FACULTY OF MECHANICAL ENGINEERING

institute of automation and computer science

Cybernetics and Robotics Division

NETME Centre

FACULTY OF MECHANICAL ENGINEERING institute of automation and computer science

Programming for robots and manipulators Lecture 4

Roman Parak@vutbr.cz

BRNO UNIVERSITY OF TECHNOLOGY

IACS

INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

Content

Introduction

Kinematics in Robotics

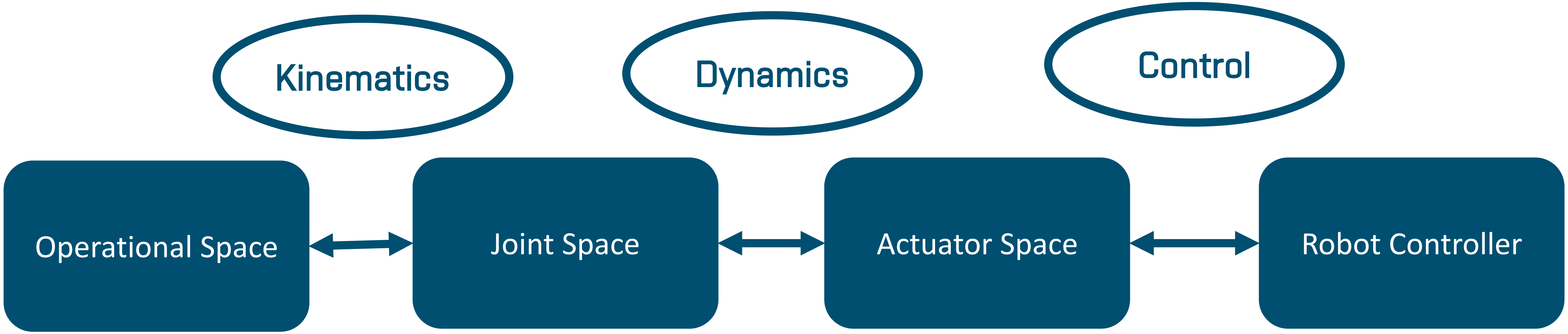
Kinematics analyzes the geometry of a motion analytically, e.g. of a robot:

- With respect to a fixed reference co-ordinate system.
- Without regard to the forces or moments that cause the motion.
- Essential concepts are position and orientation (Euler Angles, Quaternions).

Kinematics describes the analytical relationship between the joint positions and the end-effector position and orientation.

Differential kinematics describes the analytical relationship between the joint motion and the end-effector motion in terms of velocities.

Robotic Control



Kinematics in Robotics

1. Forward (Direct) Kinematics
IN: Joint relations (rotations, translations) for the robot arm.
Task: What is the orientation and position of the end effector?

2. Inverse kinematics
IN: The desired end effector position and orientation.
Task: What are the joint rotations and orientations to achieve this?

$$\theta_1, \theta_2, \dots, \theta_n \xrightarrow{\text{Forward Kinematics}} TCP \begin{cases} x, y, z,\ R_x, R_y, R_z \\ x, y, z,\ q_x, q_y, q_z, q_w \end{cases}$$

$$\theta_1, \theta_2, \dots, \theta_n \xleftarrow{\text{Inverse Kinematics}} TCP \begin{cases} x, y, z,\ R_x, R_y, R_z \\ x, y, z,\ q_x, q_y, q_z, q_w \end{cases}$$

Forward Kinematics

Forward Kinematics

The forward kinematics problem is concerned with the relationship between the individual joints of the robot manipulator and the position and orientation of the tool or end-effector.

Stated more formally, the forward kinematics problem is to determine the position and orientation of the end-effector, given the values for the joint variables of the robot.

The joint variables are the angles between the links in the case of revolute or rotational joints, and the link extension in the case of prismatic or sliding joints.

Denavit–Hartenberg (DH)

The Denavit–Hartenberg parameters are the four parameters associated with a particular convention for attaching reference frames to the links of a spatial kinematic chain, or robot manipulator.

$\boldsymbol{\theta}$

Angle about previous $\boldsymbol{z}$, from old $\boldsymbol{x}$ to new $\boldsymbol{x}$.

$\boldsymbol{d}$

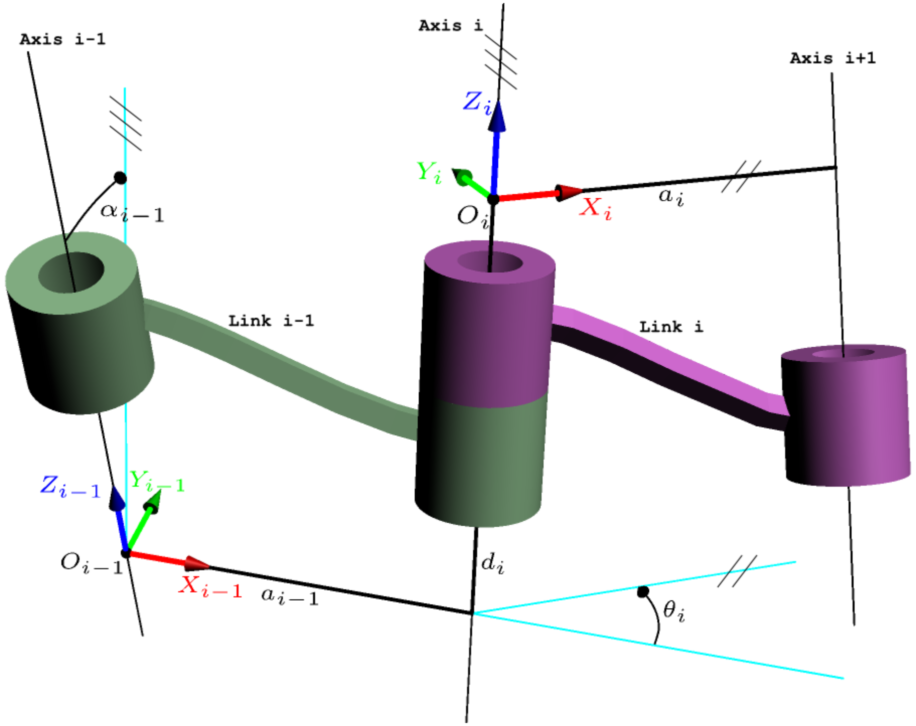
Offset along previous $\boldsymbol{z}$ to the common normal.

