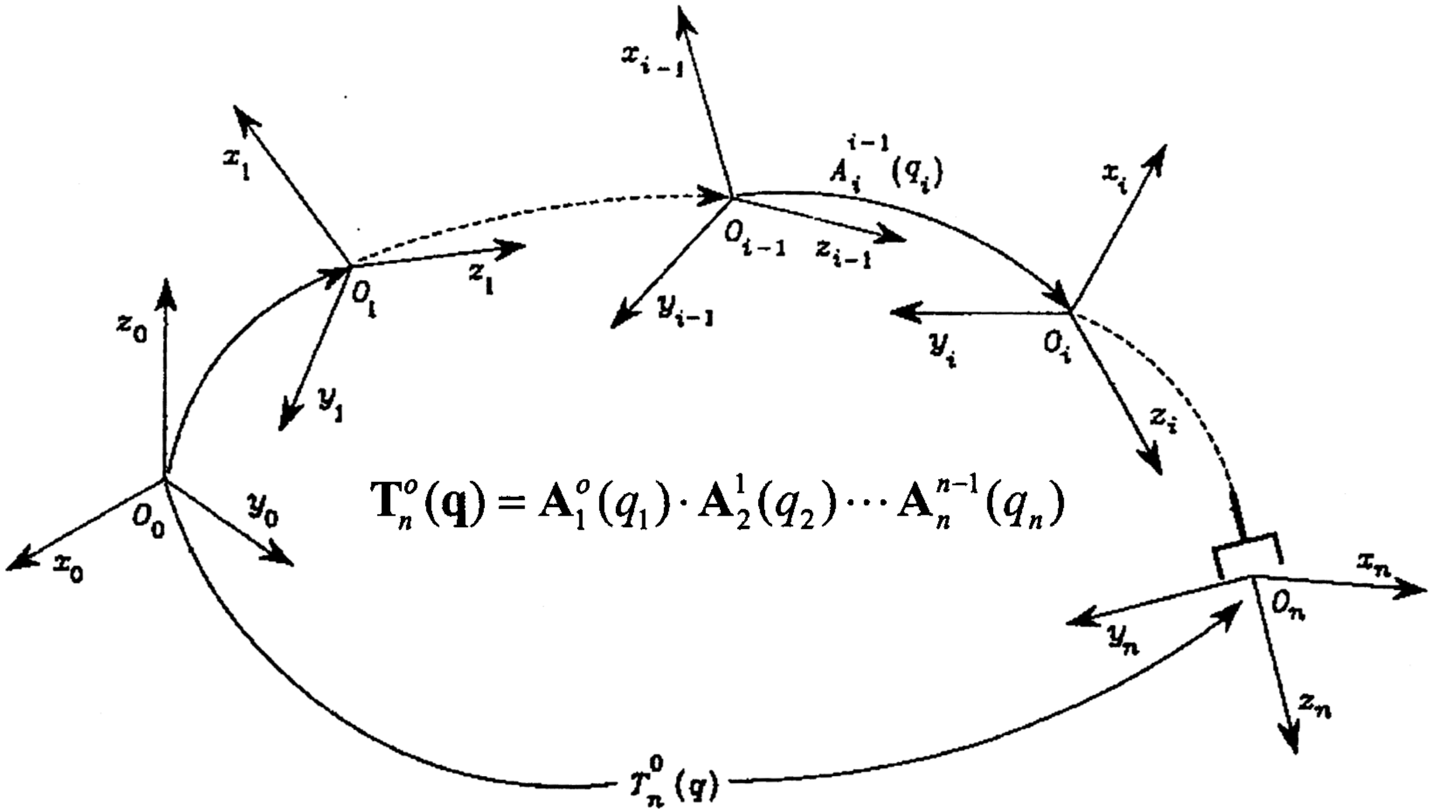
Calculation of the geometrical robot model

1. Set the coordinate frames for all segments
2. Define the table of DH parameters a_i, d_i, α_i in θ_i for all segments $i=1,2,\dots,n$
3. Calculate DH transformations $\mathbf{A}_i^{i-1}(q_i)$ for $i=1,2,\dots,n$
4. Calculate the geometrical model: $\mathbf{T}_n^o(\mathbf{q}) = \mathbf{A}_1^o(q_1)\cdot\mathbf{A}_2^1(q_2)\cdots\mathbf{A}_n^{n-1}(q_n)$



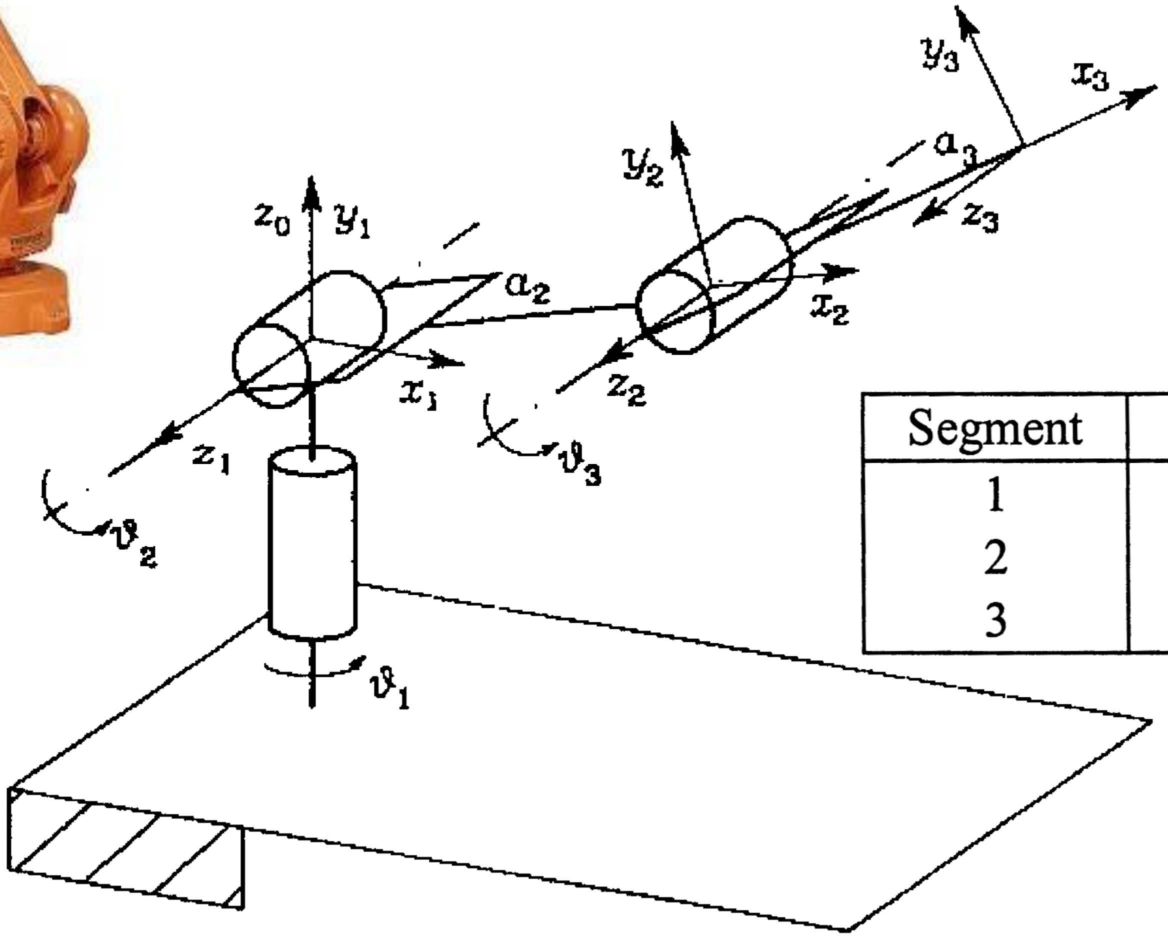
Using geometrical model

- Geometrical robot model gives the pose of the last segment of the robot (end-effector) expressed in the reference (base) frame
- Geometrical robot model defines the pose (position and orientation) of the end-effector depending on the current internal coordinates **q**



$$\mathbf{T}_n^o(\mathbf{q}) = \mathbf{A}_1^o(q_1)\cdot\mathbf{A}_2^1(q_2)\cdots\mathbf{A}_n^{n-1}(q_n)$$

Anthropomorphic robot manipulator



Segment	a_i	α_i	d_i	θ_i
1	0	$\pi/2$	0	θ_1
2	a_2	0	0	θ_2
3	a_3	0	0	θ_3

$$\mathbf{q} = [\theta_1, \theta_2, \theta_3]^T$$

Anthropomorphic robot manipulator

Segment	a_i	α_i	d_i	θ_i
1	0	$\pi/2$	0	θ_1
2	a_2	0	0	θ_2
3	a_3	0	0	θ_3

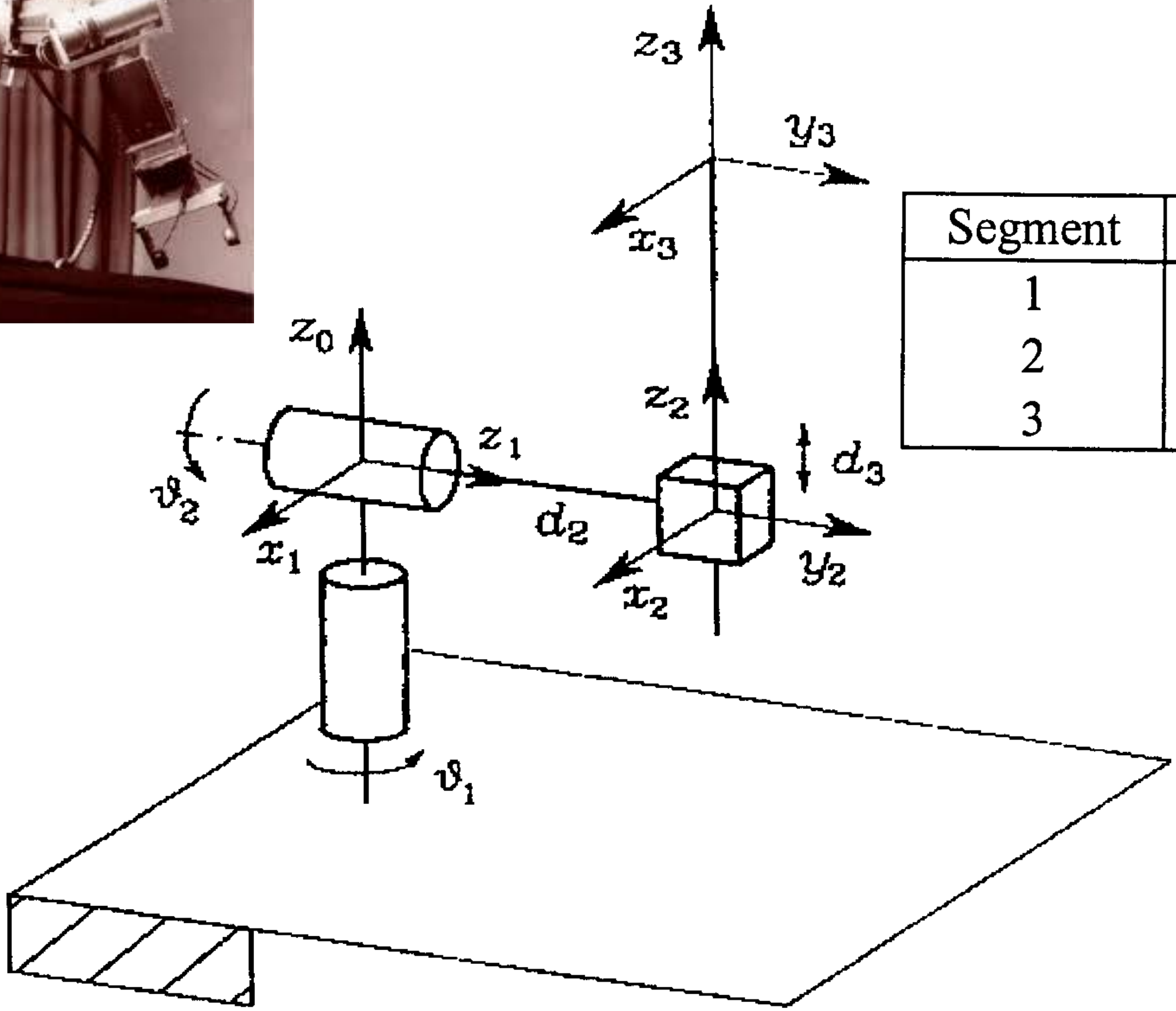
$$\mathbf{A}_i^{i-1}(q_i) = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}_1^o(\theta_1) = \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}_i^{i-1}(\theta_i) = \begin{bmatrix} c_i & -s_i & 0 & a_i c_i \\ s_i & c_i & 0 & a_i s_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad i=2,3$$

$$\mathbf{T}_3^o(\mathbf{q}) = \mathbf{A}_1^o \cdot \mathbf{A}_2^1 \cdot \mathbf{A}_3^2 = \begin{bmatrix} c_1 c_{23} & -c_1 s_{23} & s_1 & c_1(a_2 c_2 + a_3 c_{23}) \\ s_1 c_{23} & -s_1 s_{23} & -c_1 & s_1(a_2 c_2 + a_3 c_{23}) \\ s_{23} & c_{23} & 0 & a_2 s_2 + a_3 s_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Stanford robot manipulator



Segment	a_i	α_i	d_i	θ_i
1	0	$-\pi/2$	0	θ_1
2	0	$\pi/2$	d_2	θ_2
3	0	0	d_3	0

$$\mathbf{q} = [\theta_1, \theta_2, d_3]^T$$

Stanford robot manipulator

Segment	a_i	α_i	d_i	θ_i
1	0	$-\pi/2$	0	θ_1
2	0	$\pi/2$	d_2	θ_2
3	0	0	d_3	0

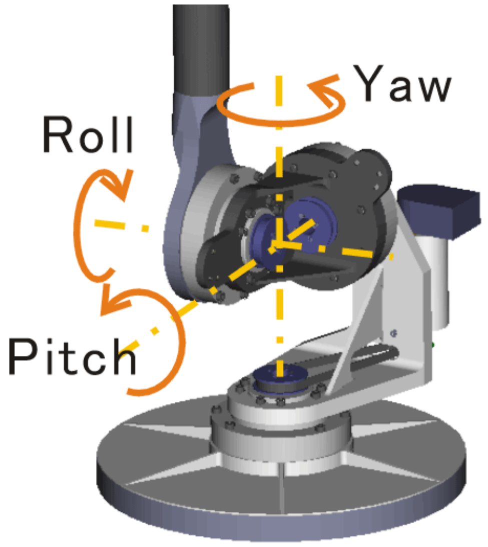
$$\mathbf{q} = [\theta_1, \theta_2, d_3]^T$$

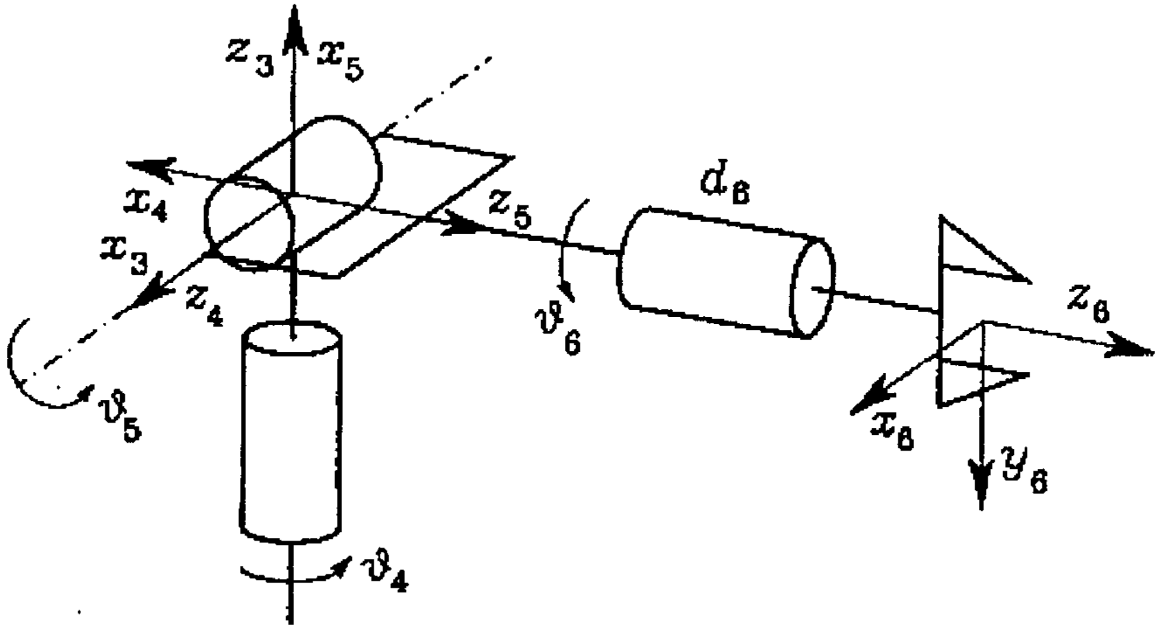
$$\mathbf{A}_1^o(\theta_1) = \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{A}_2^1(\theta_1) = \begin{bmatrix} c_2 & 0 & s_2 & 0 \\ s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{A}_3^2(d_3) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}_3^o(\underline{q}) = \mathbf{A}_1^o \cdot \mathbf{A}_2^1 \cdot \mathbf{A}_3^2 = \begin{bmatrix} c_1 c_2 & -s_1 & c_1 s_2 & c_1 s_2 d_3 - s_1 d_2 \\ s_1 c_2 & c_1 & s_1 s_2 & s_1 s_2 d_3 + c_1 d_2 \\ -s_2 & 0 & c_2 & c_2 d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Spherical robot wrist

