Invers Kinematics, singularities and robot's workspace

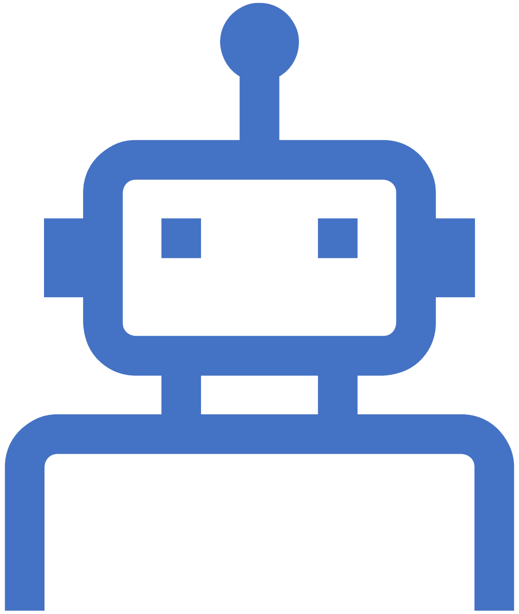
Robotics and Computer vision (RoVi)

Aljaz Kramberger

alk@mmmi.sdu.dk

SDU Robotics
The Maersk Mc-Kinney Moller Institute
University of Southern Denmark

Based on M. Mihelj, T. Bajd. et al. Robtics – second edition, Springer, 2018
Siciliano, Bruno, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. Robotics: modelling, planning and control. Springer Science & Business Media, 2010.



Agenda

- Equivalent angle axis
- joint limits and singularities,
  - example
- definition of robot’s workspace,
  - workplace vs. reachability,
  - example.

Equivalent angle axis

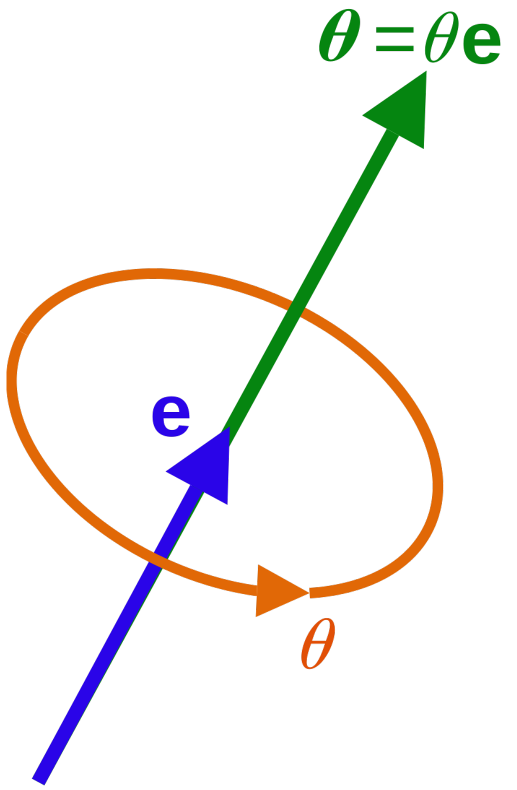
Equivalent angles axis

- Can interpolate rotation well
- Compact representation with 3 parameters
- Messy to concatenate and might need to convert to matrix form
- There are two singularities for axis-angle representation.
  - The first is at $\theta=0$, where every axis represents the identity rotation.
  - The second is at $\theta=\pi$, where each axis and its negation represent the same rotation.

Equivalent angle axis

- angle axis representation defines a rotation in three-dimensional Euclidian space, by:
  - ***unit vector*** describing the direction of the axis rotation,
  - ***angle***, describing the magnitude of the rotation.
- For example:

$$(axis, angle) = \left(\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}, \theta\right) = \left(\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \pi\right) = \begin{bmatrix} 0 \\ 0 \\ \pi \end{bmatrix}$$



https://en.wikipedia.org/wiki/Axis%E2%80%93angle_representation#/media/File:Angle_axis_vector.svg

Equivalent angle axis

- The shortest rotational path between two coordinate frames is to rotate around the so called ***Equivalent Angle Axis (EAA).***
- A rotation angle θ around an arbitrary unit vector $\mathbf{v} = (v_1, v_2, v_3)^T$ gives a rotation matrix:

$$\mathbf{R}_{EAA}(\mathbf{v},\theta) = \begin{bmatrix} v_1^2(1-C_\theta)+C_\theta & v_1v_2(1-C_\theta)-v_3S_\theta & v_1v_3(1-C_\theta)+v_2S_\theta \\ v_1v_2(1-C_\theta)+v_3S_\theta & v_2^2(1-C_\theta)+C_\theta & v_2v_3(1-C_\theta)-v_1S_\theta \\ v_1v_3(1-C_\theta)-v_2S_\theta & v_2v_3(1-C_\theta)+v_1S_\theta & v_3^2(1-C_\theta)+C_\theta \end{bmatrix}$$

- where $v_i = [\mathrm{v}]_i$. we often also use the following notation:

$$\boldsymbol{R}_{eaa}(\theta\mathrm{v}) \equiv \boldsymbol{R}_{EAA}(\mathrm{v},\theta)$$