$\boldsymbol{a}$

Length of the common normal. Assuming a revolute joint, this is the radius about previous $\boldsymbol{z}$.

$\boldsymbol{\alpha}$

Angle about common normal, from old z axis to new $\boldsymbol{z}$ axis.



[DH Representation]

Denavit–Hartenberg (DH)

In this convention, each homogeneous transformation A_i is represented as a product of four basic transformations.

$$A_i = R_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} R_{x,\alpha_i}$$

$$= \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Denavit–Hartenberg (DH)

In this convention, each homogeneous transformation A_i is represented as a product of four basic transformations.

$$A_i = R_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} R_{x,\alpha_i}$$

$$= \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Translation Part

Rotation Part

Quaternion from Rotation Matrix

Quaternion is based on mathematical complex numbers and is represented by four components x, y, z, w with the formula:

$$\mathrm{Q} = \mathrm{w} + x_i + y_j + z_k$$

Trigonometric methods

Algebraic methods

- Chiaverini-Siciliano′s method
- Hughes′ method
- Shepperd′s method

Numerical methods

- Bar-Itzhack′s method

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \Longrightarrow q_x, q_y, q_z, q_w$$

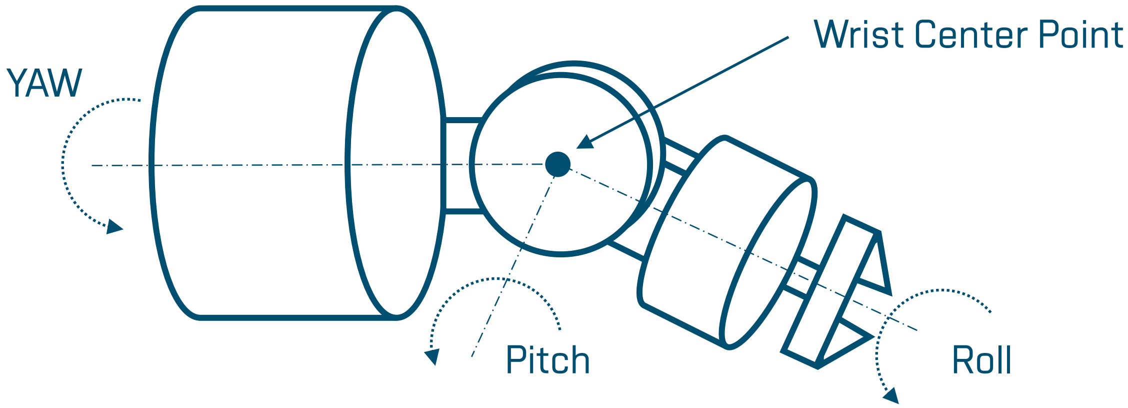
Inverse Kinematics

Inverse Kinematics

Inverse kinematics deals with the problem of finding the required joint angles to produce a certain desired position and orientation of the end-effector. Finding the inverse kinematics solution for a general manipulator can be a very tricky task (Generally, they are non-linear equations).

Close-form solutions may not be possible and multiple, infinity, or impossible solutions can arise. Nevertheless, special cases have a closed-form solution and can be solved.

The sufficient condition for solving a six-axis manipulator is that it must have three consecutive revolute axes that intersect at a common point: Pieper condition.



Methods solving the Inverse kinematics Task

1. Analytical methods (Closed-form solutions):

- Algebraic methods.
- Geometric methods.

2. Numerical methods:

- Symbolic elimination methods: involve analytical manipulations to eliminate variables from a system of nonlinear equations to reduce it to a smaller set of equations.

- Continuation methods: involve tracking a solution path from a start system with known solutions to a target system.

- Iterative methods: are in general based on Newton-Raphson method for finding roots using 1st order approximation of the original algebraic equation. They converge in a single solution (from several possible) based on the initial guess.

Example

Technologies

Python version 3:

https://www.python.org/downloads/

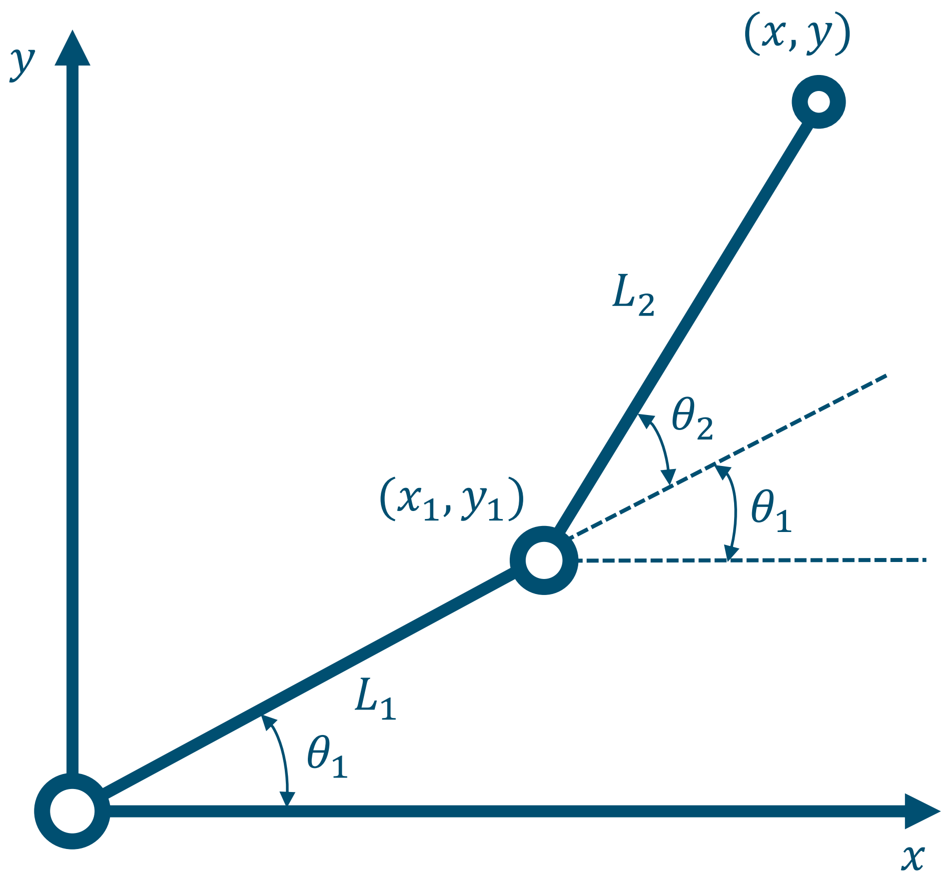
Numpy (Array computing Lib.)

```
pip3 install numpy
```