- Usually attached to the end of the robot arm





- All three axes of the rotational joints intersect in the same point

$$\mathbf{q} = [\theta_4, \theta_5, \theta_6]^T$$

Segment	a_i	α_i	d_i	θ_i
4	0	$-\pi/2$	0	θ_4
5	0	$\pi/2$	0	θ_5
6	0	0	d_6	θ_6

Spherical robot wrist

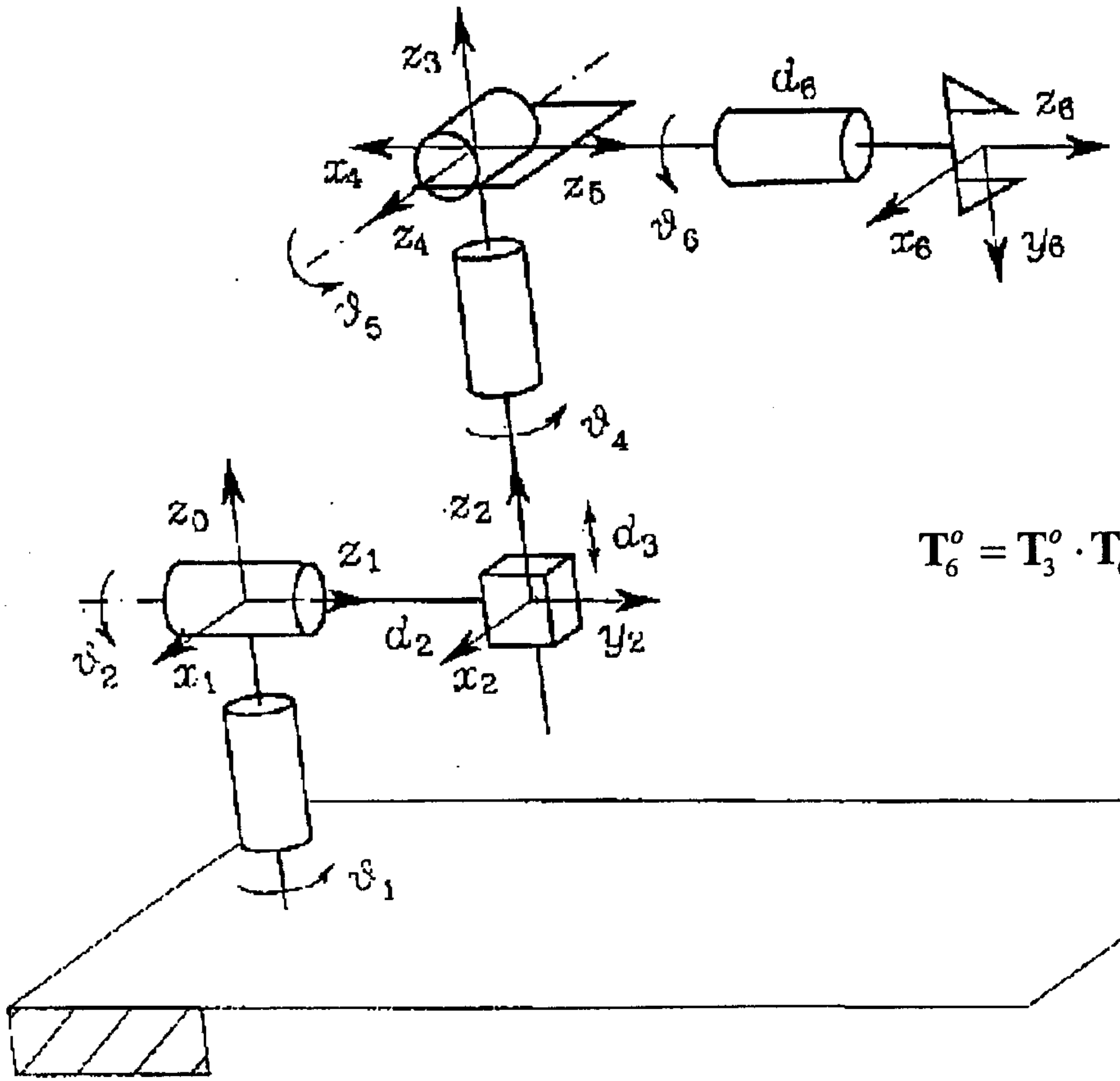
Segment	a_i	α_i	d_i	θ_i
4	0	$-\pi/2$	0	θ_4
5	0	$\pi/2$	0	θ_5
6	0	0	d_6	θ_6

$$\mathbf{q} = [\theta_4, \theta_5, \theta_6]^T$$

$$\mathbf{A}_4^3(\theta_4) = \begin{bmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{A}_5^4(\theta_5) = \begin{bmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{A}_6^5(\theta_6) = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}_6^3(\underline{q}) = \mathbf{A}_4^3 \cdot \mathbf{A}_5^4 \cdot \mathbf{A}_6^5 = \begin{bmatrix} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - s_4c_6 & c_4s_5 & c_4s_5d_6 \\ s_4c_5c_6 + c_4s_6 & -s_4c_5s_6 + c_4c_6 & s_4s_5 & s_4s_5d_6 \\ -s_5c_6 & s_5s_6 & c_5 & c_5d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Stanford manipulator with the wrist



$$\mathbf{T}_6^o = \mathbf{T}_3^o \cdot \mathbf{T}_6^3 = \begin{bmatrix} \mathbf{n}^o & \mathbf{s}^o & \mathbf{a}^o & \mathbf{p}^o \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Stanford manipulator with the wrist

$$\mathbf{p}^o = \begin{bmatrix} c_1 s_2 d_3 - s_1 d_2 + \left(c_1\left(c_2 c_4 s_5 + s_2 c_5\right) - s_1 s_4 s_5\right) d_6 \\ s_1 s_2 d_3 + c_1 d_2 + \left(s_1\left(c_2 c_4 s_5 + s_2 c_5\right) + c_1 s_4 s_5\right) d_6 \\ c_2 d_3 + \left(-s_2 c_4 s_5 - c_2 c_5\right) d_6 \end{bmatrix}$$

$$\mathbf{n}^o = \begin{bmatrix} c_1\left(c_2\left(c_4 c_5 c_6 - s_4 s_6\right) - s_2 s_5 c_6\right) - s_1\left(s_4 c_5 c_6 + c_4 s_6\right) \\ s_1\left(c_2\left(c_4 c_5 c_6 - s_4 s_6\right) - s_2 s_5 c_6\right) + c_1\left(s_4 c_5 c_6 + c_4 s_6\right) \\ -s_2\left(c_4 c_5 c_6 - s_4 s_6\right) - c_2 s_5 s_6 \end{bmatrix}$$

$$\mathbf{s}^o = \begin{bmatrix} c_1\left(-c_2\left(c_4 c_5 s_6 + s_4 c_6\right) + s_2 s_5 s_6\right) - s_1\left(-s_4 c_5 c_6 + c_4 c_6\right) \\ s_1\left(-c_2\left(c_4 c_5 s_6 + s_4 c_6\right) + s_2 s_5 s_6\right) + c_1\left(-s_4 c_5 c_6 + c_4 c_6\right) \\ s_2\left(c_4 c_5 s_6 + s_4 c_6\right) + c_2 s_5 s_6 \end{bmatrix}$$

$$\mathbf{a}^o = \begin{bmatrix} c_1\left(c_2 c_4 s_5 + s_2 c_5\right) - s_1 s_4 s_5 \\ s_1\left(c_2 c_4 s_5 + s_2 c_5\right) + c_1 s_4 s_5 \\ -s_2 c_4 s_5 + c_2 c_5 \end{bmatrix}$$

Inverse kinematics model

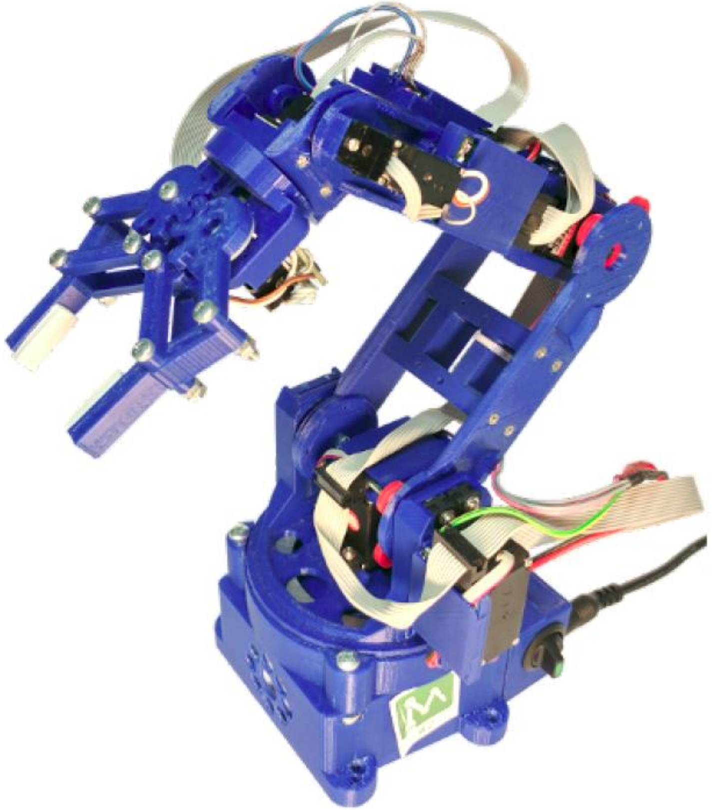
- Direct kinematics defines the pose of the end-effector depending on the internal coordinates
  - Where will the end-effector move
  - The pose of the end-effector is uniquely determined

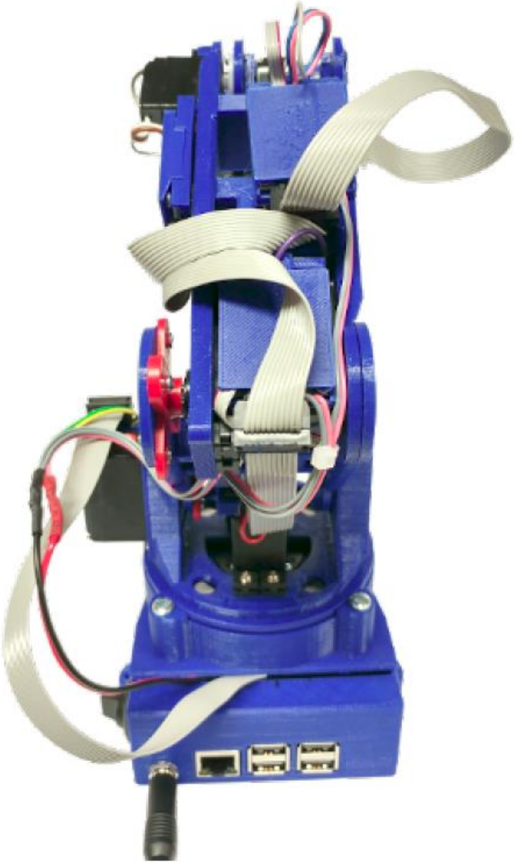
$$\mathbf{T}(\mathbf{q})$$

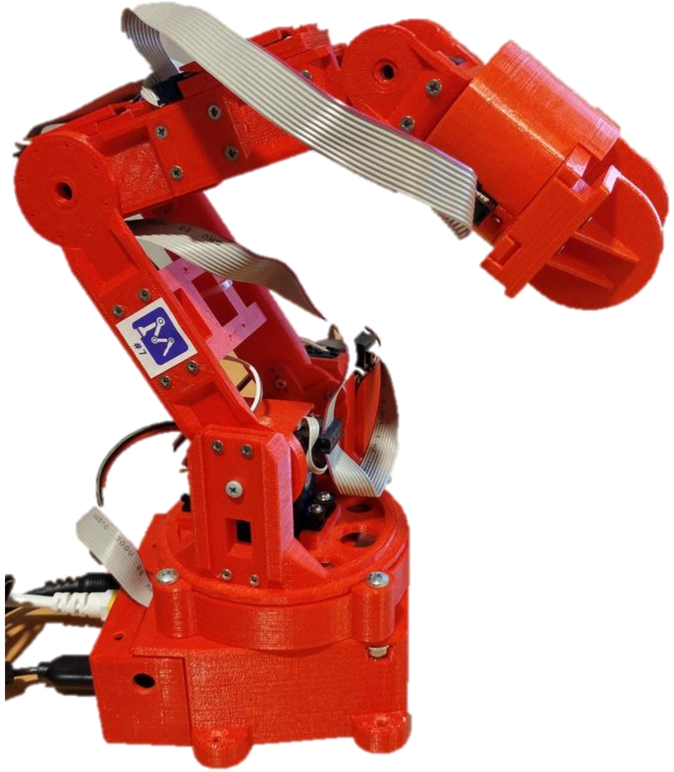
- Inverse kinematics defines the internal coordinates that would bring the robot end-effector in the desired pose
  - How to move the end-effector to reach the desired pose
  - Challenging problem:
    - Nonlinear equations
    - The solution is not uniquely defined
      - Several solutions
      - Sometimes even infinite number of solutions
      - Sometimes the solution does not exist
  - Take into account several criteria that determine which solution is optimal
  - Sometimes we can get analytical solution, sometimes only numerical are possible

$$\mathbf{q}(\mathbf{T})$$

ViCoS LCLWOS robot manipulator







Requirements

Low production cost
The cost of a single unit should be below 300€, it should use widely available components where possible.

Easy construction and maintenance
Construction from parts should be simpe, parts should be easily replaceable.

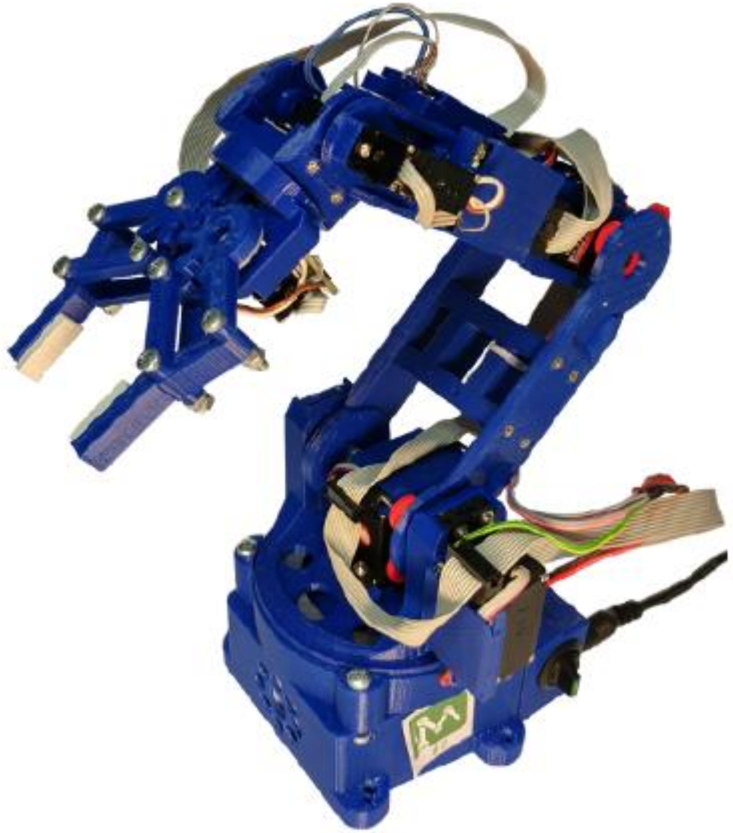
Simplicity and ease of use
The interaction with the platform should be multi-level

Robustness
Both the hardware and software should be robust enough to withstand long-term use.