Equivalent angle axis

- If a rotation matrix is given **R**, the EAA representation is calculated as follows (assuming $0 \leq \theta \leq \pi$)

$$
\begin{aligned}
\theta &= \cos^{-1}\left(\frac{R_{11} + R_{22} + R_{33} - 1}{2}\right) \\
\mathbf{v} &= \frac{1}{2\sin\theta}\begin{bmatrix} R_{32} - R_{23} \\ R_{13} - R_{31} \\ R_{21} - R_{12} \end{bmatrix} \\
&= \frac{1}{\|(R_{32} - R_{23}, R_{13} - R_{31}, R_{21} - R_{12})\|}\begin{bmatrix} R_{32} - R_{23} \\ R_{13} - R_{31} \\ R_{21} - R_{12} \end{bmatrix}
\end{aligned}
$$

Equivalent angle axis

- Which we formally define as $\theta \mathbf{v} = \mathbf{W}_{\mathrm{rot}}(\mathbf{R})$, where we have

$$\mathbf{W}_{rot}(\mathbf{R}) = \frac{\cos^{-1}\left(\frac{R_{11}+R_{22}+R_{33}-1}{2}\right)}{\|(R_{32}-R_{23}, R_{13}-R_{31}, R_{21}-R_{12})\|} \begin{bmatrix} R_{32}-R_{23} \\ R_{13}-R_{31} \\ R_{21}-R_{12} \end{bmatrix}$$

- In singularities ($\theta \simeq 0$ *or* $\theta \simeq \pi$) this formula should be avoided, thus if $\theta \simeq 0$ ($\sin\theta \simeq 0$) we use:

$$\mathbf{W}_{rot}(\mathbf{R}) = \frac{1}{2} \begin{bmatrix} R_{32}-R_{23} \\ R_{13}-R_{31} \\ R_{21}-R_{12} \end{bmatrix}$$

Equivalent angle axis

- Or if $\theta \simeq \pi$, $\sin\theta \simeq \pi - \theta$ we obtain:

$$2(\pi - \theta) = \|(R_{32} - R_{23}, R_{13} - R_{31}, R_{21} - R_{12})\|$$

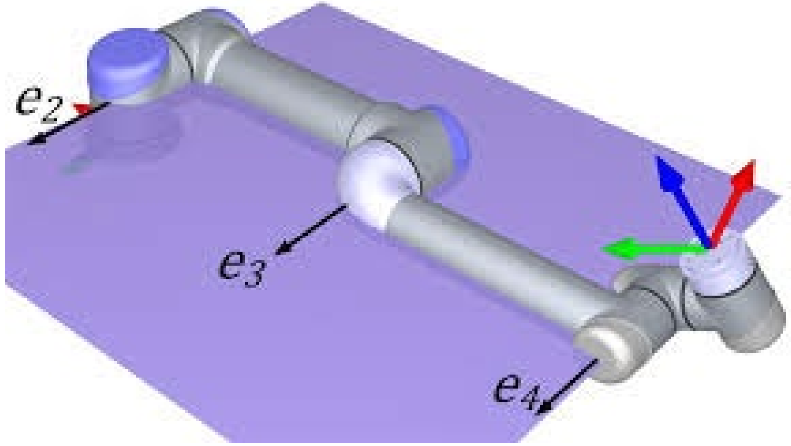
or

$$\theta = \pi - \frac{\|(R_{32} - R_{23}, R_{13} - R_{31}, R_{21} - R_{12})\|}{2}$$

Joint limits and singularities

Definition

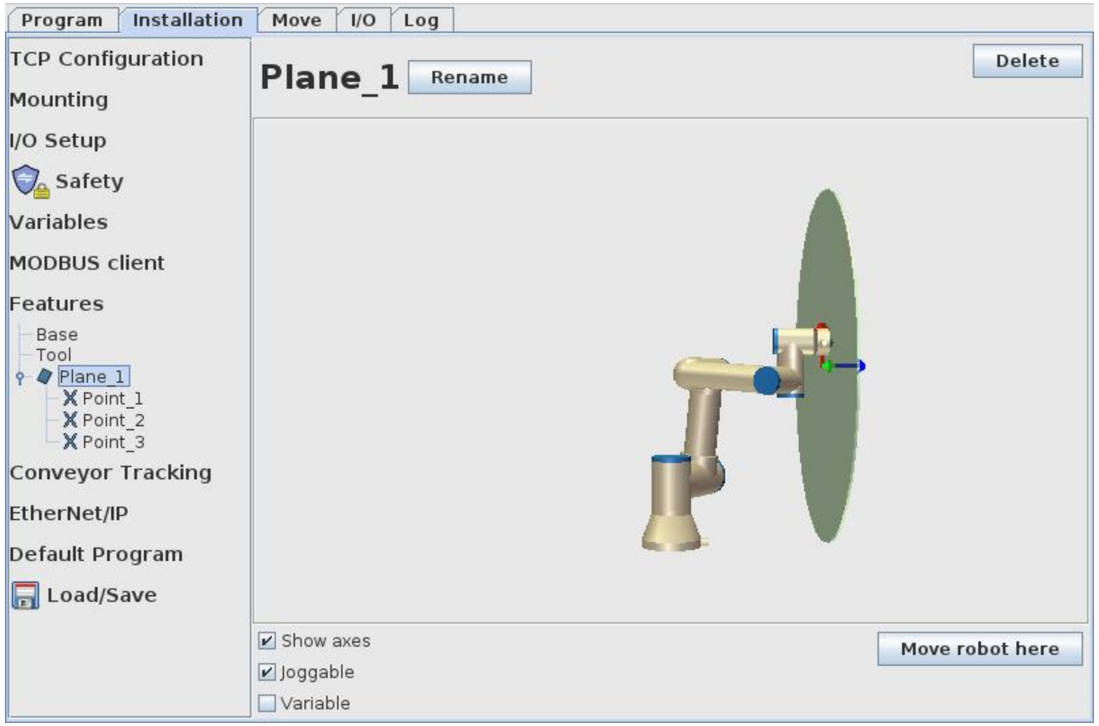
- The workspace of robots is limited by **singularities** and **joint limits**.
- When robots are controlled by task space control algorithms, it is necessary to handle singularities and joint limits during the execution of tasks.