Mataplotlib (Visualization Lib.)

```
pip3 install matplotlib
```

Forward/Inverse Kinematics Demonstration
ABB IRB 910SC (SCARA)



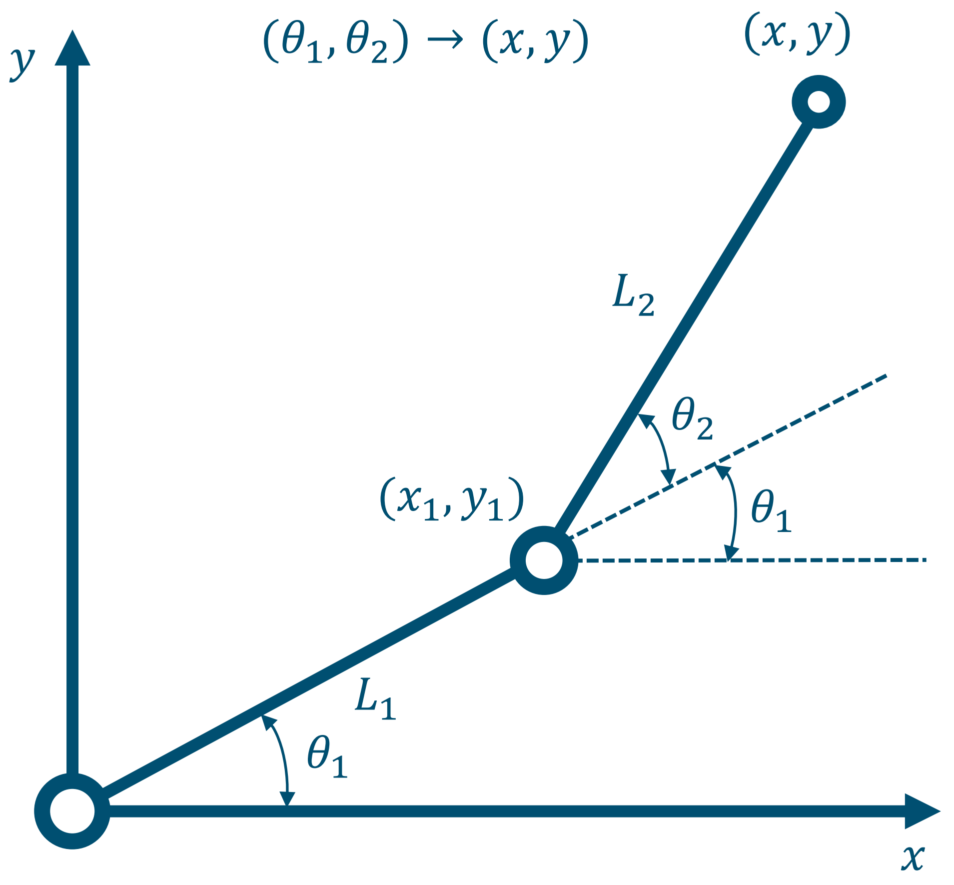
Forward Kinematics

$$(\theta_1, \theta_2) \rightarrow (x, y)$$

Inverse Kinematics

$$(\theta_1, \theta_2) \leftarrow (x, y)$$

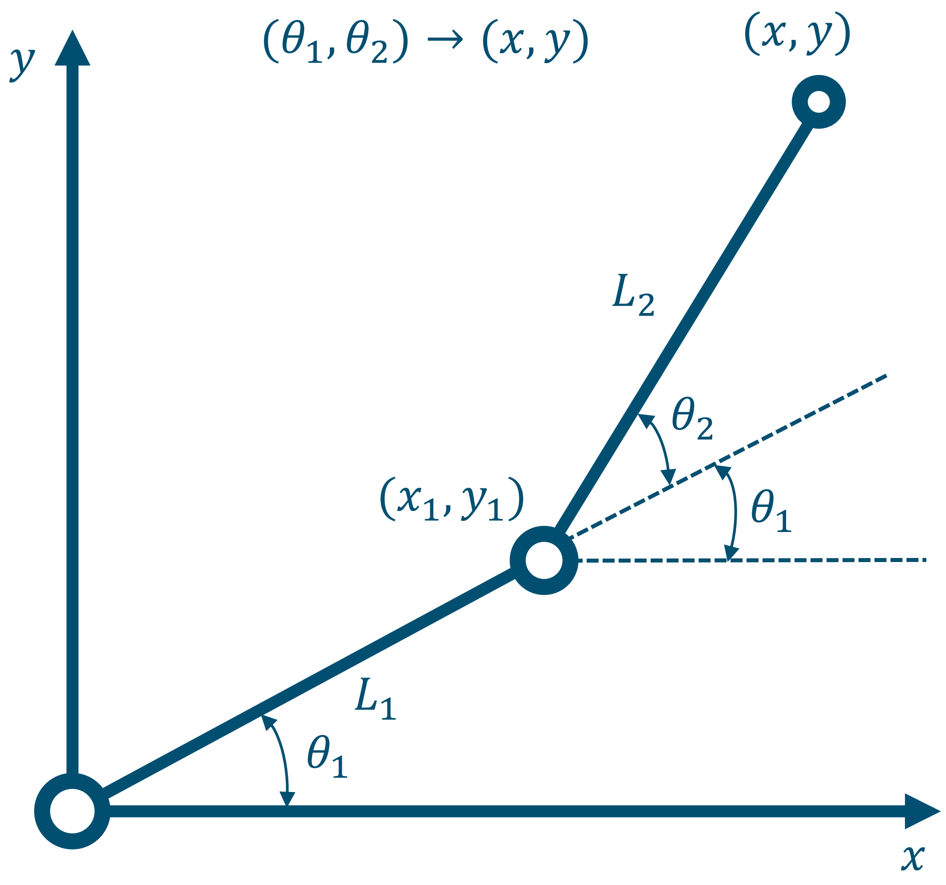
Forward Kinematics



a) DH Parameters – Table?

b) Position of the x_1, y_1 and x, y?