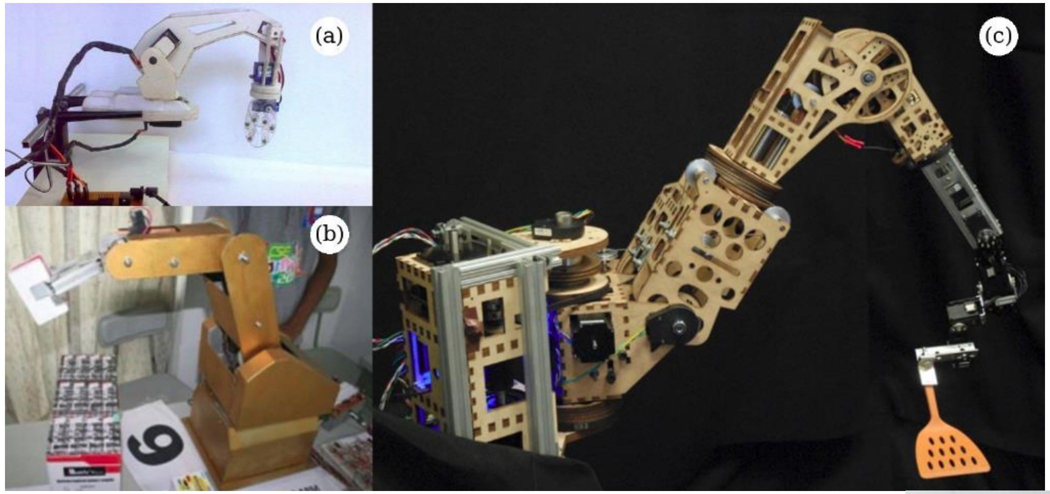
Safety
The manipulator should be safe enough to be operated even by kids.

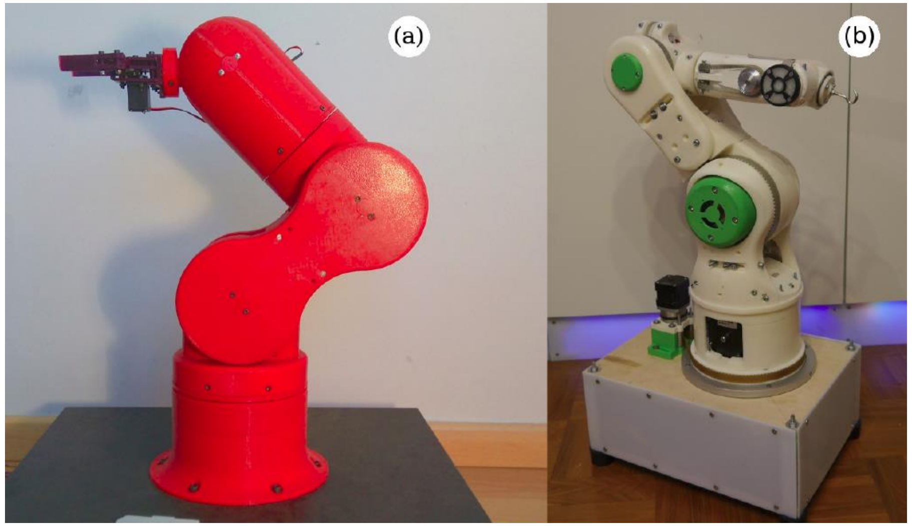
Realism
The experience should be real enough for students of different university-level courses on robotics.

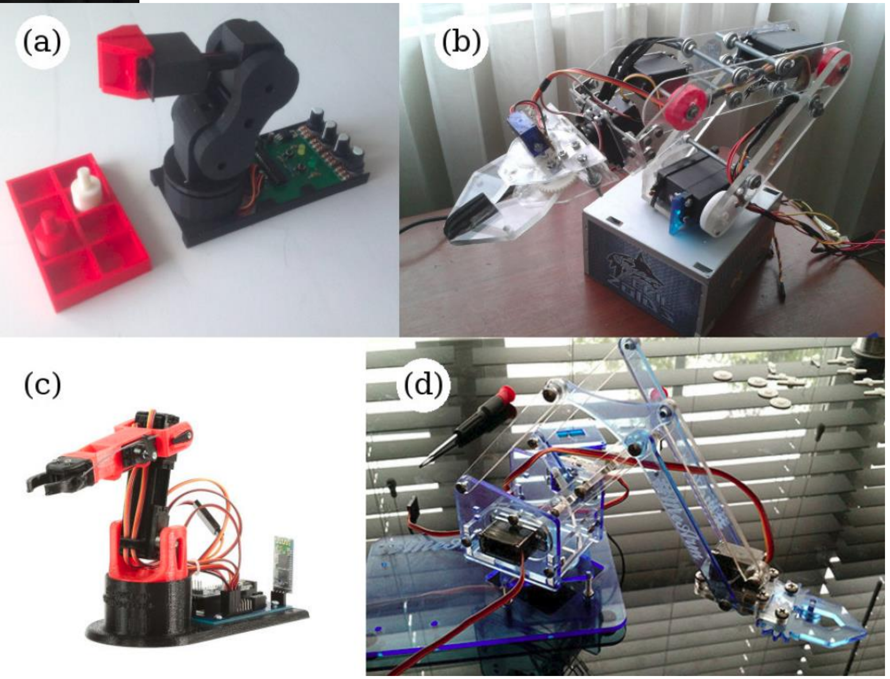
Openness
Bot the hardware designs and software sources should be available for others to



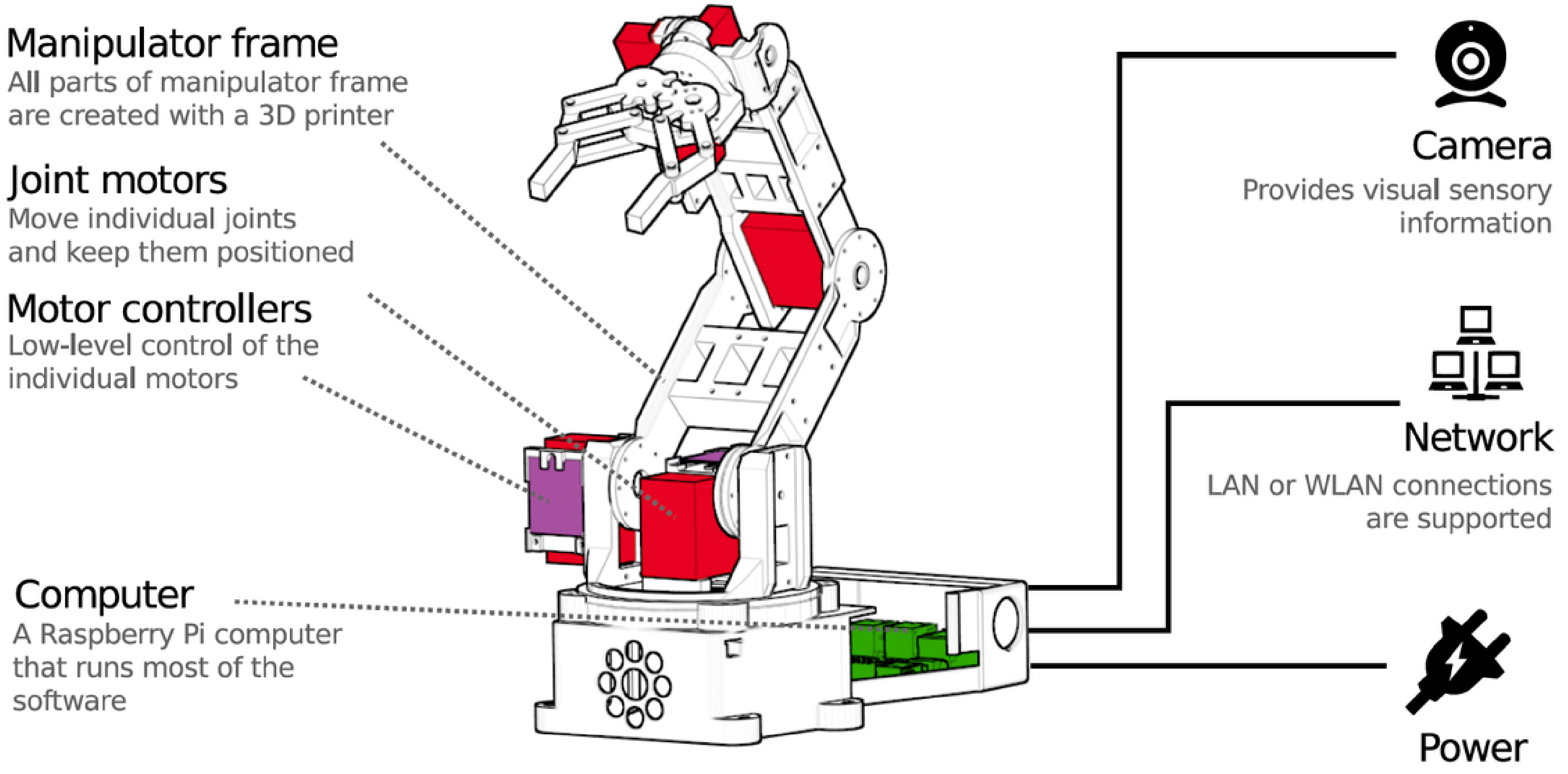
Related work



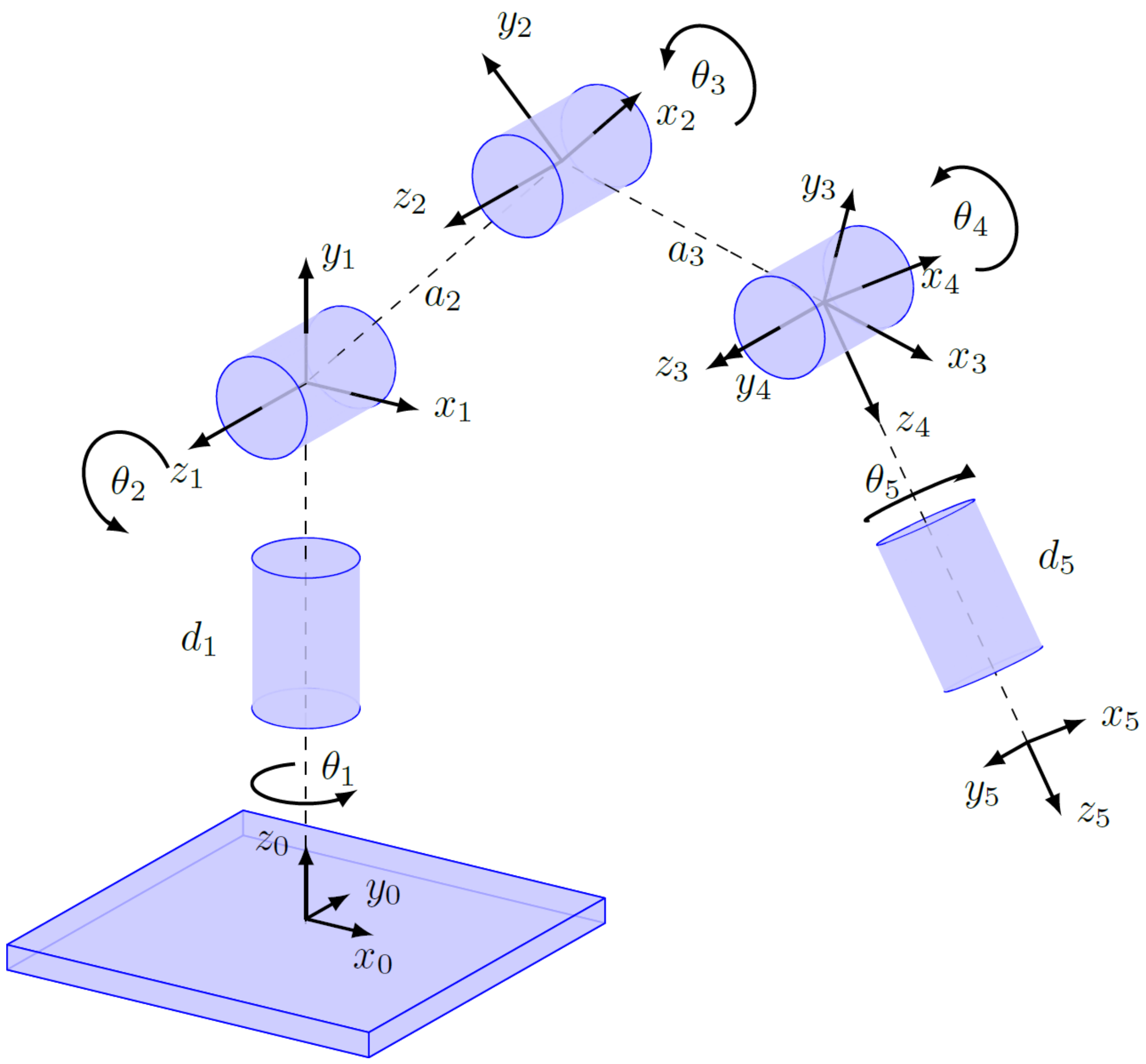




Robot manipulator



Forward model



DH parameters:

Segment	a_i	d_i	α_i	θ_i
1	0 mm	48 mm	90°	θ_1
2	108 mm	0 mm	0°	θ_2
3	112 mm	0 mm	0°	θ_3
4	0 mm	0 mm	90°	θ_4
5	0 mm	90 mm	0°	θ_5

Transformation

$$A_i^{i-1} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_1^0 = \begin{bmatrix} \cos\theta_1 & 0 & \sin\theta_1 & 0 \\ \sin\theta_1 & 0 & -\cos\theta_i & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2^1 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & a_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & a_2\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3^2 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & a_3\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & a_3\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

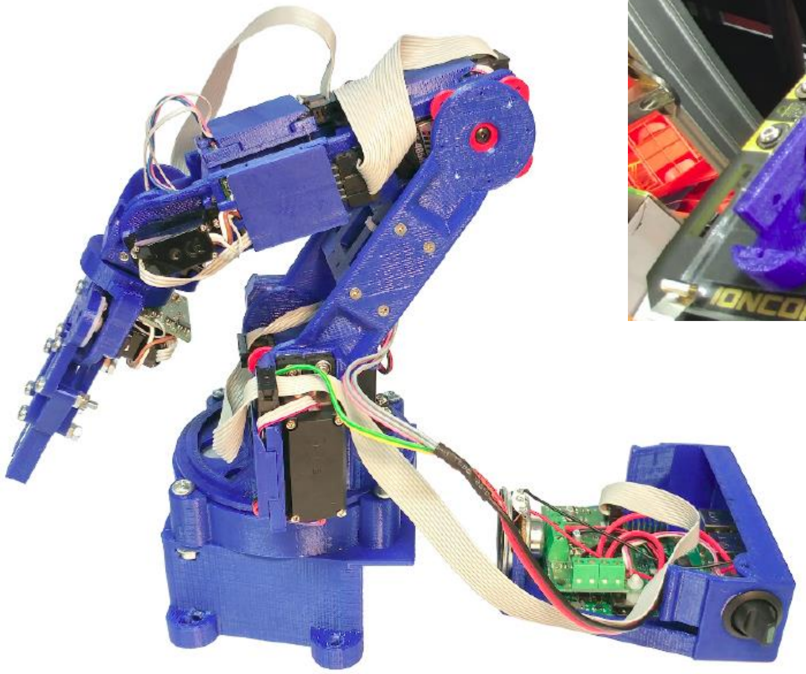
$$A_4^3 = \begin{bmatrix} \cos\theta_4 & 0 & \sin\theta_4 & 0 \\ \sin\theta_4 & 0 & -\cos\theta_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

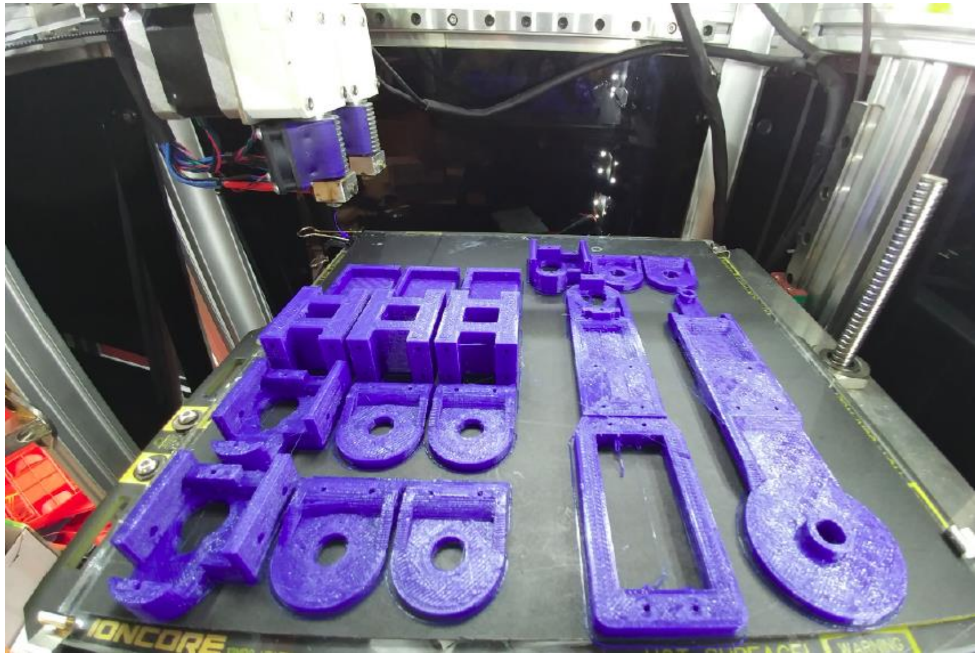
$$A_5^4 = \begin{bmatrix} \cos\theta_5 & -\sin\theta_5 & 0 & 0 \\ \sin\theta_5 & \cos\theta_5 & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_5^0 = A_1^0 A_2^1 A_3^2 A_4^3 A_5^4$$