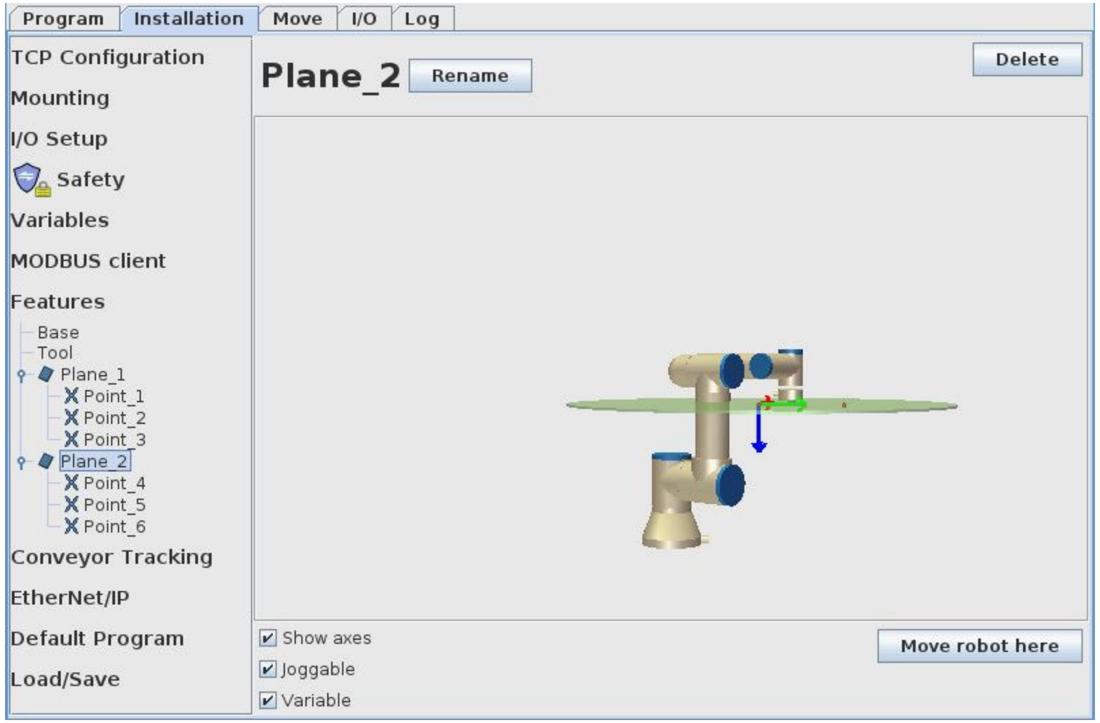


Joint limits

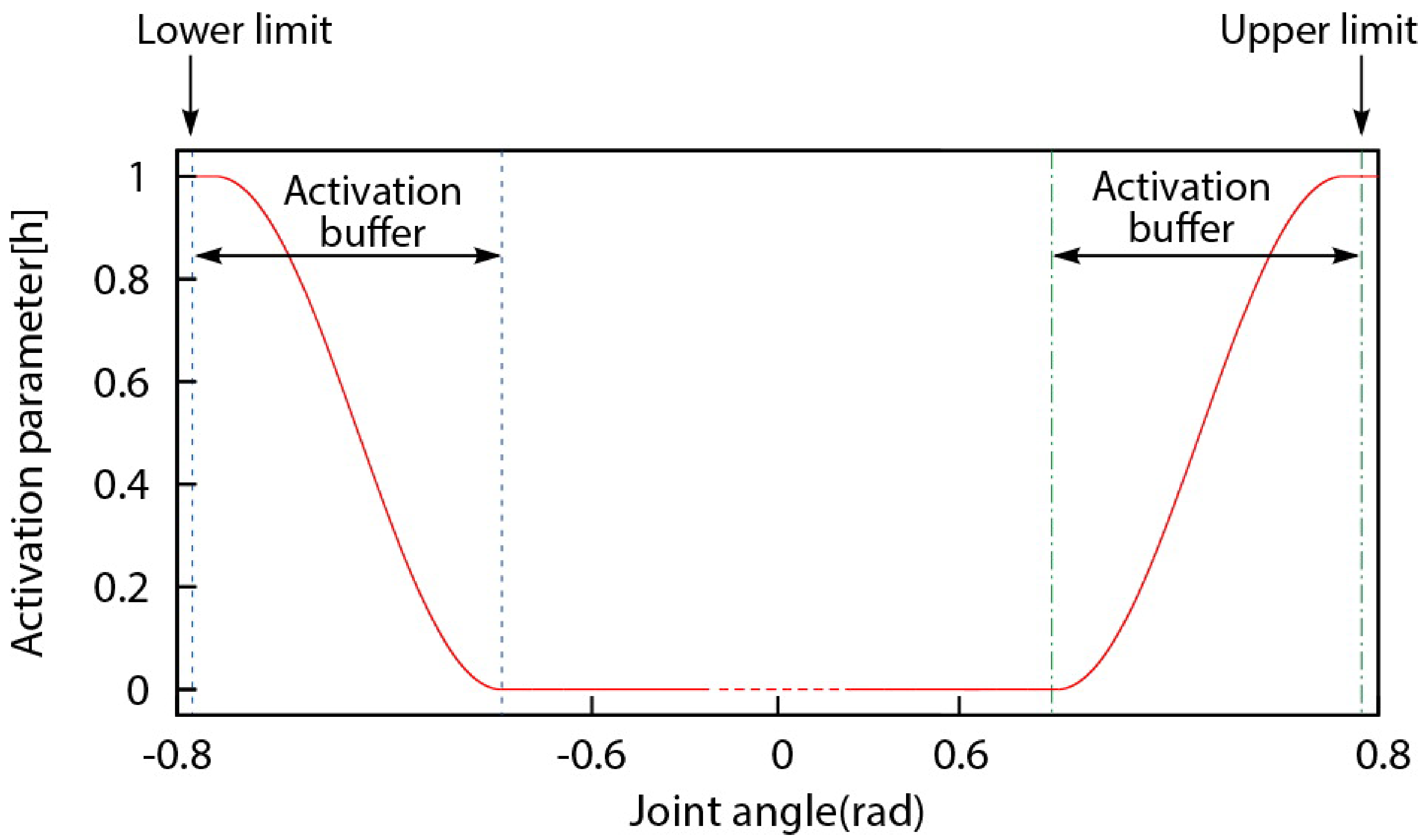
- Joint limits represent the hard limit of the robot's workspace.
- There are two types:
  - hardware joint limits (end switches),
  - software joint limits (definitions in the robot kinematics).
- Hardware joint limits are specific to every robot design and cannot be overcome.
- Software joint limits can be user defined, to prevent movement in a certain area of the workspace or safety zones.