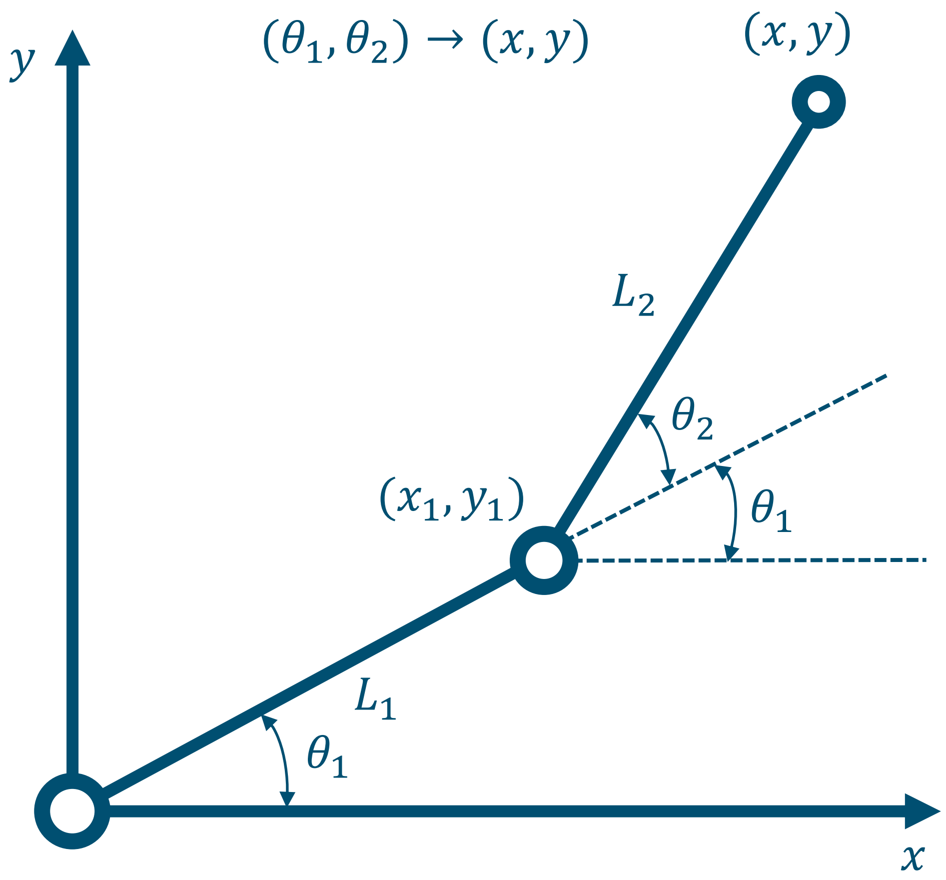
Forward Kinematics



a) DH Parameters – Table?