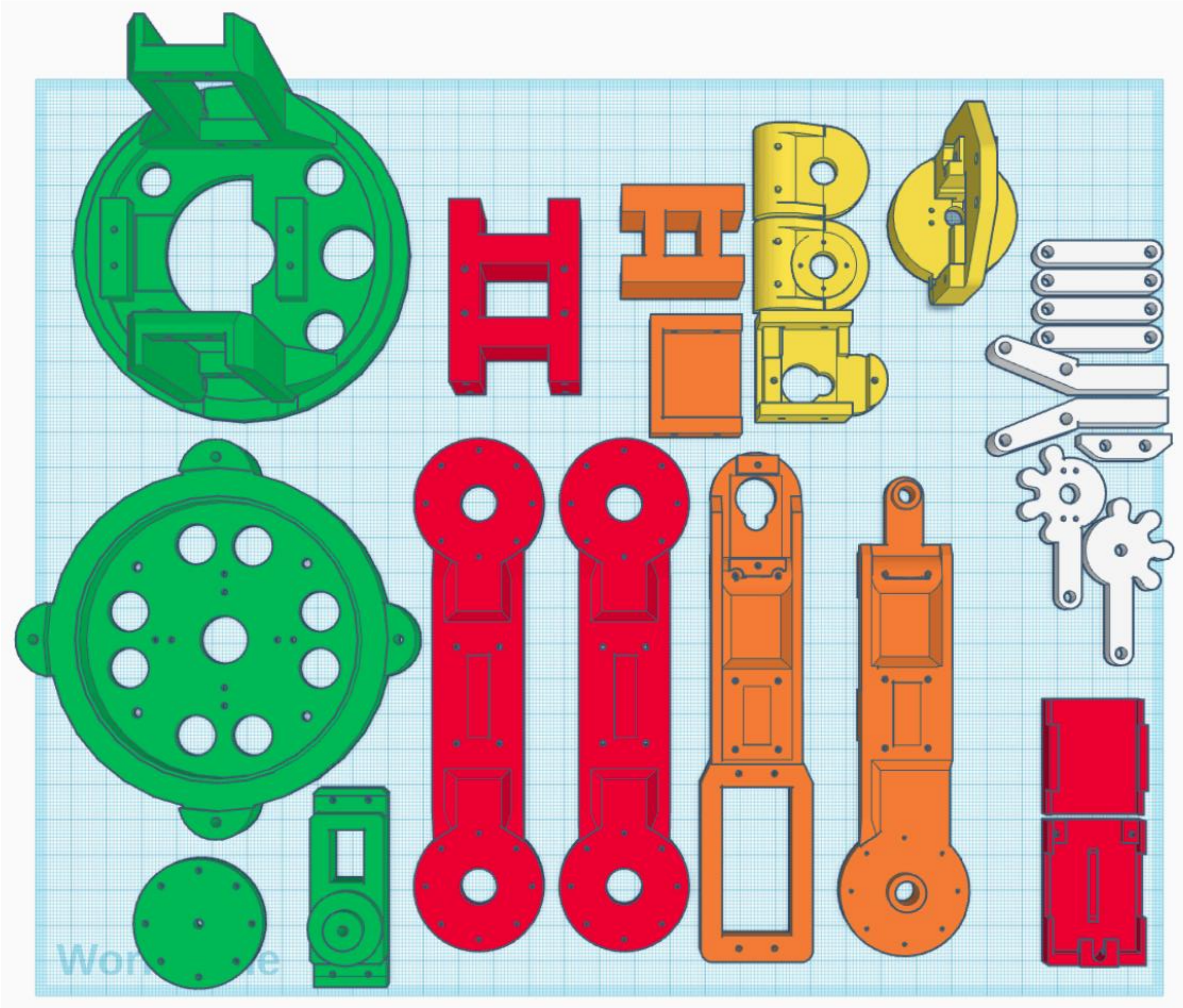
Frame

- 3D printed



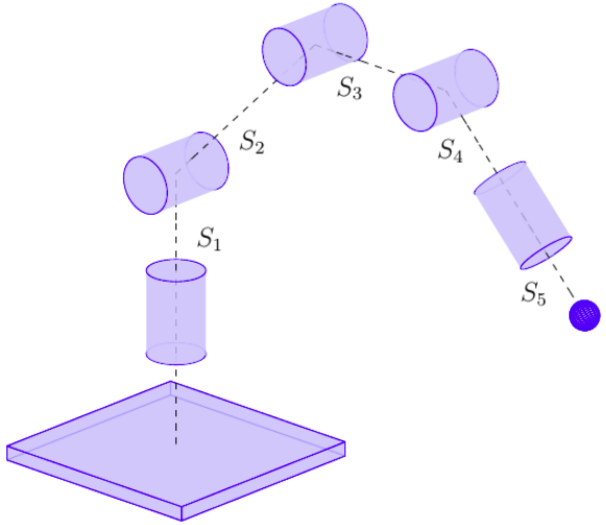


Frame parts



Motors

- Servo motors



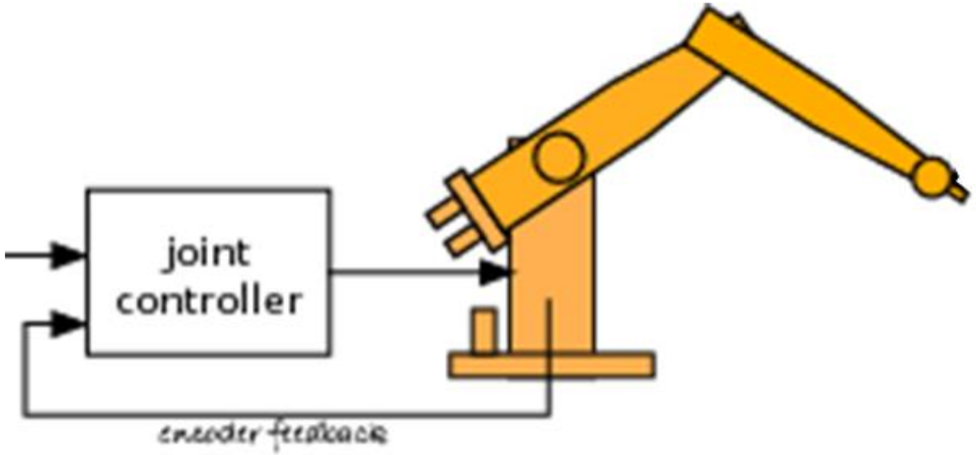
Sklep	Servomotor
S1	HS-311
S2	HS-645MG
S3	HS-485HB
S4	ES08AII
S5	ES08AII
Prijemalo	ES08AII

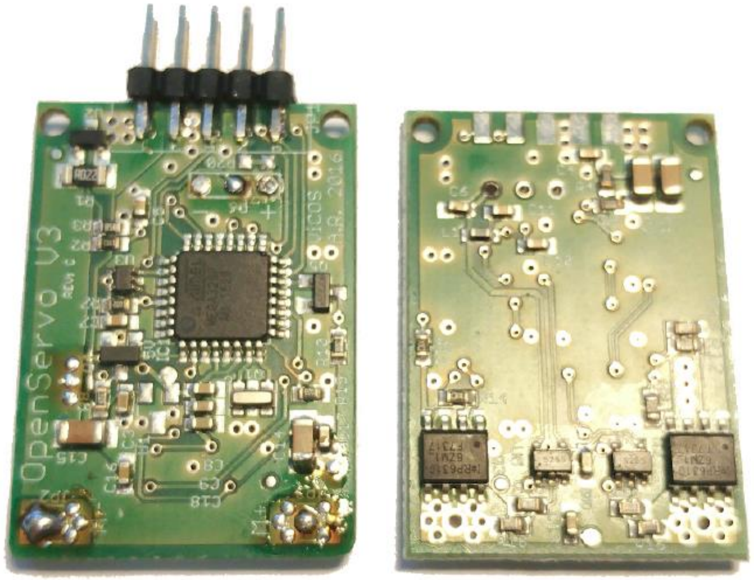
Motors

Servo-motor	Nape-tost	Rotacijska hitrost	Navor	Kratko-stični tok	Teža	Material zobnikov
HS-645MG	$4,8\ V$	$0,24\ s/60°$	$7,7\ kg/cm$	$2,5\ A$	$55,2\ g$	kovina
	$6,0\ V$	$0,20\ s/60°$	$9,6\ kg/cm$			
HS-485HB	$4,8\ V$	$0,22\ s/60°$	$4,8\ kg/cm$	$1,2\ A$	$45,0\ g$	karbonit
	$6,0\ V$	$0,20\ s/60°$	$6,0\ kg/cm$			
HS-311	$4,8\ V$	$0,19\ s/60°$	$3,0\ kg/cm$	$0,8\ A$	$43,0\ g$	najlon
	$6,0\ V$	$0,15\ s/60°$	$3,5\ kg/cm$			
ES08AII	$4,8\ V$	$0,12\ s/60°$	$1,5\ kg/cm$	$0,7\ A$	$8,5\ g$	najlon
	$6,0\ V$	$0,10\ s/60°$	$1,8\ kg/cm$			

Upgrading servomotors

- New control circuit
- Potentiometer
- OpenServo
- Current protection





PID controller

- Proportional, integral and derivative part

$$u(t_k) = K_p e(t_k) + K_i \sum_{n=1}^{k} e(t_n)\Delta t + \frac{e(t_k) - e(t_{k-1})}{\Delta t}$$

```
// Proporcialni del
p_component = seek_position − current_position;
pwm_output = (int32_t)p_component ∗ (int32_t)p_gain;
// Odvodni del
d_component = (p_component − p_component_old) ∗ 256;
pwm_output+ = (int32_t)d_component ∗ (int32_t)d_gain;
// Integralni del
i_component+ = p_component;
pwm_output+ = (((int32_t)i_component ∗ (int32_t)i_gain) >> 8);
//
pwm = ((int16_t)(pwm_output >> 8));
```

Sklep	K_p	K_i	K_d
S1	500	300	5
S2	500	300	5
S3	500	300	5
S4	300	200	5
S5	150	200	5
Prijemalo	500	200	5

AD converter

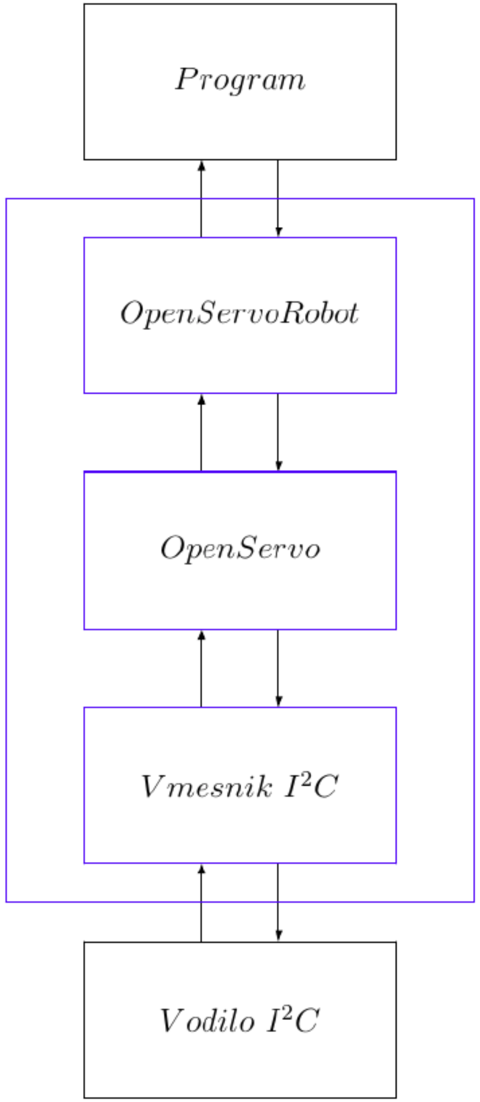
- Increasing resolution
- Multiple sampling and decimation
- Resolution increased from 10 to 12 bits
- Sampling frequency 256 Hz

$$ADC_{(10+n)bit} = \frac{\sum_{k=1}^{4^n} ADC_{10bit}}{2^n}$$

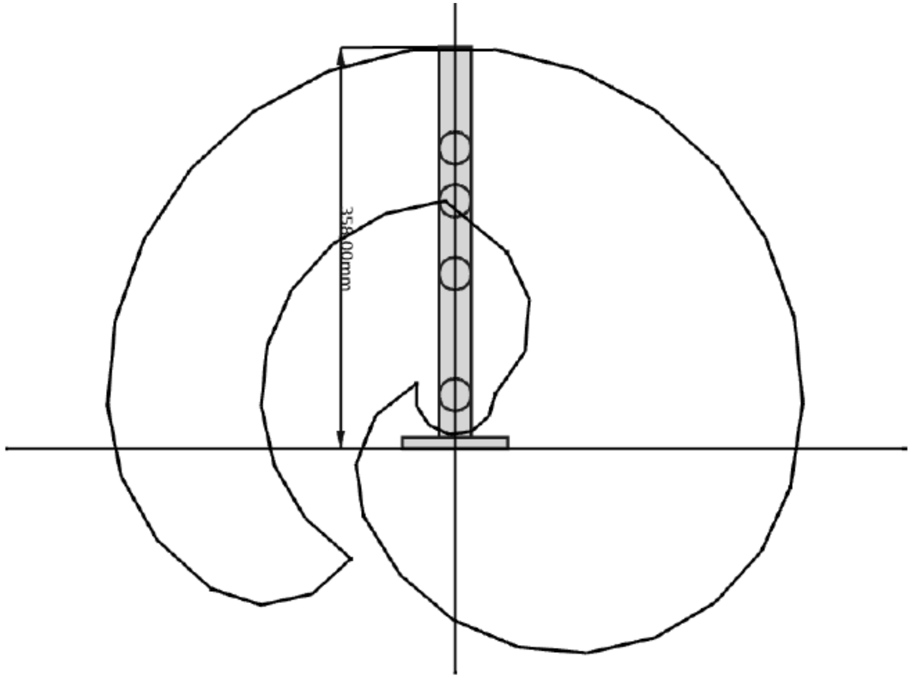
Ločljivost	Čas branja	Največja frekvenca
10 bitov	$\approx 0,0864\ ms$	$\approx 11574\ Hz$
11 bitov	$\approx 0,3456\ ms$	$\approx 2893\ Hz$
12 bitov	$\approx 1,3824\ ms$	$\approx 723\ Hz$
13 bitov	$\approx 5,5296\ ms$	$\approx 180\ Hz$

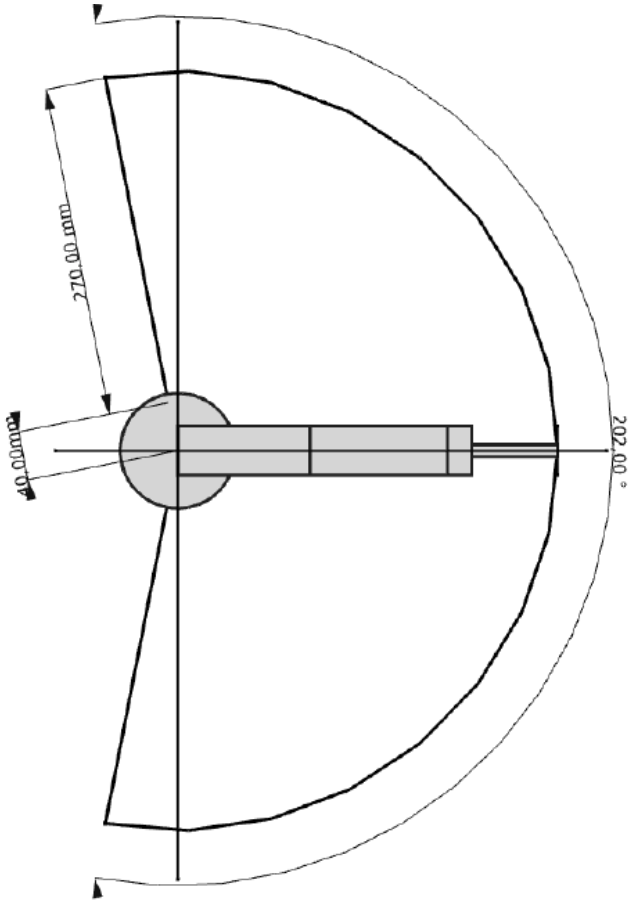
Communication

- I^2C interface
  - I^2C bus
- OpenServo
  - Communication with microcontroller
- OpenServoRobot
  - Robot model (DH parameters)
  - Communication with application