Safety zone example





Detection of joint limits



Singularities

A robot singularity is a configuration in which the robot end-effector becomes blocked in certain directions.

At a singularity, a robotic arm loses one or more degrees of freedom.

Jacobian

- The time derivative of the kinematics equations yields the Jacobian of the robot, which relates the joint angle velocities to the linear and angular velocity of the end-effector.
- The principle of virtual work shows that the Jacobian also provides a relationship between joint torques and the resultant force and torque applied by the end-effector.

Singularities

- A robot singularity is a physical blockage, not some kind of abstract mathematical problem, although we have a simple mathematical explanation for it.
- For a six-axis robot arm the problem of singularities can be explained with the following inverse velocity kinematic equation

$$\boldsymbol{v_e} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}, \text{where } \boldsymbol{v_e} = [\dot{\boldsymbol{p}}_{\boldsymbol{e}}^{\boldsymbol{T}}, \dot{\boldsymbol{\omega}}_{\boldsymbol{e}}^{\boldsymbol{T}}]^{\boldsymbol{T}}$$

where the Jacobian matrix is rank deficient $det(\boldsymbol{J}) \approx 0$ and finding joint velocities for a certain Cartesian velocity vector $\boldsymbol{v}$ becomes impossible.

Singularities

- Singularities represent configurations at which mobility of the structure is reduced, i.e., it is not possible to impose an arbitrary motion to the end-effector.
- When the structure is at a singularity, infinite solutions to the inverse kinematics problem may exist.
- In the neighborhood of a singularity, small velocities in the operational space may cause large velocities in the joint space.

Singularities can be classified into:

- **Boundary singularities** that occur when the manipulator is either outstretched or retracted. It may be understood that these singularities do not represent a true drawback, since they can be avoided on condition that the manipulator is not driven to the boundaries of its reachable workspace.
- **Internal singularities** that occur inside the reachable workspace and are generally caused by the alignment of two or more axes of motion, or else by the attainment of particular end-effector configurations. Unlike the above, these singularities constitute a serious problem, as they can be encountered anywhere in the reachable workspace for a planned path in the operational space.

Problems

The problem with singularities is not only the impossibility of crossing them, but also the high joint velocities resulting from passing close to them.

- A robot is said to be close to singularity when the determinant of its Jacobian matrix is **close to zero**, which yields the effect of division by a very small number.
- Such high joint velocities may be unexpected and can pose **safety risks** in the case of big, fast industrial robots.
- Furthermore, when following a specific Cartesian path and passing close to a singularity, the feasible end-effector velocities are significantly reduced.
- Finally, due to control problems, the path accuracy of a robot controlled in Cartesian space deteriorates significantly in the vicinity of singularities.

Common singularities for a 6 axis robot

https://youtu.be/lD2HQcxeNoA

Singularities

Computation of internal singularities via the Jacobian determinant may be tedious and of no easy solution for complex structures. For manipulators having a **spherical wrist**, by analogy with what has already been seen for inverse kinematics, it is possible to split the problem of singularity computation into two separate problems:

- computation of *arm singularities* resulting from the motion of the first 3 or more links,
- computation of *wrist singularities* resulting from the motion of the wrist joints.