Link	a_i	α_i	d_i	θ_i
1	$a_1 = L_1$	0	0	θ_1
2	$a_2 = L_2$	0	0	θ_2

Forward Kinematics

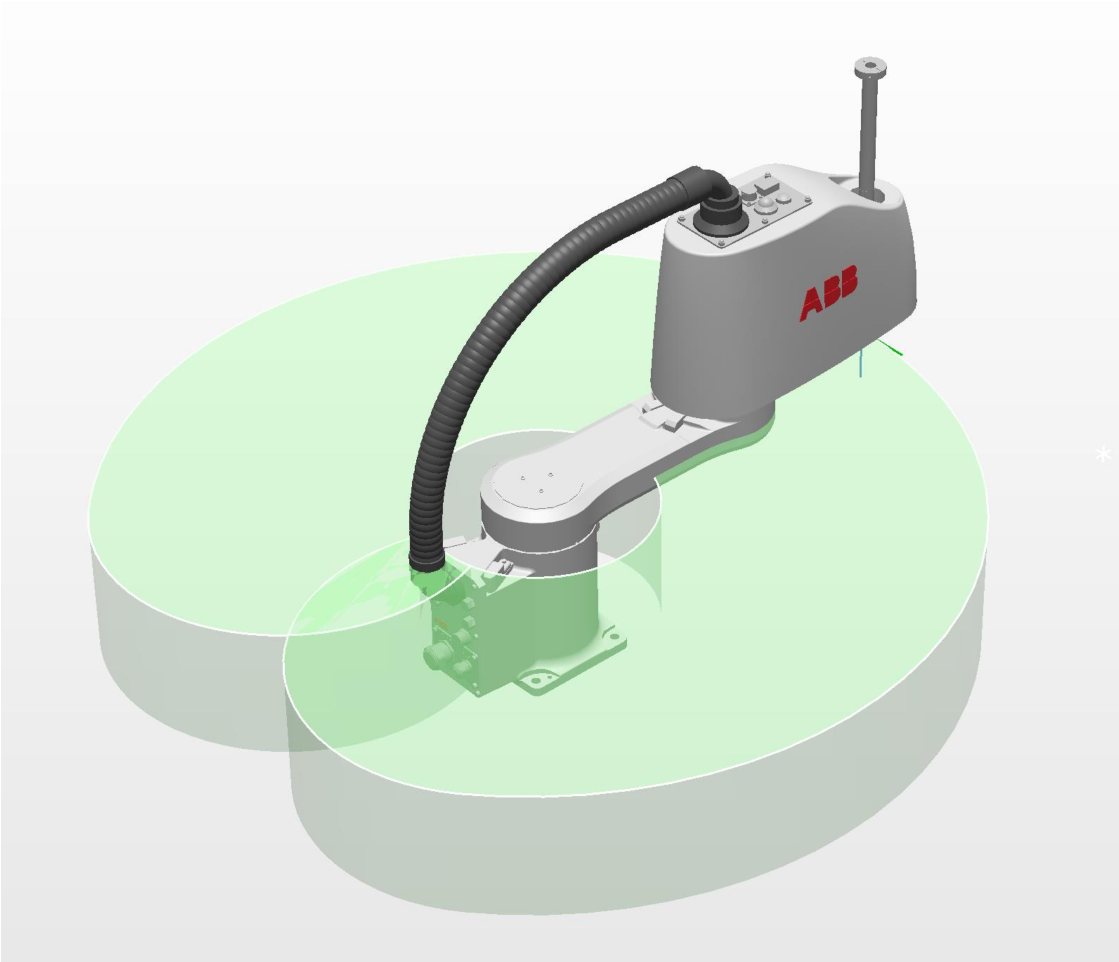


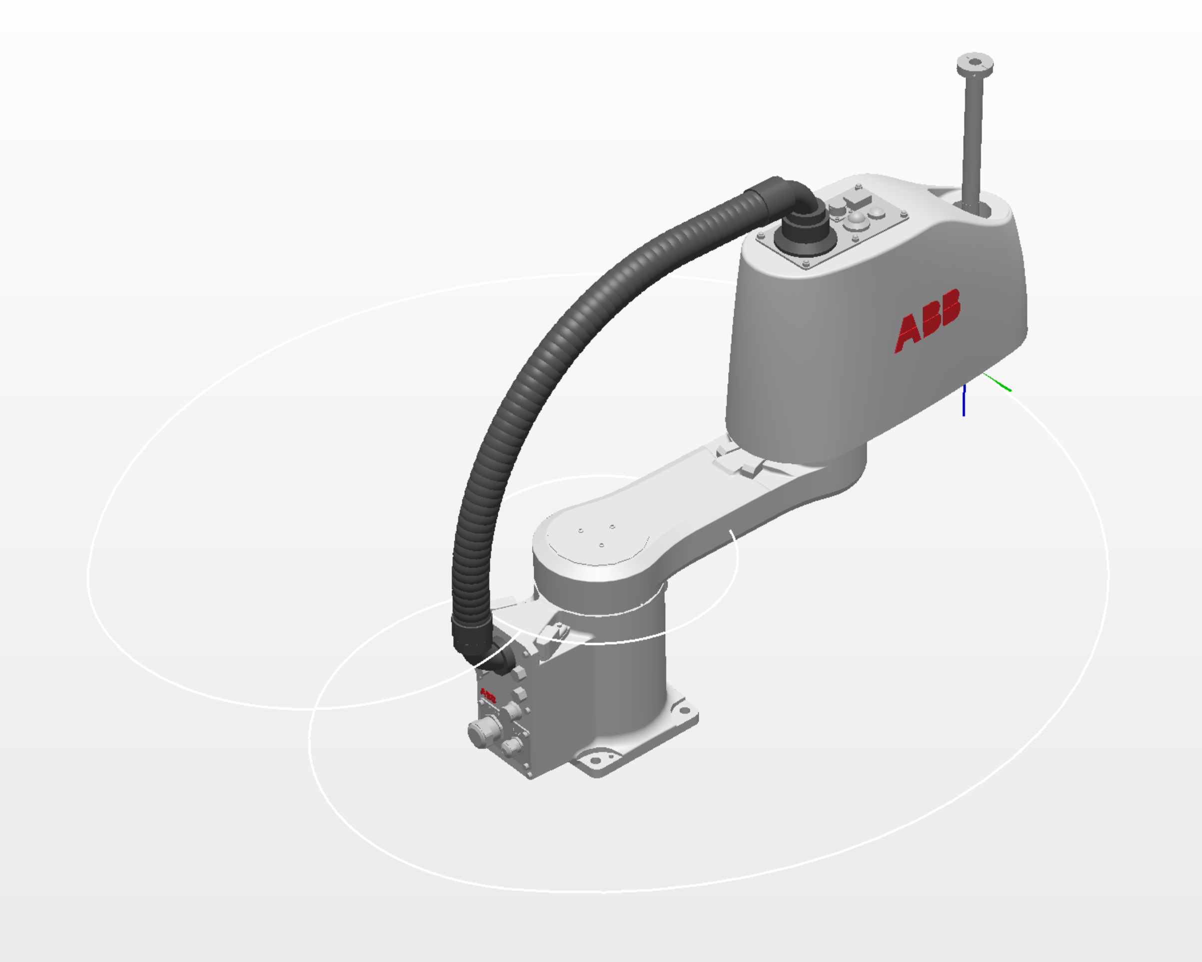
b) Position of the x_1, y_1 and x, y?

$$x_1 = L_1 cos\theta_1$$
$$y_1 = L_1 sin\theta_1$$

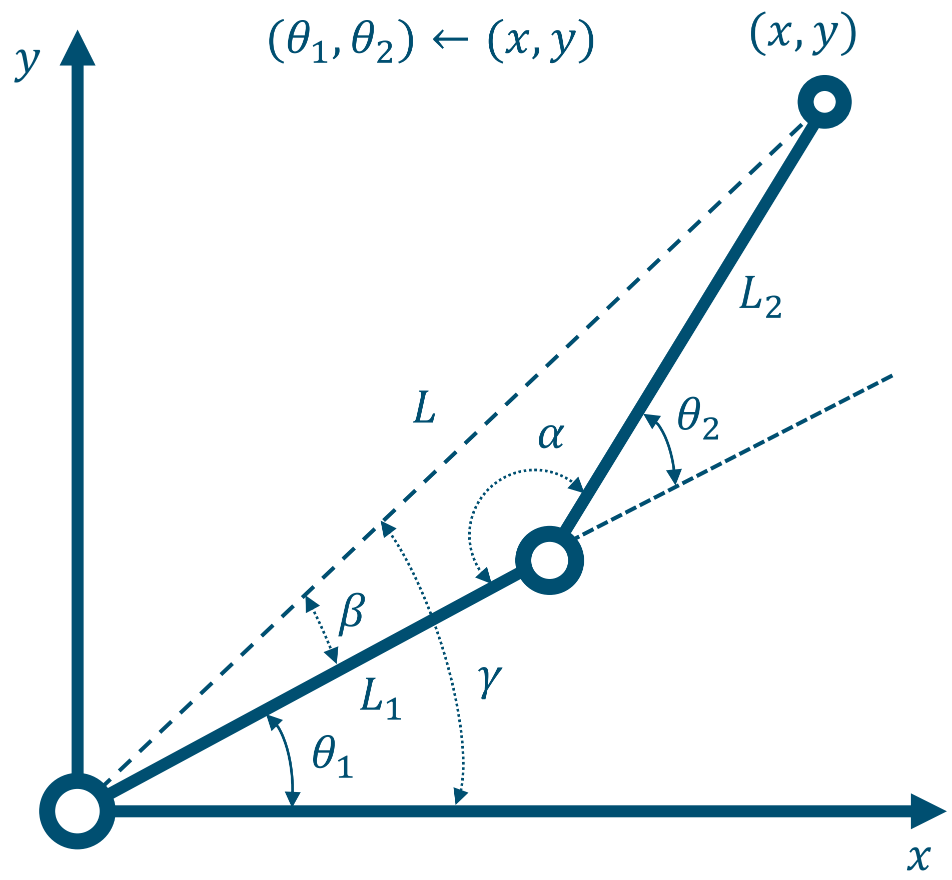
$$x = L_1 cos\theta_1 + L_2 cos(\theta_1 + \theta_2)$$
$$y = L_1 sin\theta_1 + L_2 sin(\theta_1 + \theta_2)$$