Workspace



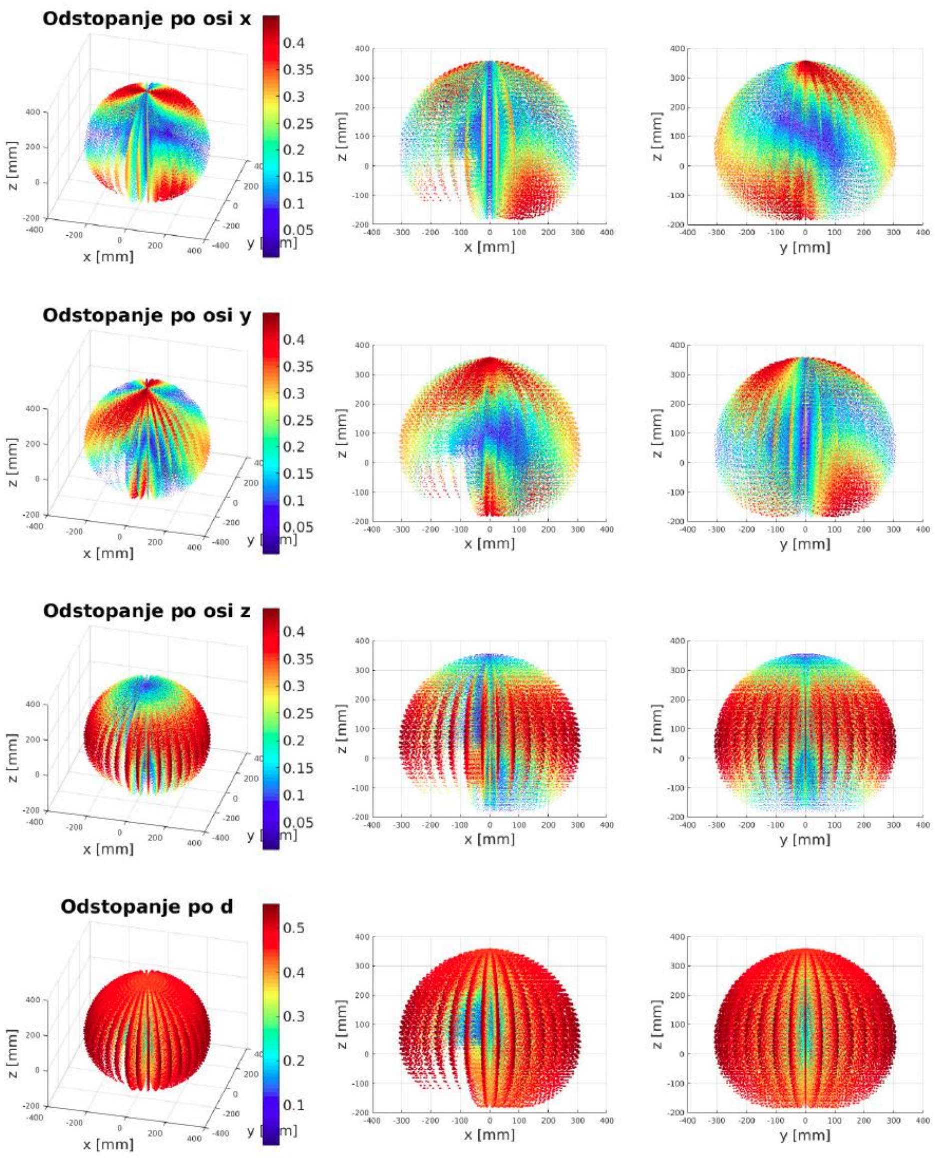


Theoretical accuracy

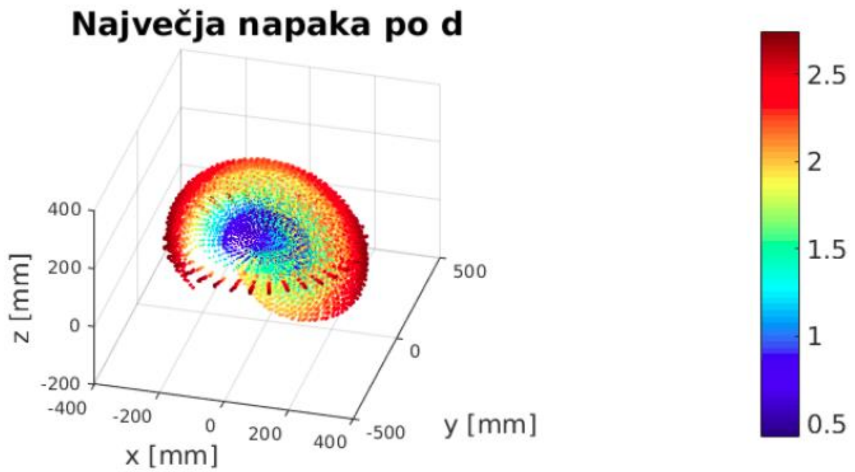
- Expected deviation of the estimated position from the reference position of the end effector
- Only considering motor errors
- Theoretical upper limit of precision

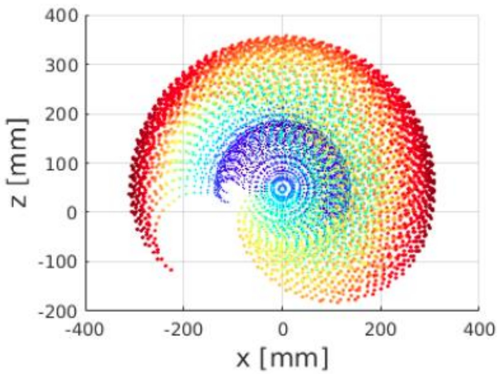
Osi	Odstopanje (mm)			Standardni odklon (mm)		
	Najmanj	Največ	Povprečno	Najmanj	Največ	Povprečno
x	0,00	0,45	0,14	0,01	0,61	0,20
y	0,00	0,45	0,14	0,00	0,61	0,21
z	0,00	0,44	0,16	0,03	0,56	0,66
d	0,10	0,55	0,29	0,14	0,73	0,41

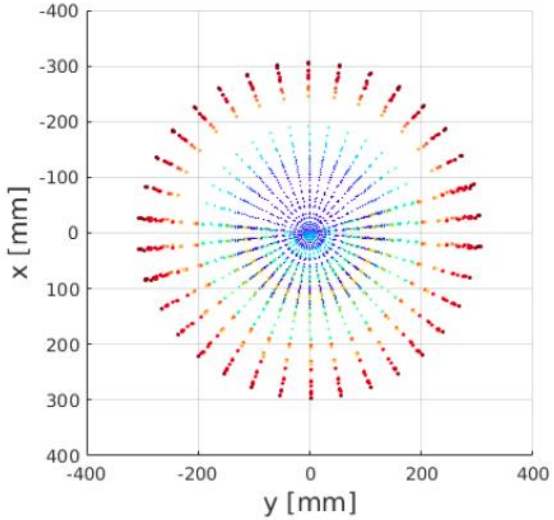
Theoretical accuracy



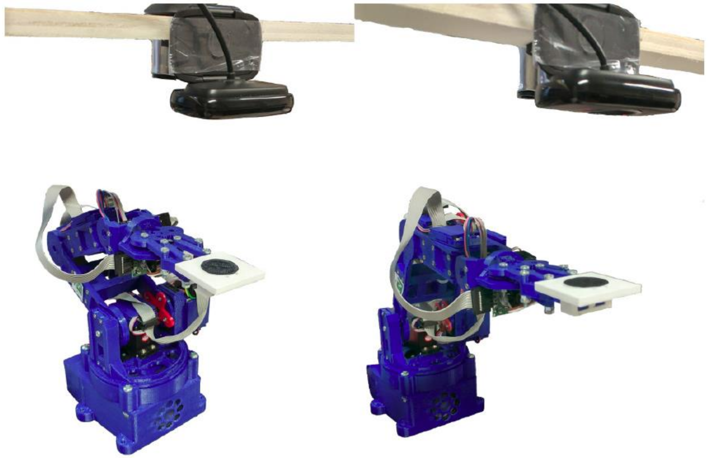
Theoretical accuracy



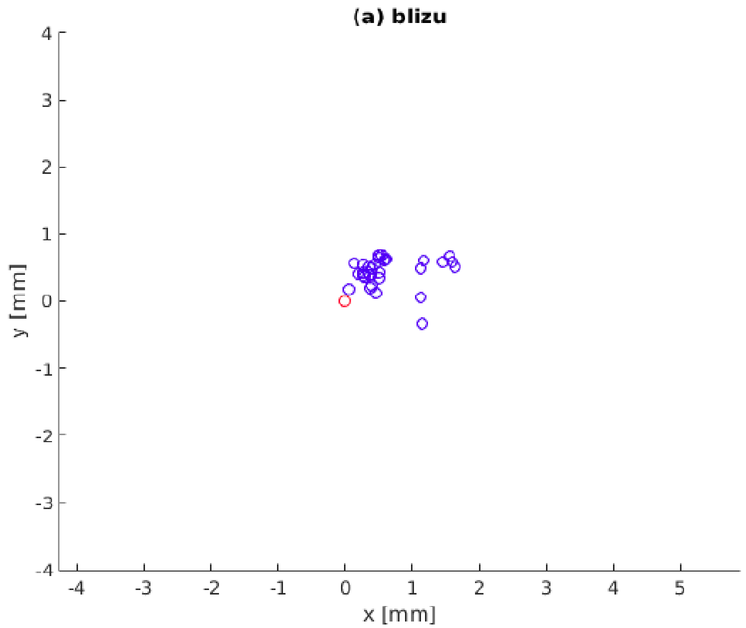


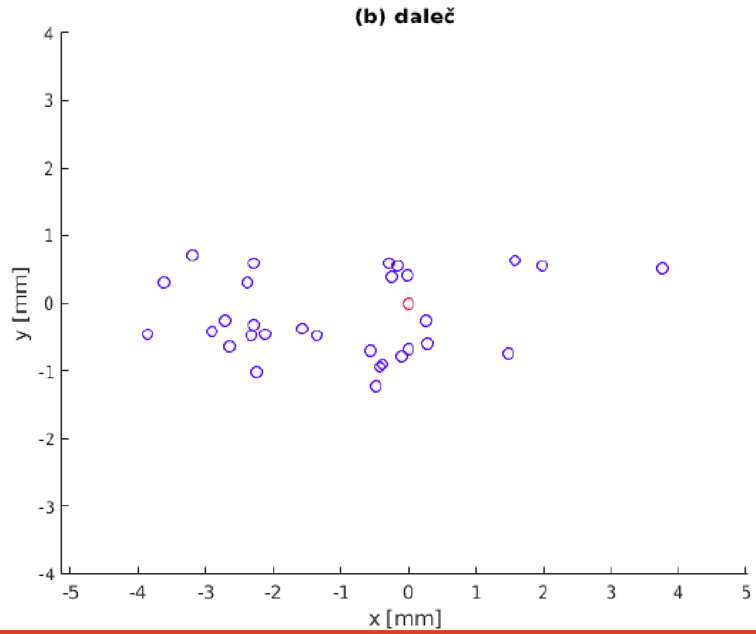


Empiciral repetabilty

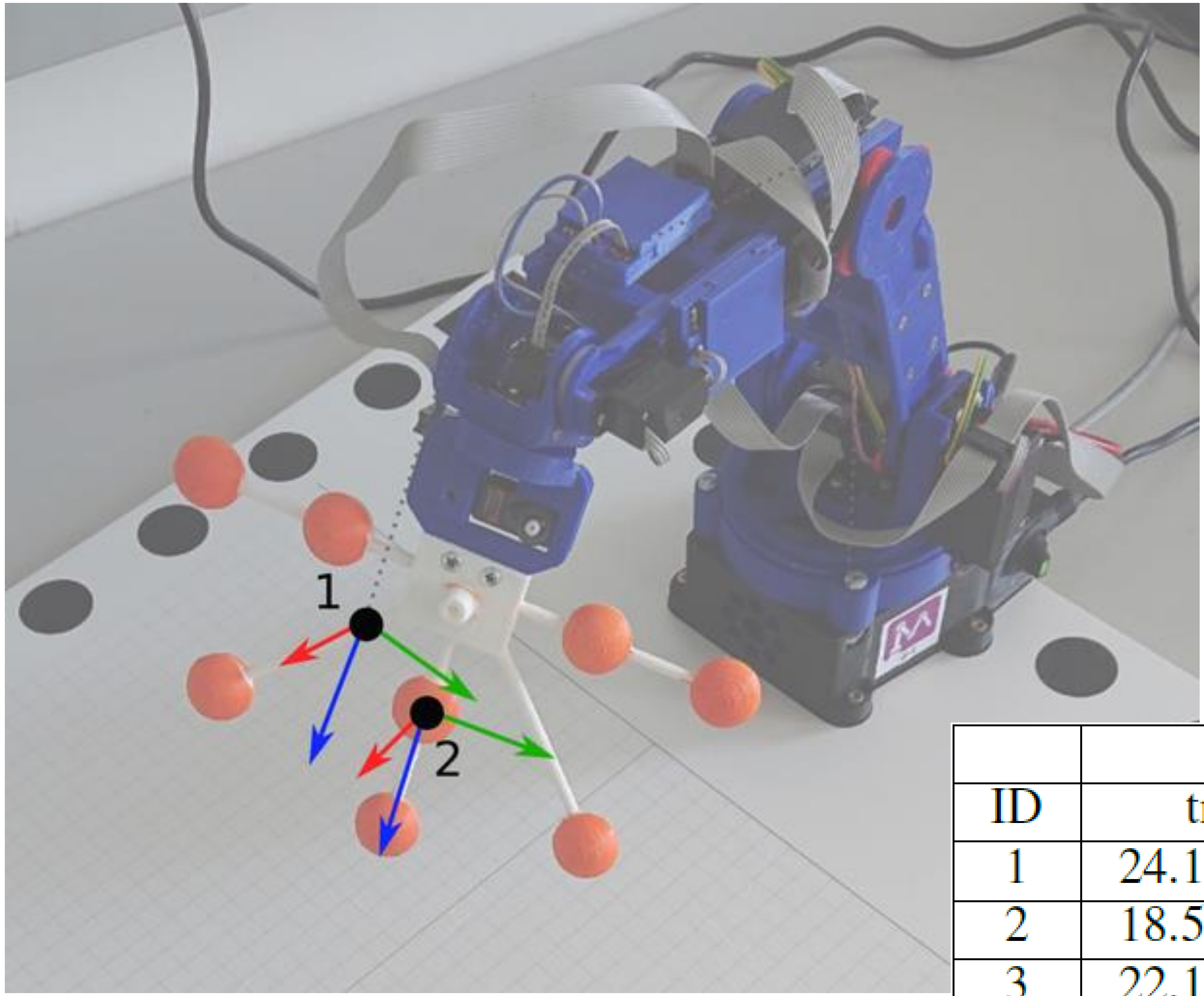


Meritev	Os x	Os y
Blizu	1,4 mm	0,6 mm
Daleč	3,8 mm	0,7 mm





Calibration



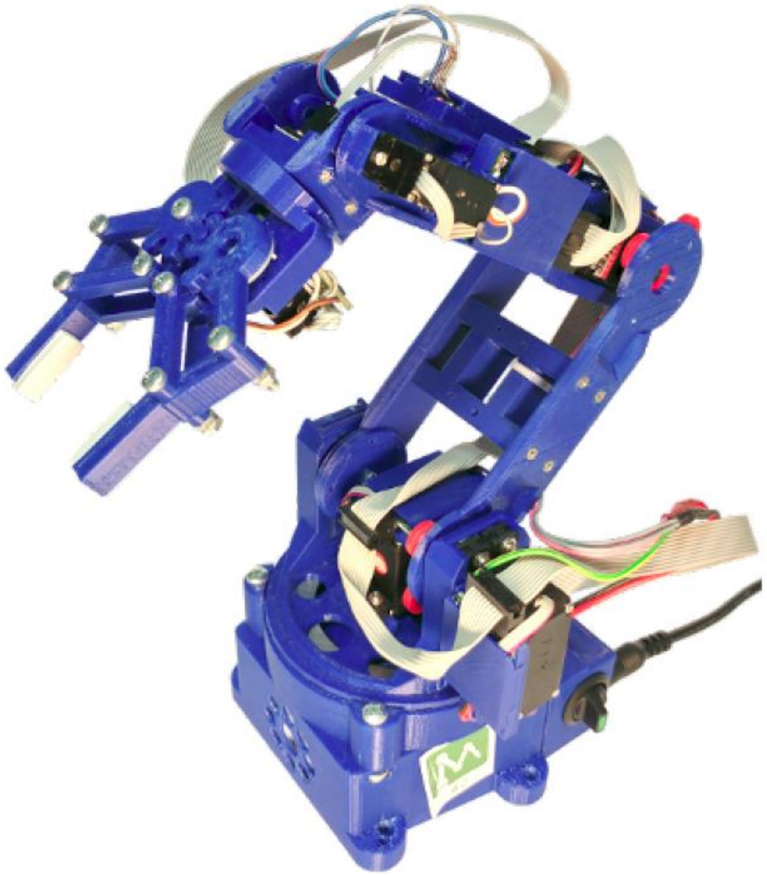
Grasping a cube (100 times):