Singularities

- Let's consider a case n=6; the Jacobian can be partitioned into (3x3) blocks

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}_{11} & \boldsymbol{J}_{12} \\ \boldsymbol{J}_{21} & \boldsymbol{J}_{22} \end{bmatrix}$$

- and since the outer 3 joints are revolute the expression of the two right blocks is:

$$\boldsymbol{J}_{12} = \begin{bmatrix} \boldsymbol{z}_3 \times (\boldsymbol{p}_e - \boldsymbol{p}_3) & \boldsymbol{z}_4 \times (\boldsymbol{p}_e - \boldsymbol{p}_4) & \boldsymbol{z}_5 \times (\boldsymbol{p}_e - \boldsymbol{p}_5) \end{bmatrix}$$

$$\boldsymbol{J}_{22} = \begin{bmatrix} \boldsymbol{z}_3 & \boldsymbol{z}_4 & \boldsymbol{z}_5 \end{bmatrix}.$$

quick re-cap

$\boldsymbol{z}_{i-1}$ is given by the third column of the rotation matrix $\boldsymbol{R}_{i-1}^{0}$, i.e.,

$$\boldsymbol{z}_{i-1} = \boldsymbol{R}_1^0\,(q1)\ldots\boldsymbol{R}_{i-1}^{i-2}(q_{i-1})\boldsymbol{z}_0$$

where $\boldsymbol{z}_0 = [0\ 0\ 1]^T$ allows the selection of the third column.

$\boldsymbol{p}_e$ is given by the first three elements of the fourth column of the transformation matrix $\boldsymbol{T}_e^0$, i.e., by expressing $\widetilde{\boldsymbol{p}}_e$ in the (4 × 1) homogeneous form

$$\widetilde{\boldsymbol{p}}_e = \boldsymbol{A}_1^0\,(q1)\ldots\boldsymbol{A}_n^{n-1}\,(q_n)\widetilde{\boldsymbol{p}}_0$$

where $\widetilde{\boldsymbol{p}}_e = [0\ 0\ 0\ 1]^T$ allows the selection of the fourth column.

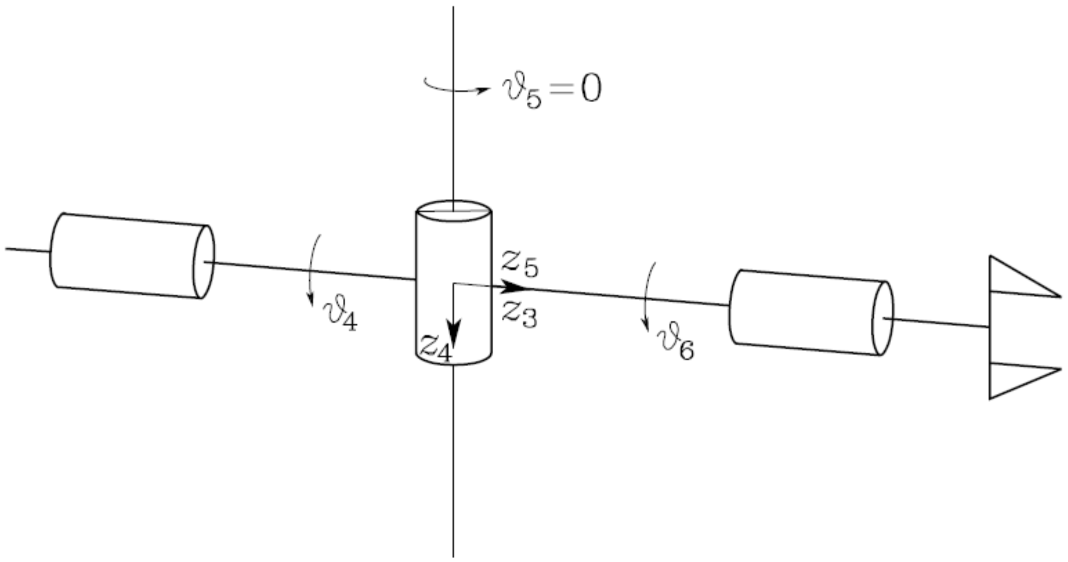
$\boldsymbol{p}_{i-1}$ is given by the first three elements of the fourth column of the transformation matrix $\boldsymbol{T}_{i-1}^0$, i.e., by expressing $\widetilde{\boldsymbol{p}}_e$ in the (4 × 1) homogeneous form

$$\widetilde{\boldsymbol{p}}_{i-1} = \boldsymbol{A}_1^0\,(q1)\ldots\boldsymbol{A}_{n-1}^{n-2}\,(q_{i-1})\widetilde{\boldsymbol{p}}_0.$$