Work Envelope



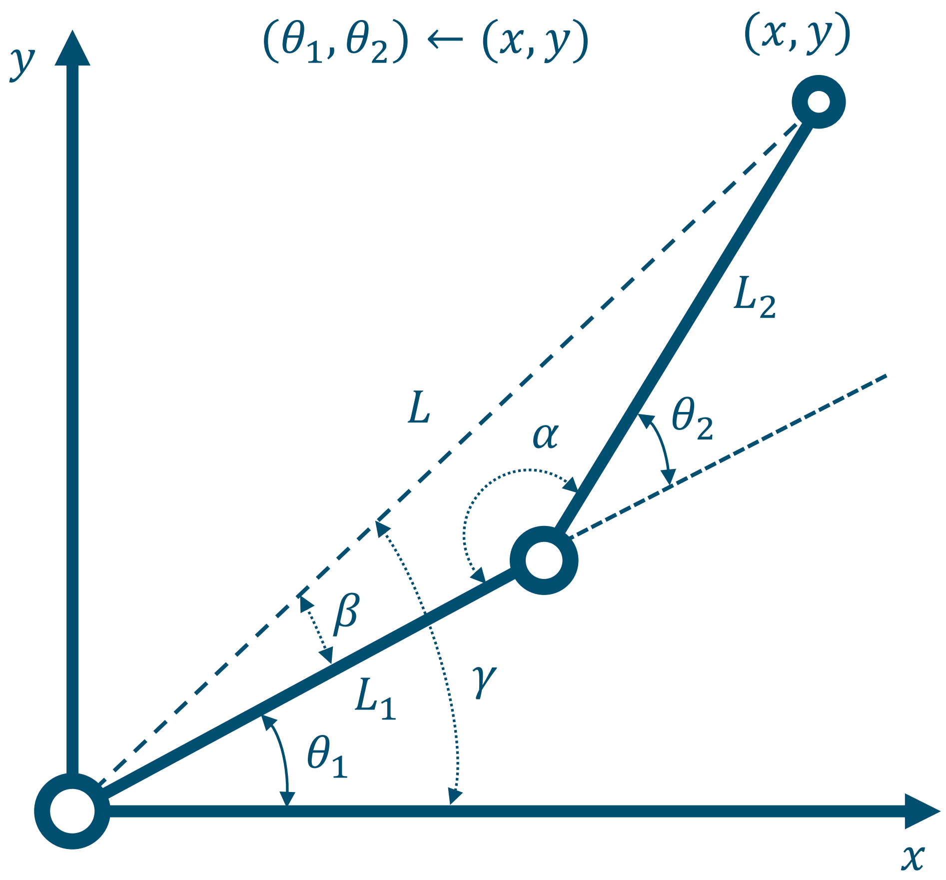


Inverse Kinematics



a) Rotation of the $\boldsymbol{\theta_1}, \boldsymbol{\theta_2}$?

Inverse Kinematics



$$L = \sqrt{x^2 + y^2}$$

$$R_{2DOF} \begin{cases} \theta_1 = \gamma - \beta \\ \theta_2 = \pi - \alpha \end{cases}$$

Law of cosines (Cosine Theorem)