ID	Uncalibrated	Calibrated
1	69	99
2	97	100
3	18	88
4	27	99

Mean error:

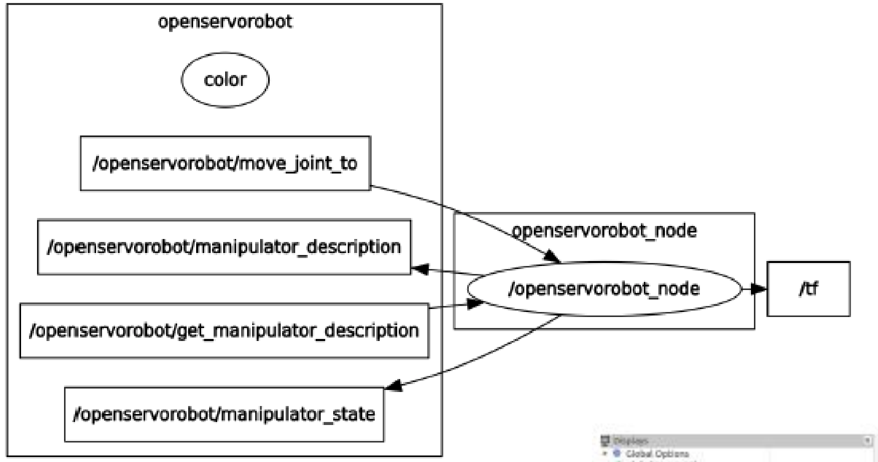
	Uncalibrated		Calibrated	
ID	train	test	train	test
1	24.1 ± 5.1	23.0 ± 4.9	4.3 ± 3.7	5.3 ± 3.9
2	18.5 ± 3.7	20.5 ± 4.9	3.5 ± 2.8	4.7 ± 3.6
3	22.1 ± 4.7	22.5 ± 3.9	4.4 ± 4.7	3.6 ± 4.0
4	24.9 ± 5.6	24.0 ± 6.2	4.4 ± 4.2	5.3 ± 5.3

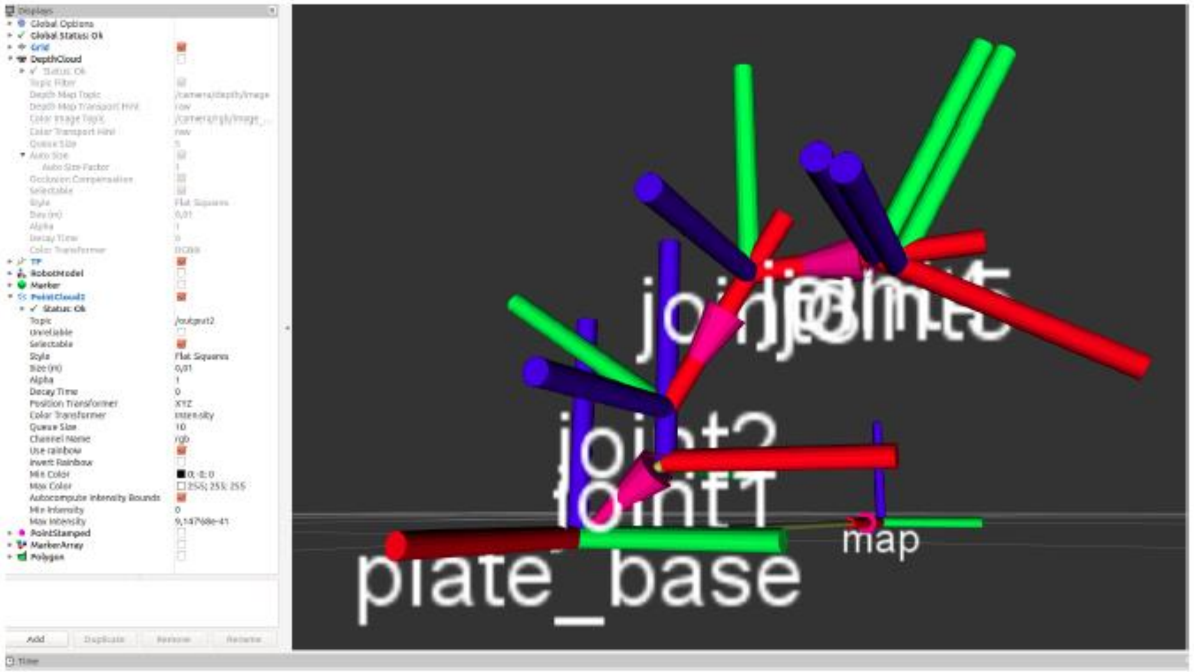
Characteristics



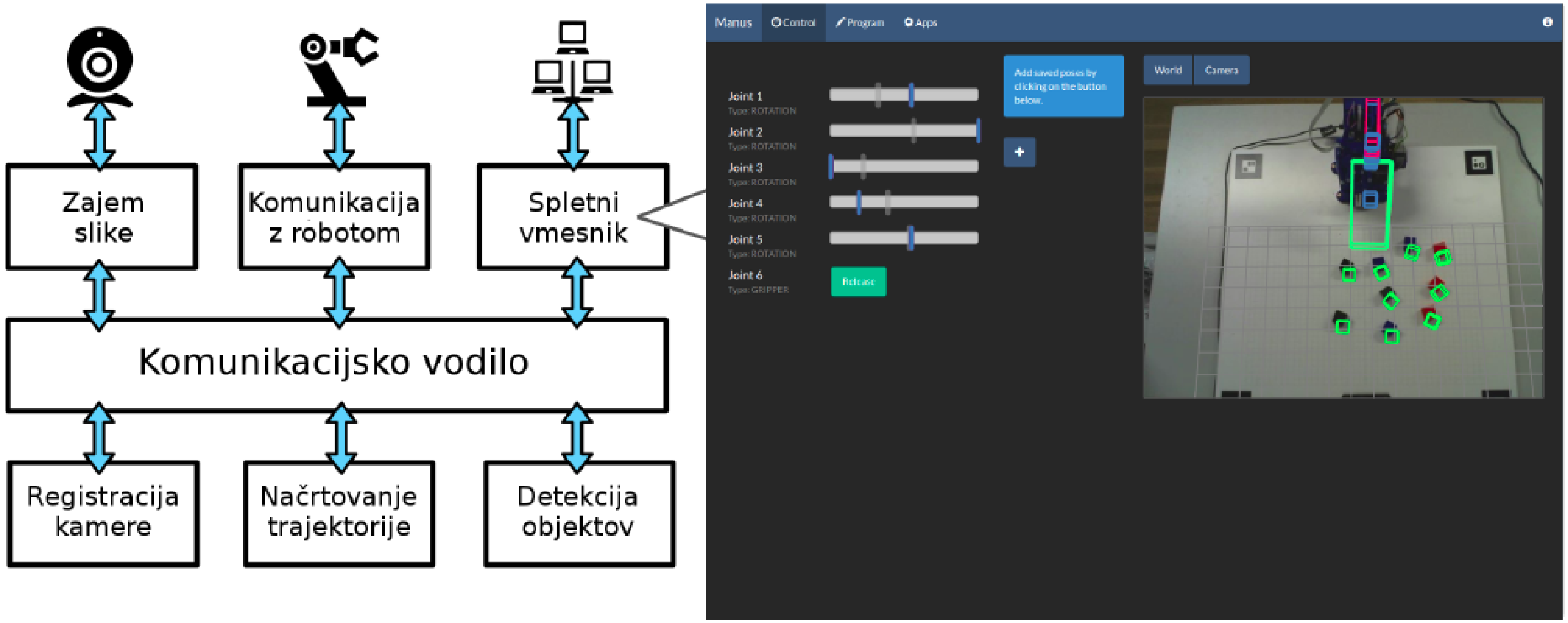
Lastnost	Vrednost
Višina	$358\ mm$
Radij	$310\ mm$
Ponovljivost	$4\ mm$
Teža	$842\ g$
Nosilnost	$80\ g$
Napajanje	$5,0\ V$
Tok	$6000\ mA$