Singularities

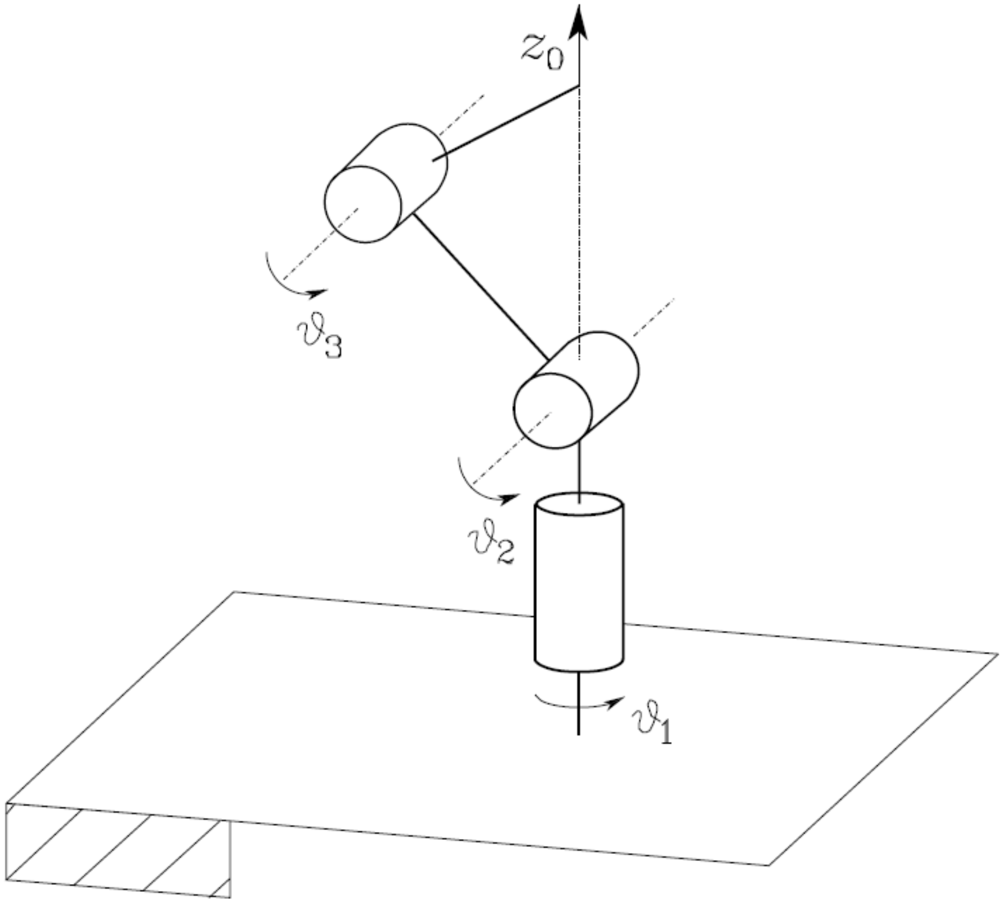
- As singularities are typical of the mechanical structure and do not depend on the frames chosen to describe kinematics, it is convenient to choose the origin of the end-effector frame at the intersection of the wrist axes and the Jacobian becomes:
$$\boldsymbol{J_{12}} = [0\ 0\ 0\ ]$$
- In this case, computation of the determinant is greatly simplified, as this is given by the product of the determinants of the two blocks on the diagonal, i.e.,
$$det(\boldsymbol{J}) = det(\boldsymbol{J_{11}})det(\boldsymbol{J_{22}})$$
- In turn, because of the singularity decoupling, the condition $det(\boldsymbol{J_{11}}) = 0$, determines the *arm singularity*,
- and $det(\boldsymbol{J_{22}}) = 0$ determines the *wrist singularity*.

Wrist singularity



- The wrist singularity occurs when vectors z_3, z_4, z_5 are linearly dependent e.g. z_3 and z_5 are aligned $\vartheta_5 = 0$ and $\vartheta_5 = \pi$.
- This can be determined by inspecting $\boldsymbol{J}_{22}$ of the Jacobian.
- Taking into consideration only the first configuration , the loss of mobility is caused by the fact that rotations of equal magnitude about opposite directions on ϑ_4 and $\vartheta_6$do not produce any end-effector rotation.
- Further, the wrist is not allowed to rotate about the axis orthogonal to z_4 and z_3.