$$L^2 = L_1^2 + L_2^2 - 2L_1L_2cos\alpha$$

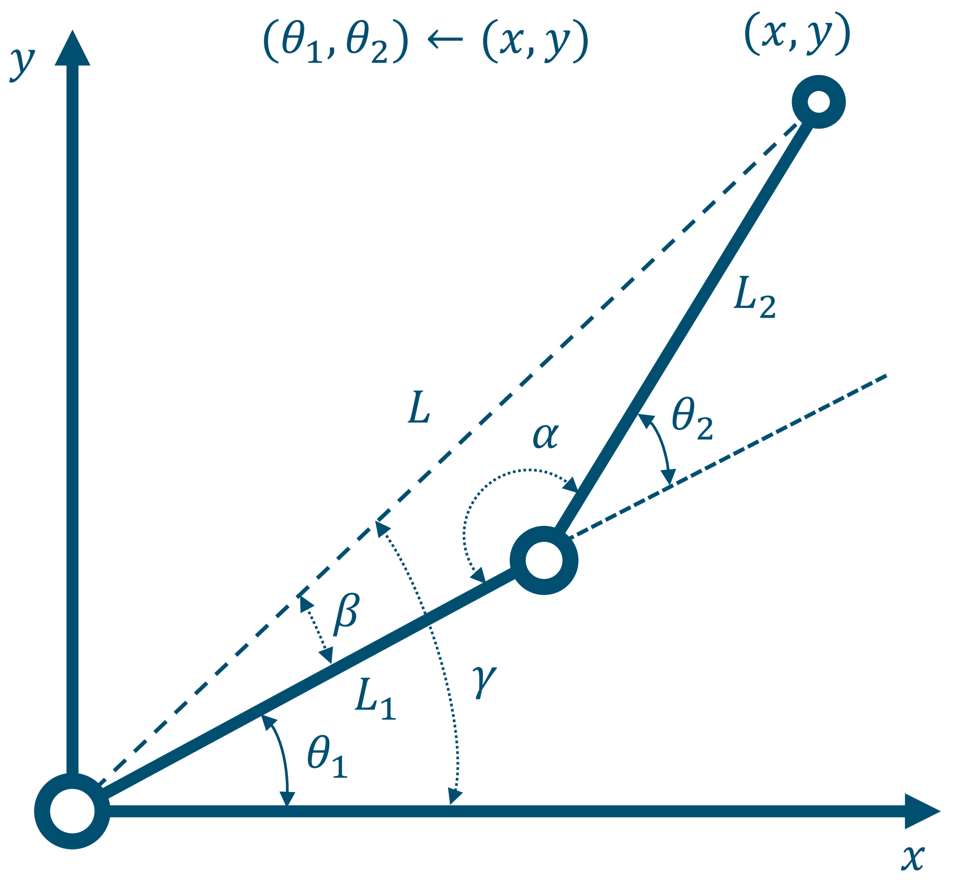
$$L_2^2 = L_1^2 + L^2 - 2L_1Lcos\beta$$

$$cos\alpha = \frac{L_1^2 + L_2^2 - L^2}{2L_1L_2}$$

$$cos\beta = \frac{L_1^2 + L^2 - L_2^2}{2L_1L}$$

$$tan\gamma = \frac{y}{x}$$

Inverse Kinematics



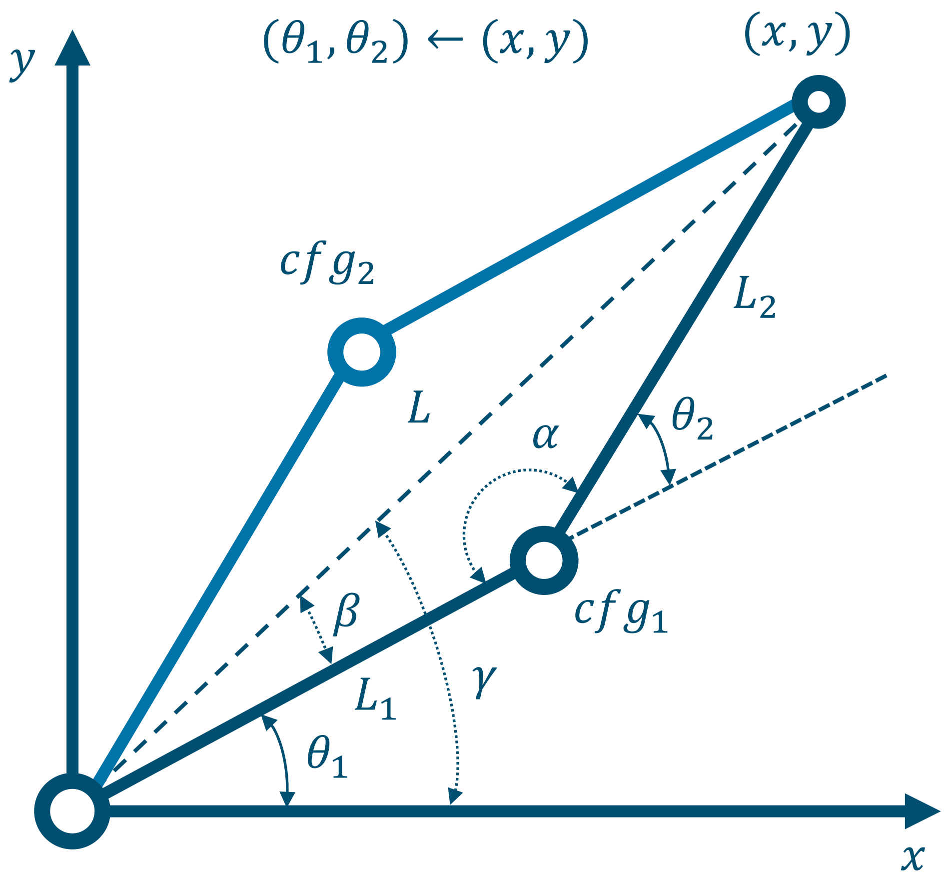
$$R_{2DOF}\begin{cases}\theta_1 = \gamma - \beta \\ \theta_2 = \pi - \alpha\end{cases}$$