Integration in ROS



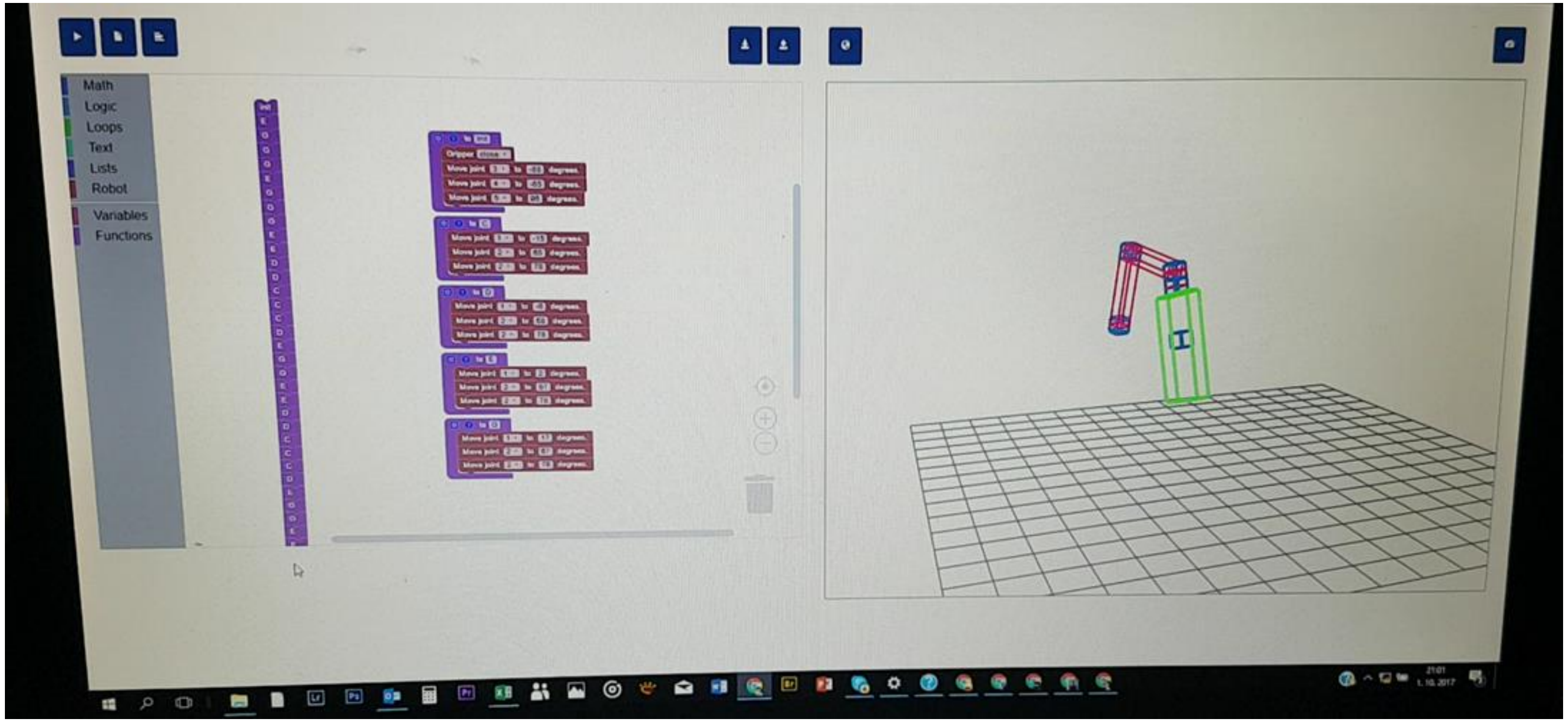


Integration in Manus



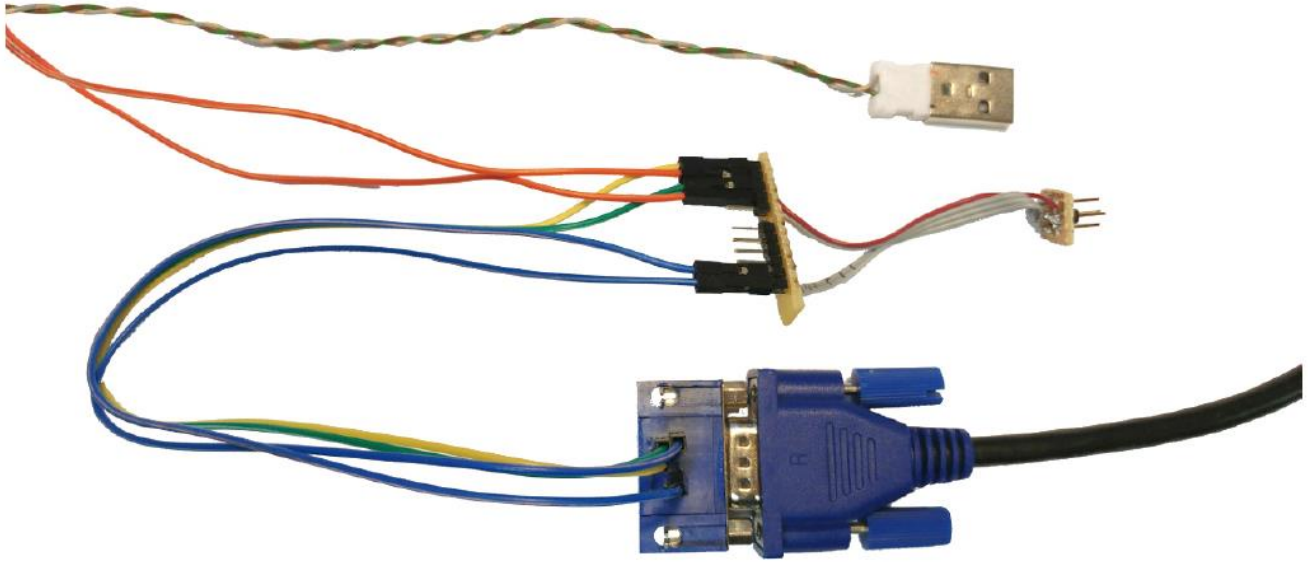
Integration with programming languages

- Matlab
- Python
- Blockly

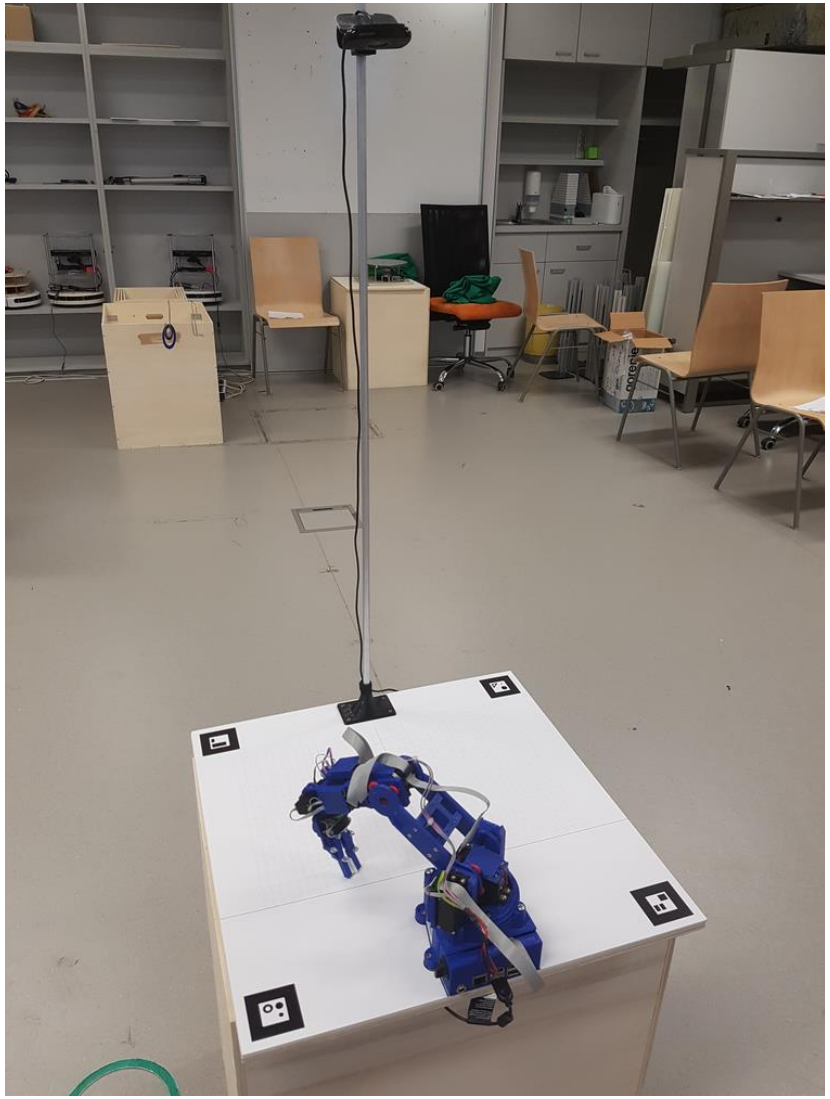


Communication using VGA and USB cable

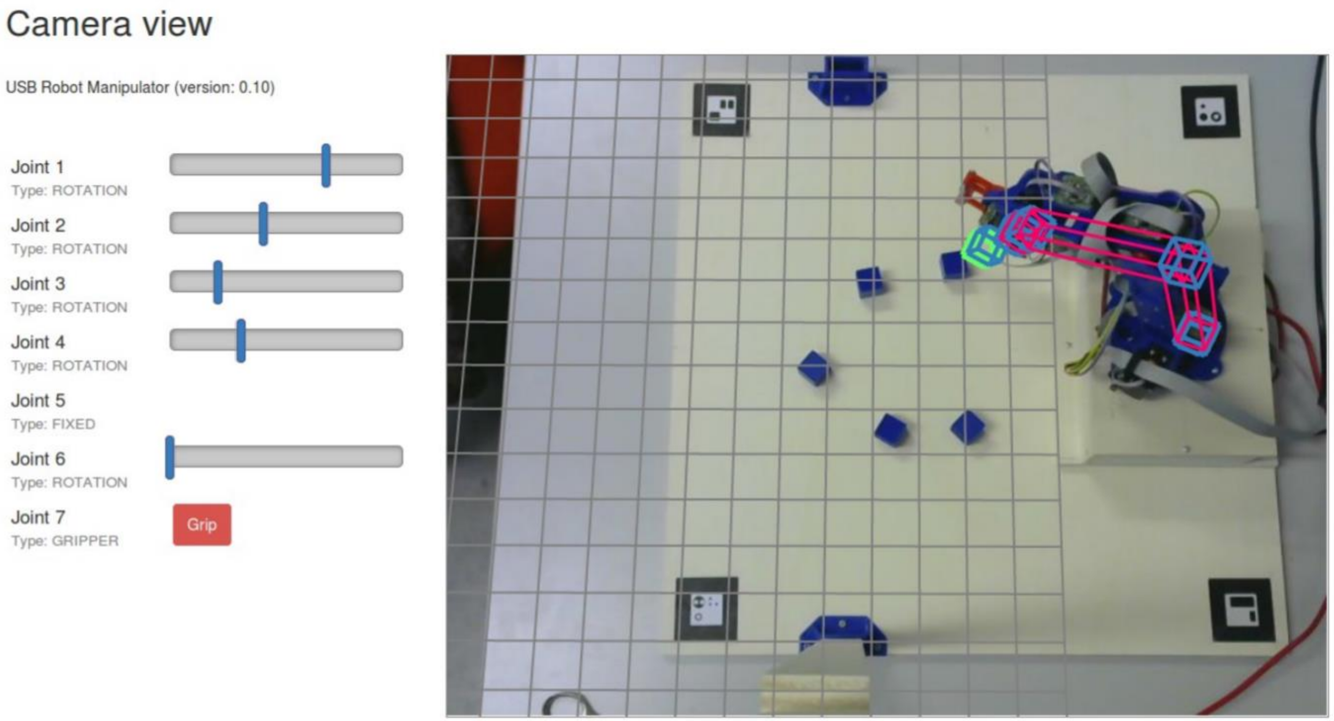
- VGA cable and I^2C protocol
- USB port

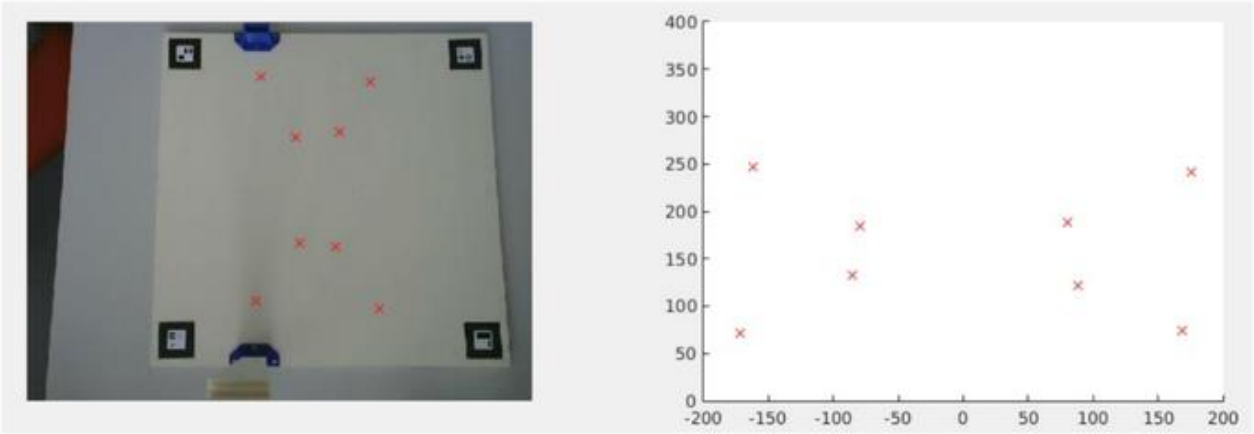


Registration with camera

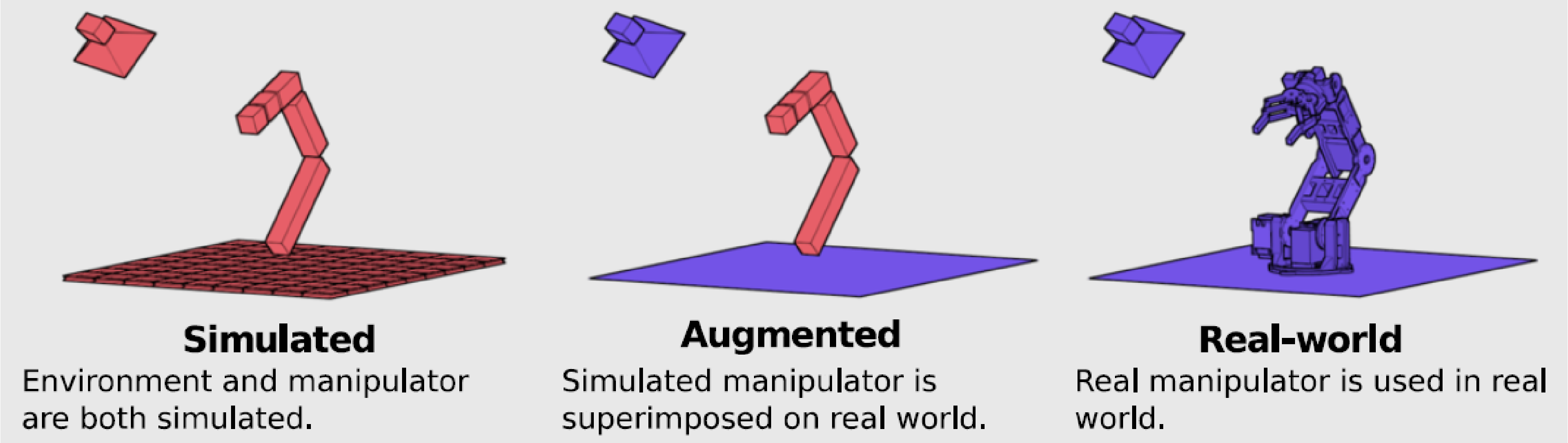


Augmented reality



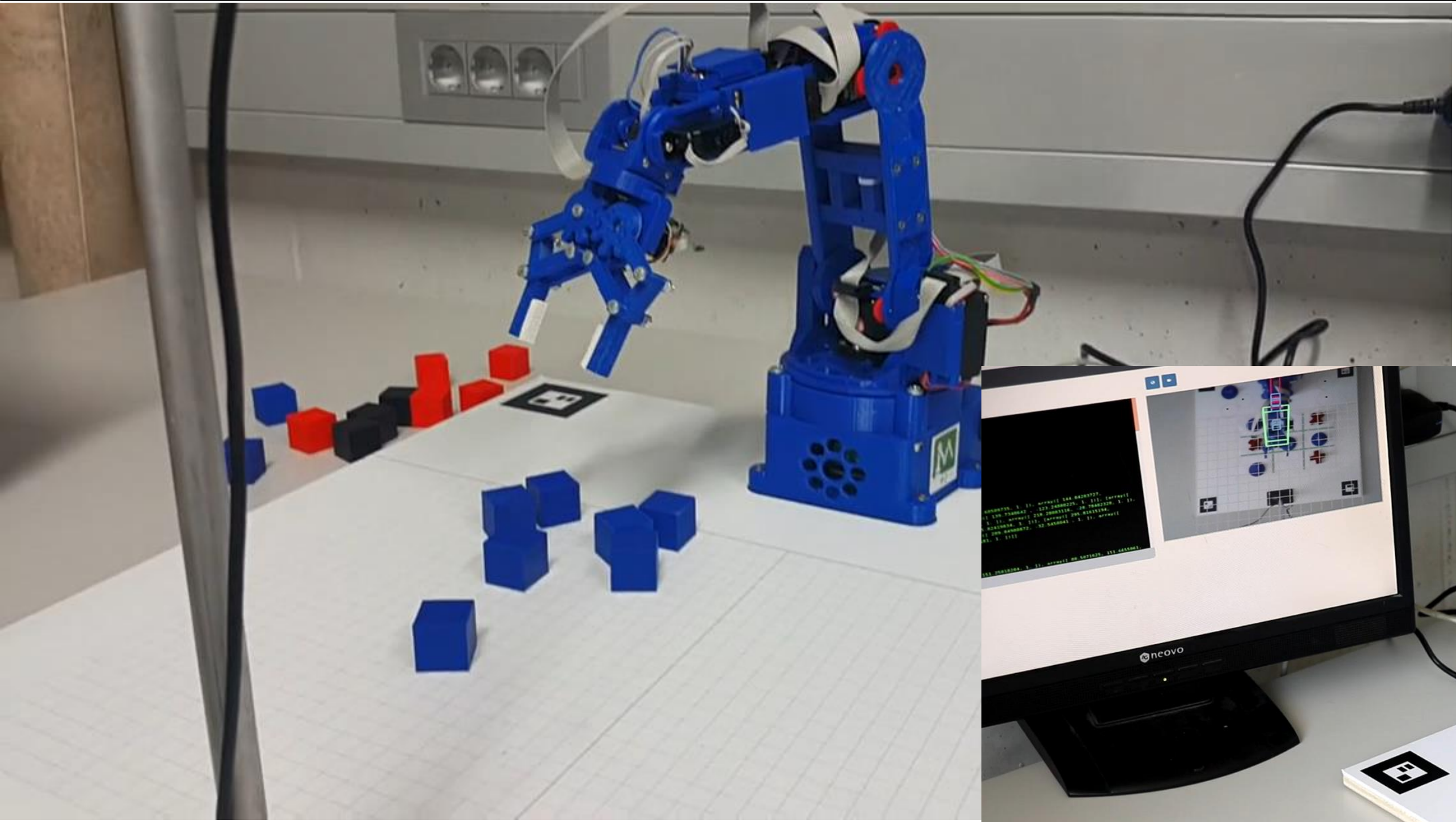


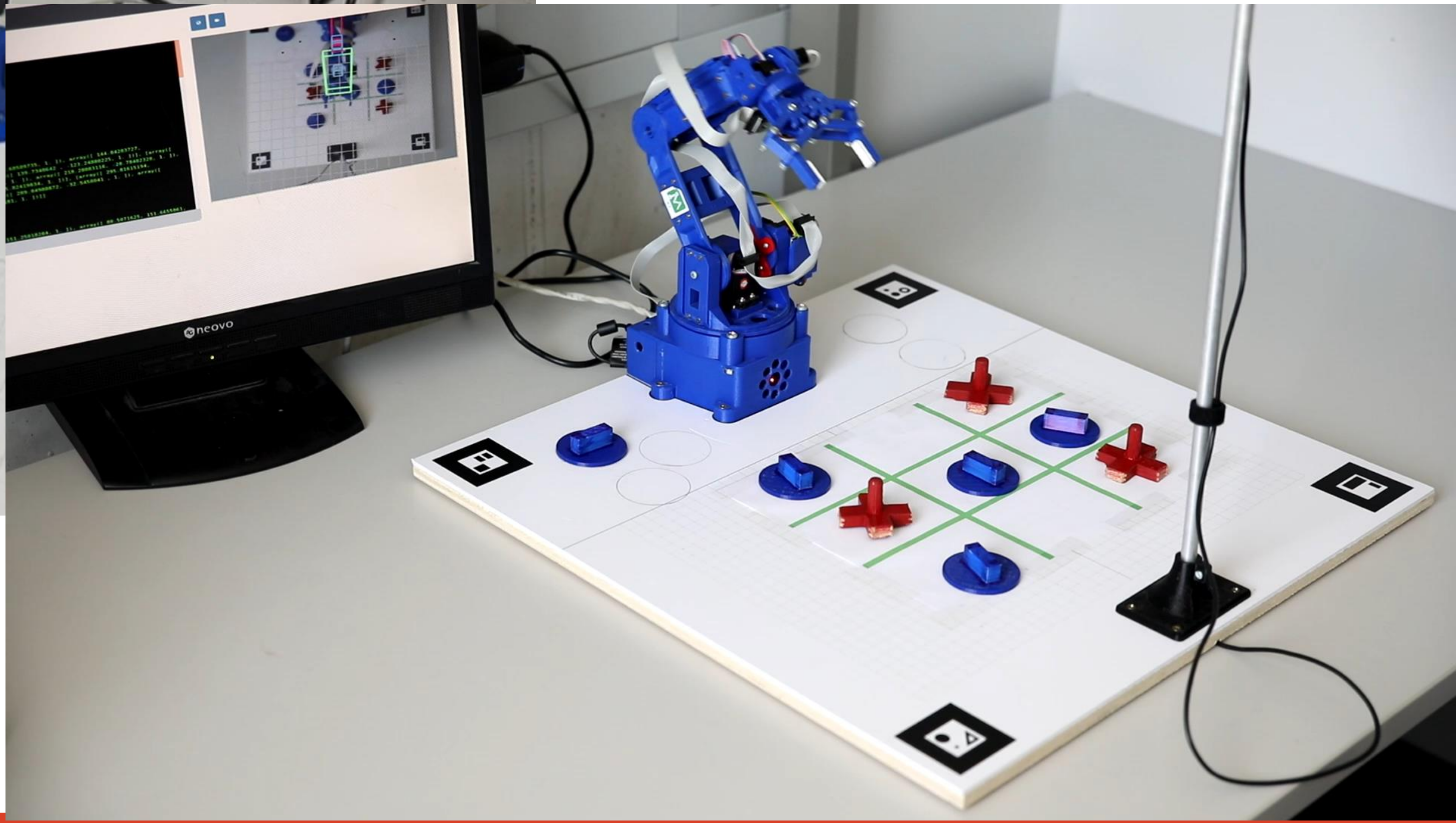
Multi-level teaching approach



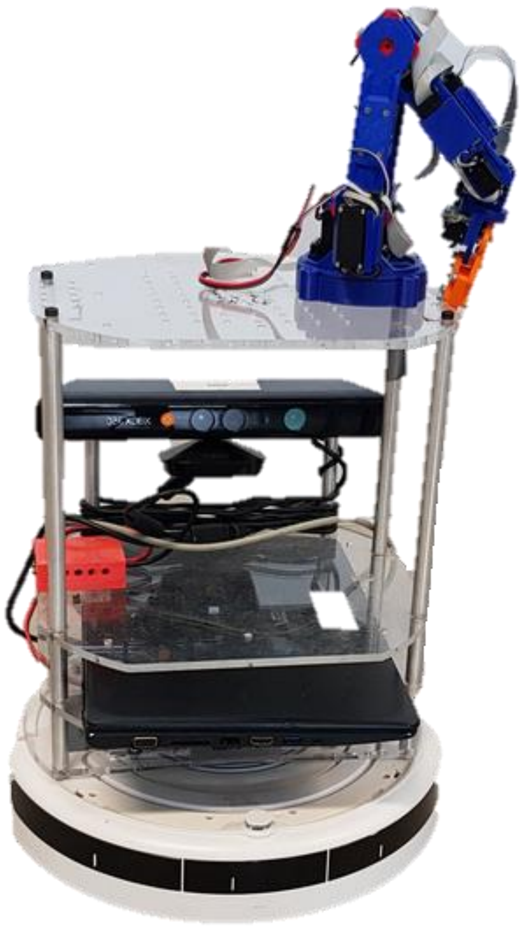
	perception	action	activity
1	simulated	none	Learning basic computer vision algorithms by processing stored images.
2	real	none	Learning more advanced computer vision algorithms by capturing and processing live images.
3	none	simulated	Learning the basics of robot manipulation in simulated environment.
4	none	real	Learning to operate the robot manipulator in the real world.
5	real	simulated	Detecting the objects in the scene and pointing at them with the virtual robot manipulator.
6	real	real	Detecting and grasping the objects in the scene with the physical robot manipulator.