Arm singularity

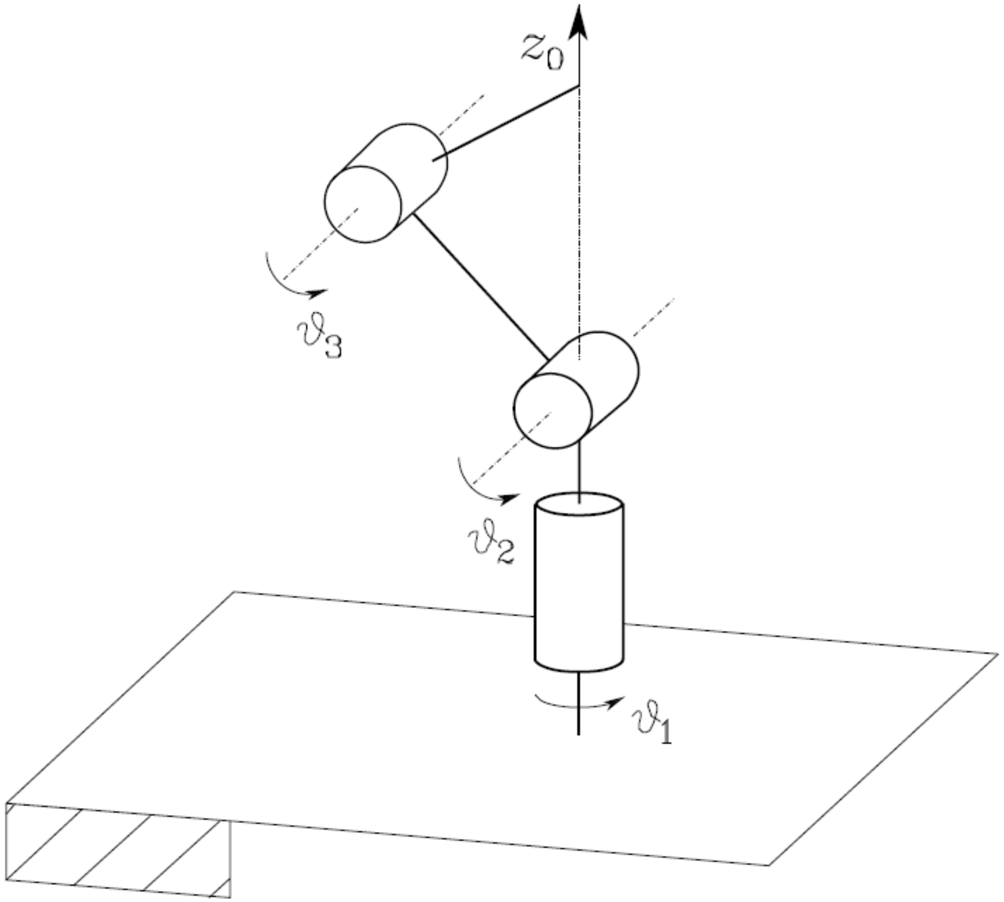


- let's consider an autapomorphic manipulator with a Jacobian,

$$\boldsymbol{J} = \begin{bmatrix} -s_1(a_2c_2 + a_3c_{23}) & -c_1(a_2s_2 + a_3s_{23}) & -a_3c_1s_{23} \\ c_1(a_2c_2 + a_3c_{23}) & -s_1(a_2s_2 + a_3s_{23}) & -a_3s_1s_{23} \\ 0 & a_2c_2 + a_3c_{23} & a_3c_{23} \\ 0 & s_1 & s_1 \\ 0 & -c_1 & -c_1 \\ 1 & 0 & 0 \end{bmatrix}$$

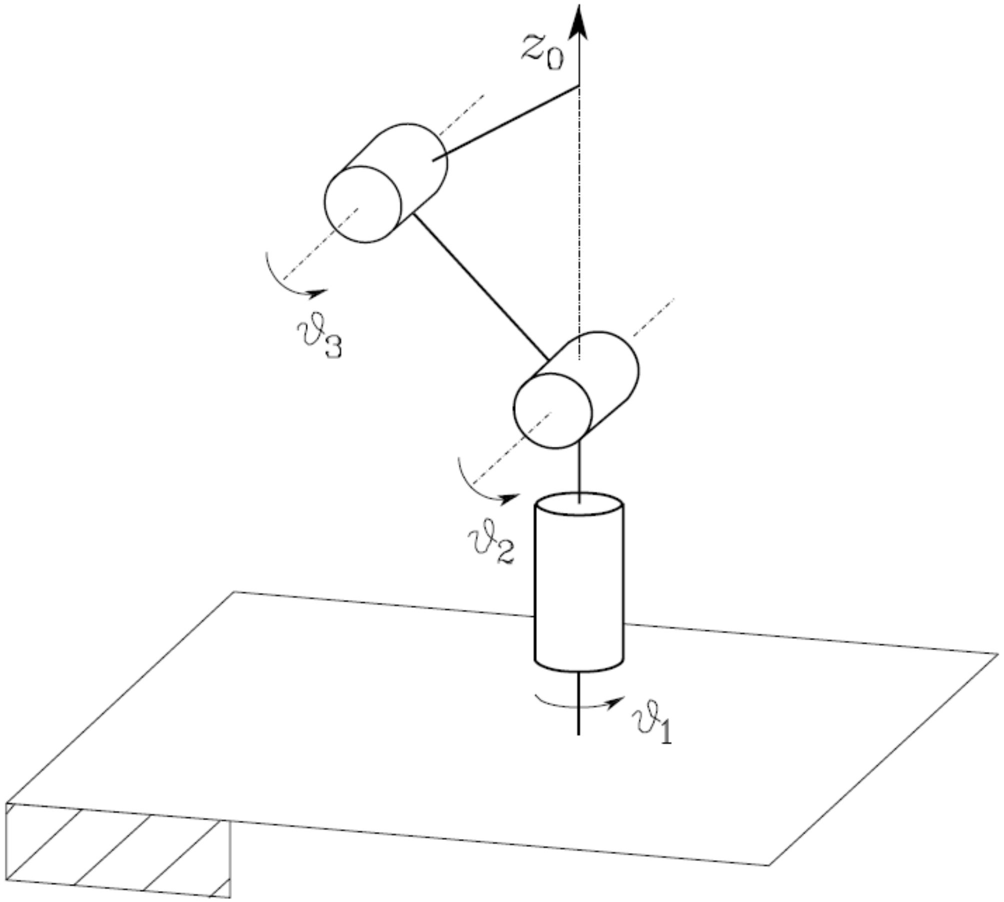
- We are considering a manipulator with 3 DOF therefore the upper half ([3x3]) of the J matrix is relevant.

Arm singularity



- The determinant is:
$$\det(\boldsymbol{J}_P) = -a_2 a_3 s_3 (a_2 c_2 + a_3 c_{23}).$$
- For $a_2 a_3 \neq 0$ and the det vanishes if $s_3 = 0$ and/or $(a_2 c_2 + a_3 c_{23}) = 0$. The first situation occurs when $\vartheta_3 = 0$, $\vartheta_3 = \pi$ the elbow is outstretched or retracted outlining a *elbow singularity* .
- The shoulder singularity occurs when a wrist point lies on axis z_0 → $p_x = p_y = 0$.
- The entire axis z_0 describes a continuum of singular configurations; a rotation of ϑ_1 does not cause any translation of the wrist position (the first column of $\boldsymbol{J}_P$ is always null at a shoulder singularity), and then the inverse kinematics equation admits infinite solutions.