Inverse Trigonometric Functions

$$\theta_1 = arctan\frac{y}{x} - \arccos\frac{L_1^2 + L^2 - L_2^2}{2L_1 L}$$

$$\theta_2 = \pi - \arccos\frac{L_1^2 + L_2^2 - L^2}{2L_1 L_2}$$

Inverse Kinematics

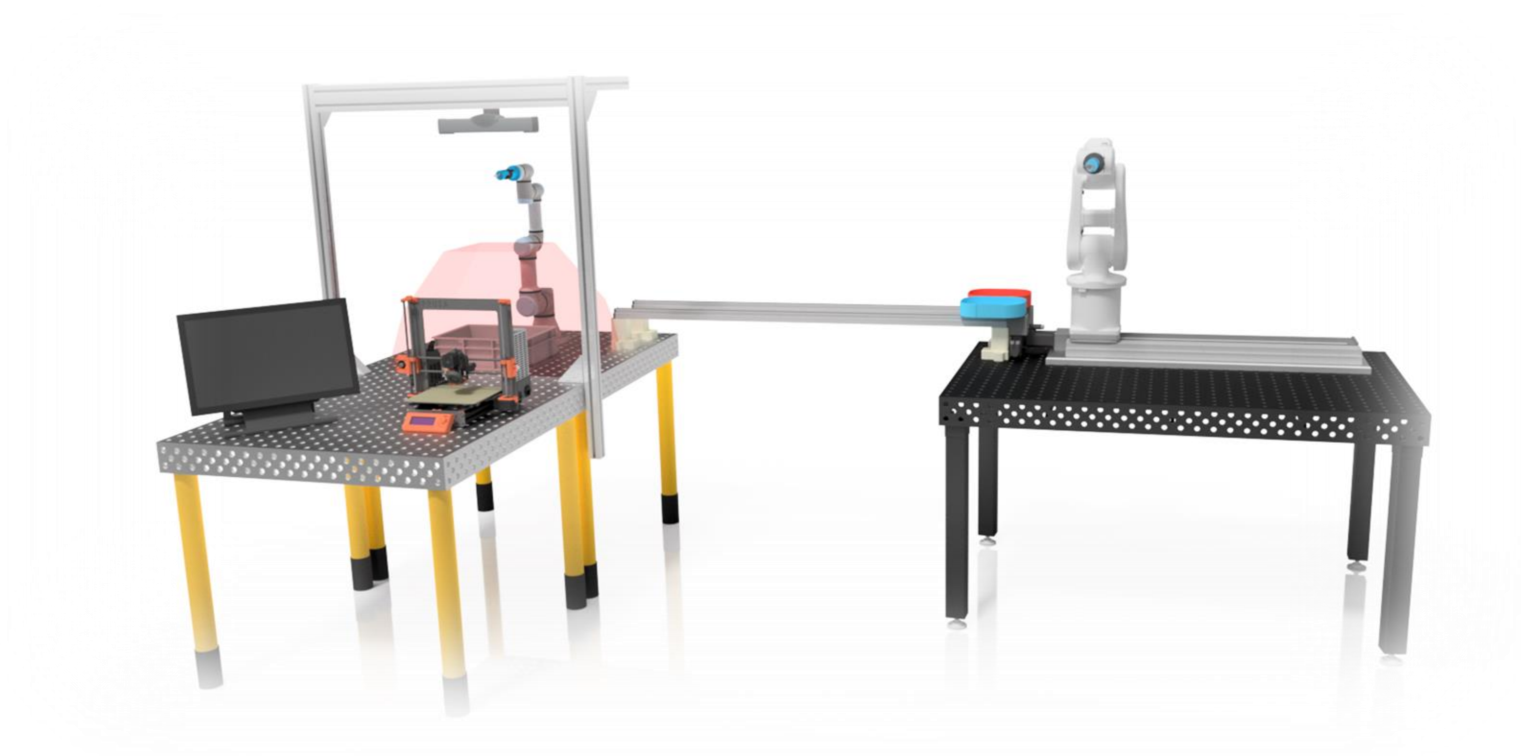


$$cfg_1 \begin{cases} \theta_1 = \gamma - \beta \\ \theta_2 = \pi - \alpha \end{cases}$$

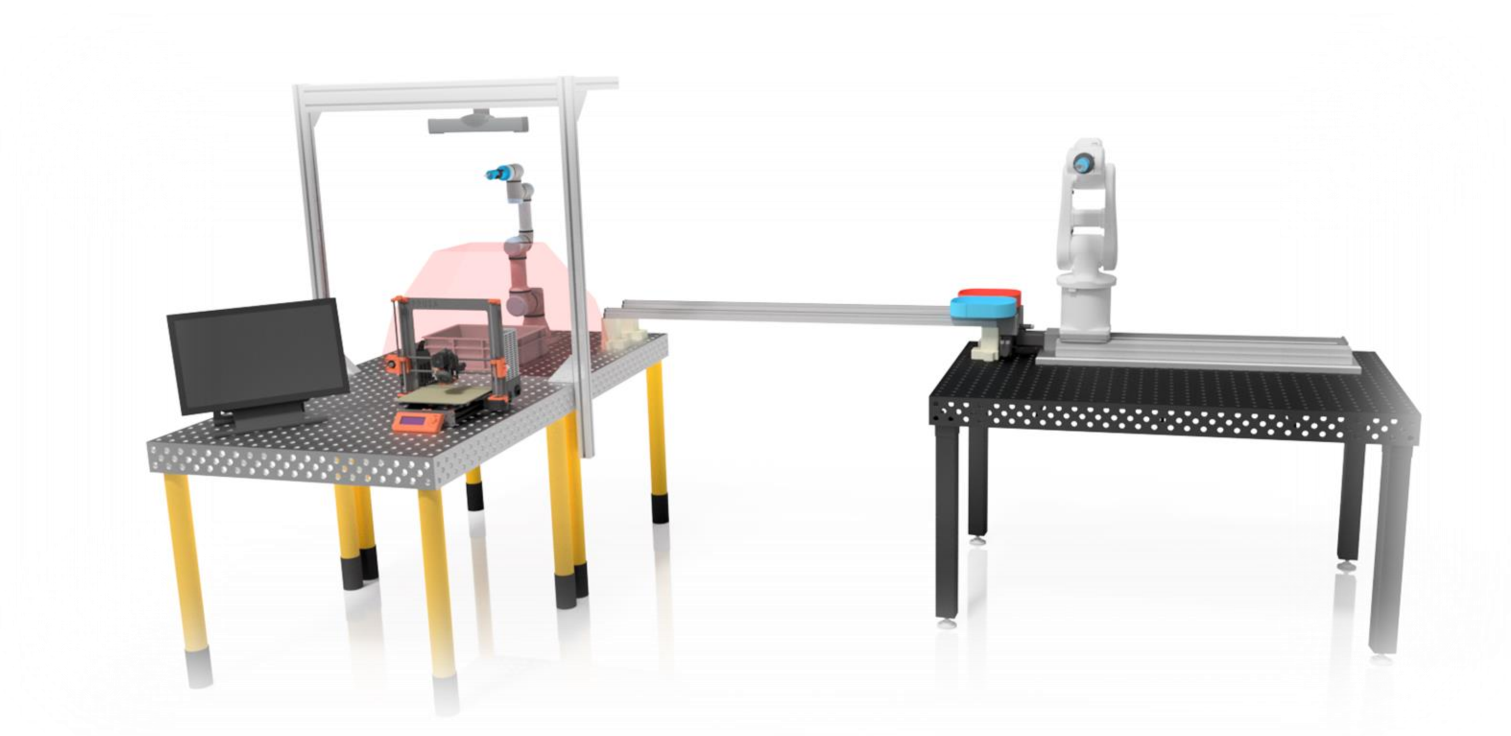
$$cfg_2 \begin{cases} \theta_1 = \gamma + \beta \\ \theta_2 = \alpha - \pi \end{cases}$$

The number of solutions depends on the number of joints in the manipulator.

Thank You!



Questions?



FACULTY OF MECHANICAL ENGINEERING

institute of automation and computer science