Video

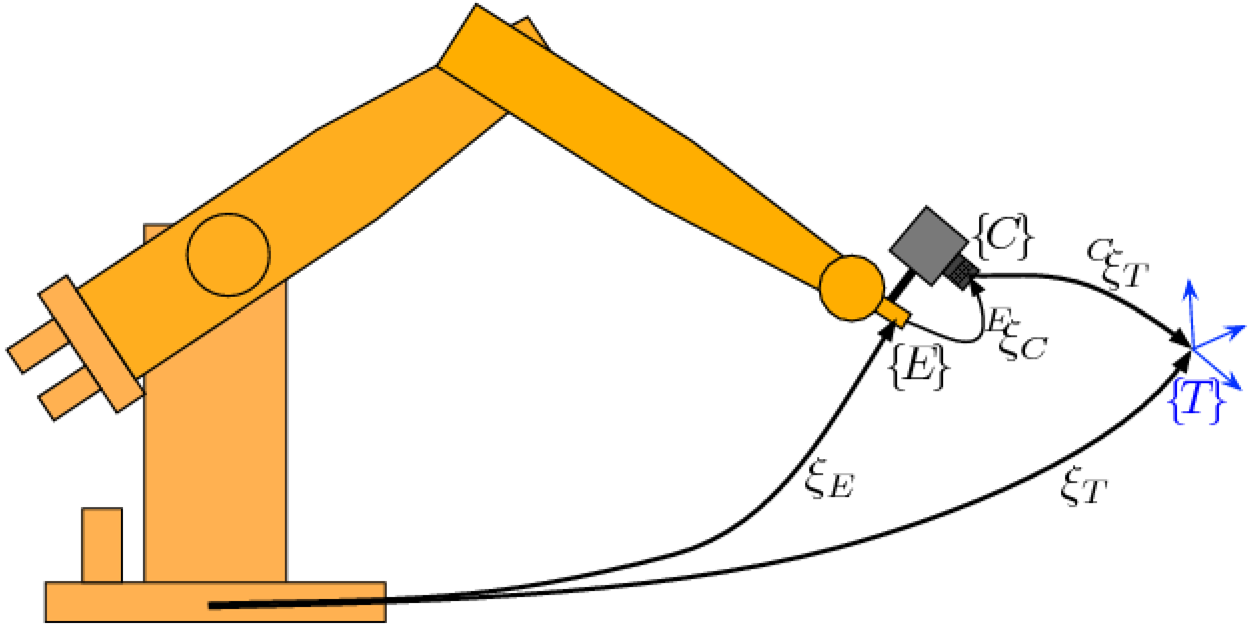




Mobile manipulation

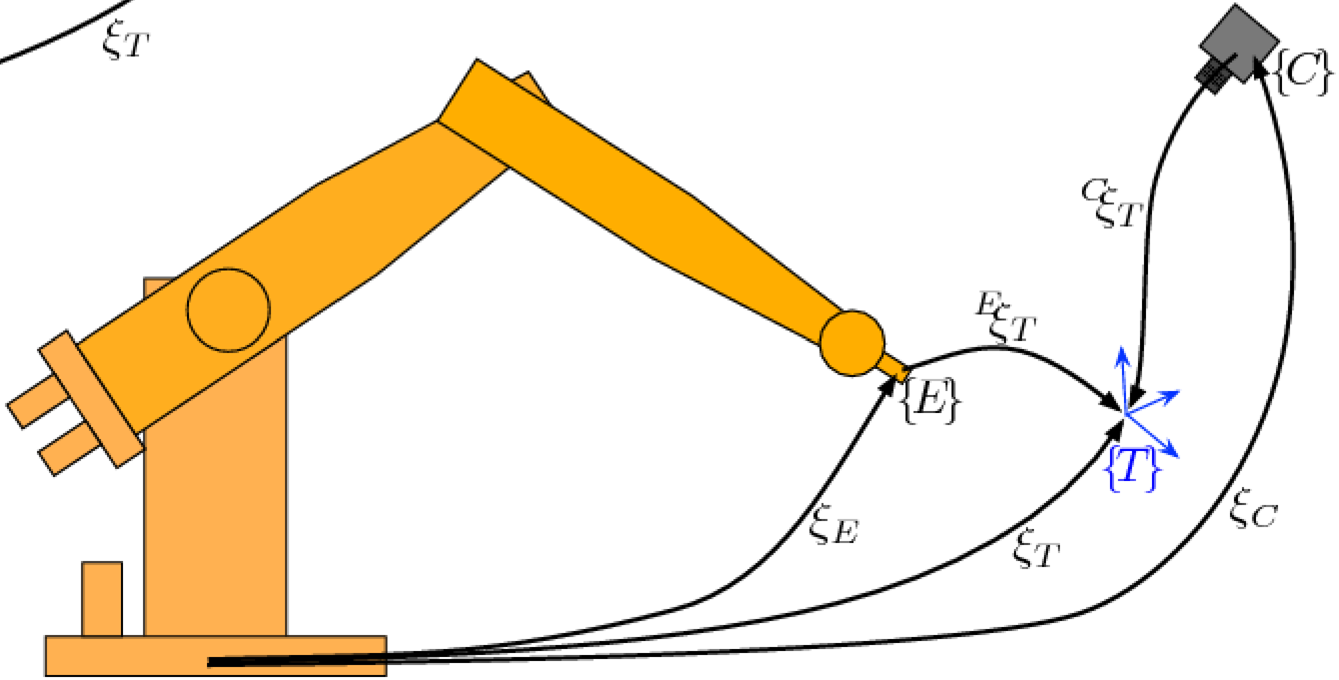


Vision-based control



Moving the end effector to the desired pose determined by the pose of an object

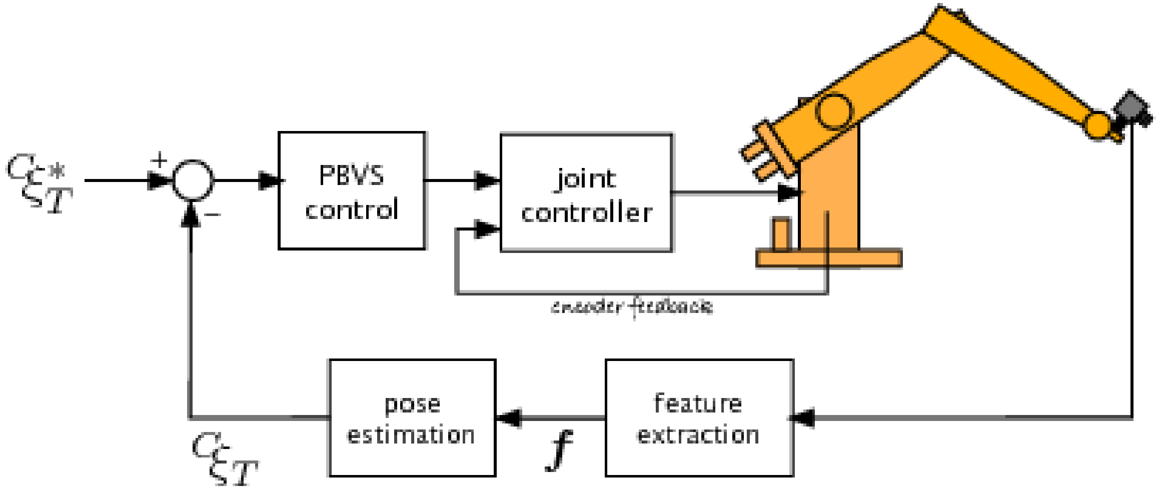
- Camera on the robot



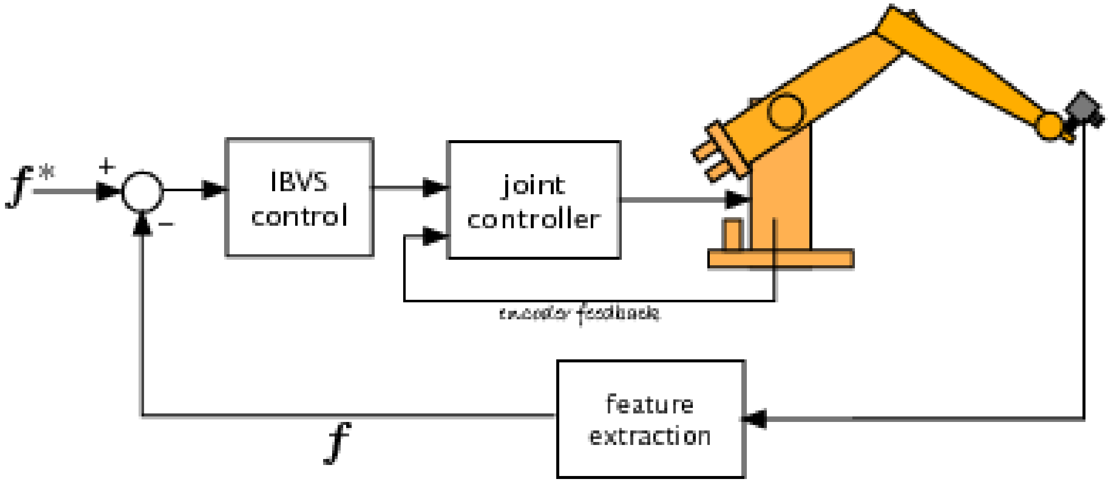
- Fixed camera

Vision-based control

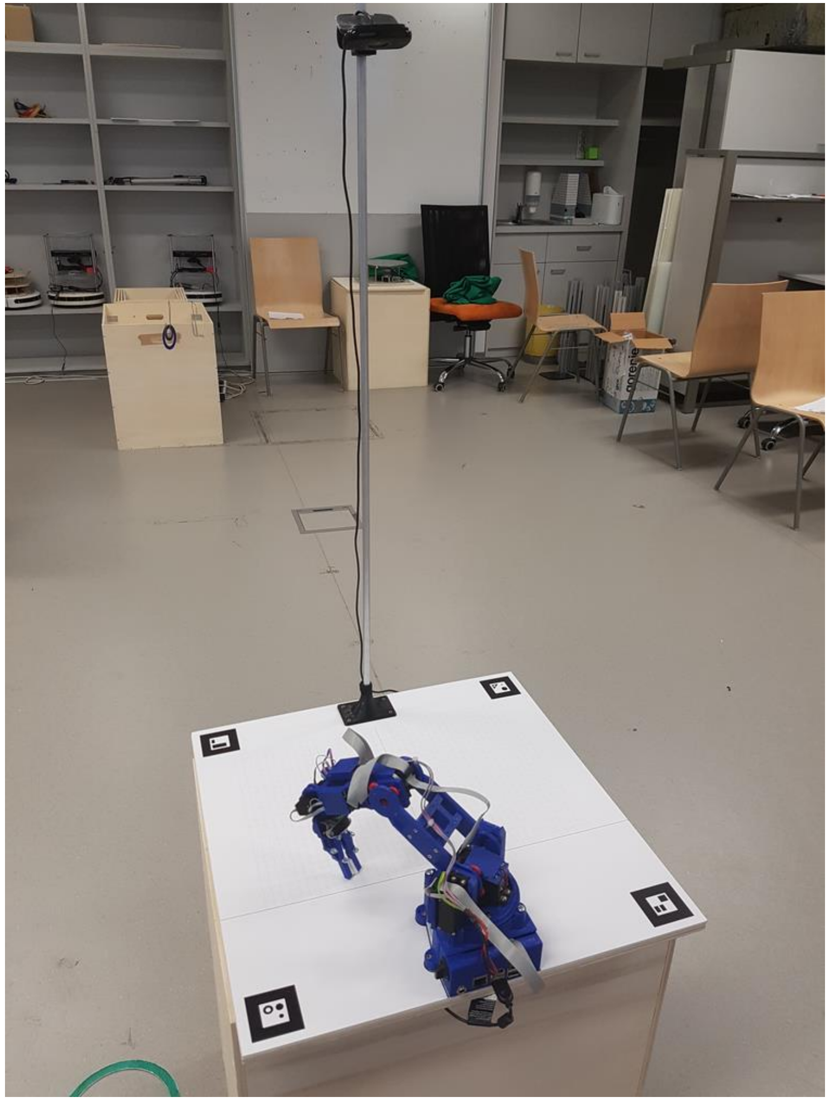
- Position-based servoing
  - Explicit control
  - In world coordinate frame

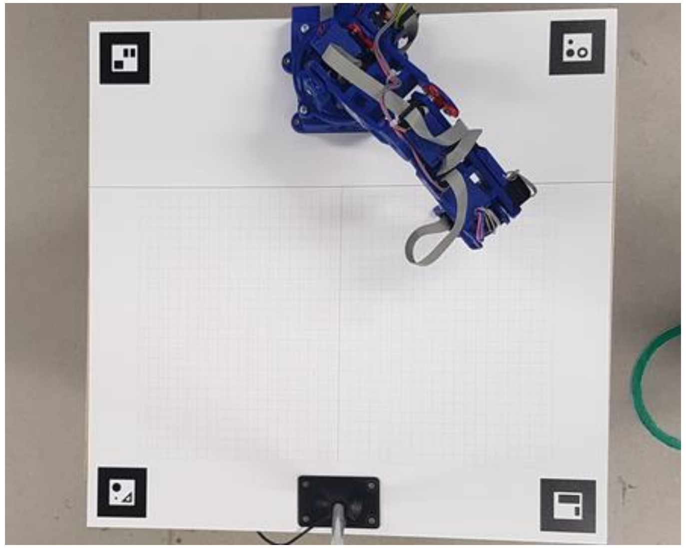


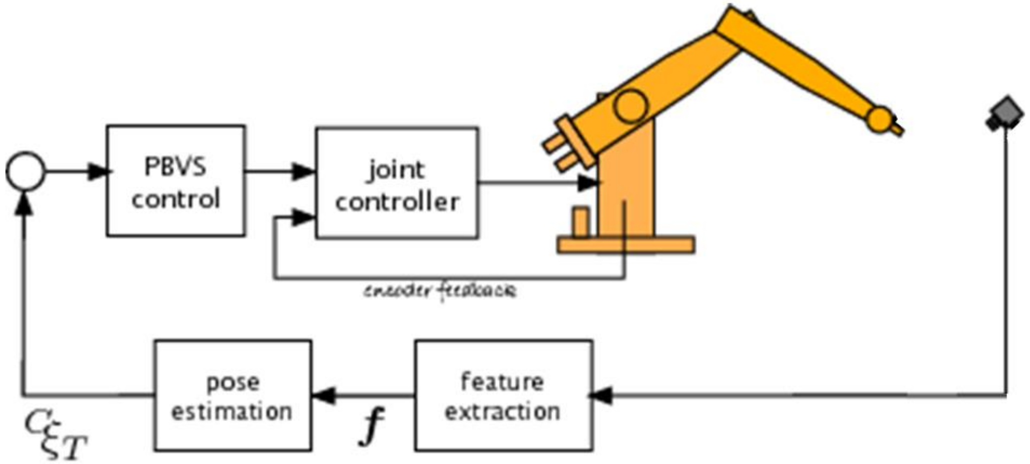
- Image-based servoing
  - Implicit control
  - In image coordinate frame



Vision-based control







Vision-based control

- Mobile manipulation
- Joint control of mobile robot and robot manipulator