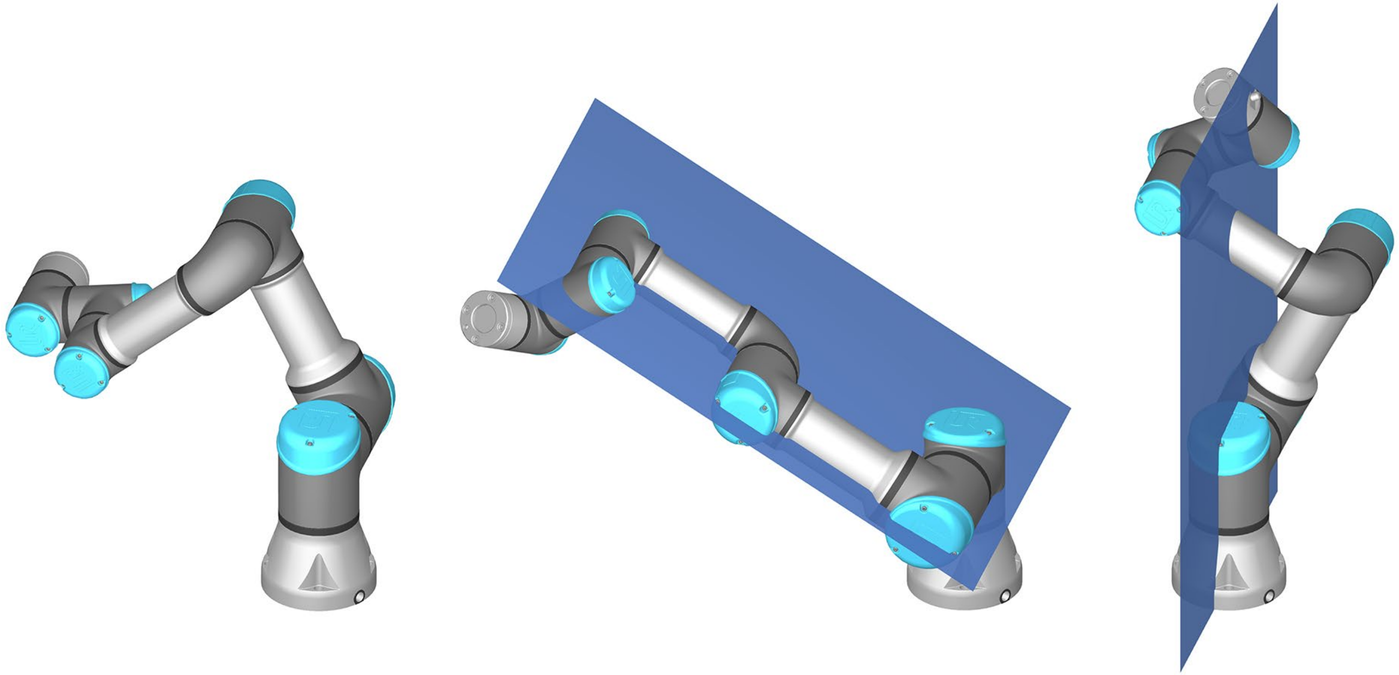
Arm singularity



- Unlike the wrist singularities, the arm singularities are well identified in the operational space, and thus they can be suitably avoided in the end-effector trajectory planning stage.

Collaborative robot UR5

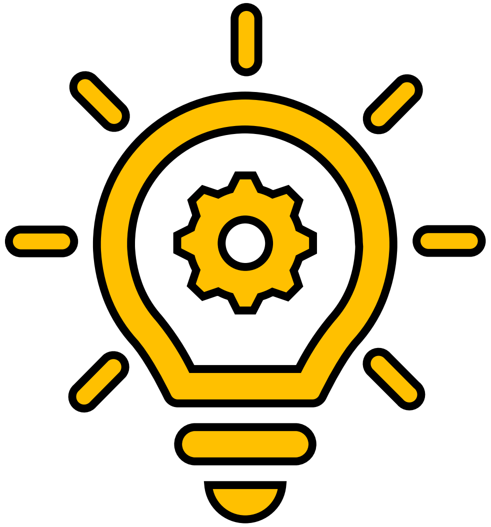
- Wrist singularity (left), joints 4 and 6 are parallel,
- Elbow singularity (middle), joints 2,3 and 4 are coplanar.
- Shoulder singularity (right), occurs when the intersection axes of joint 5 and 6 lies in the plane passing through the axes of joints 1 and joints 2.



How do we deal with singularities and joint limits?

1 . solution: is to detect them and avoid them when planning robot movements.

2. solution: include the specifications in the Jacobian definition.



The Jacobian method

In the equation
$$\boldsymbol{J}(\boldsymbol{q})\boldsymbol{\Delta q} = \boldsymbol{\Delta u}$$
embed the following expression,

$$\mathrm{q}_{\min} \leq q + \Delta q \leq q_{max}$$
and the result is a optimization problem

$$min||\boldsymbol{J}(\boldsymbol{q})\boldsymbol{\Delta q} - \boldsymbol{\Delta u}||$$
This is a quadratic Programming Problem and can be solved with various algorithms (*Interior Point Method described in Nocedal and Wright: Numerical Optimization, pp. 480-485*).

The Jacobian method

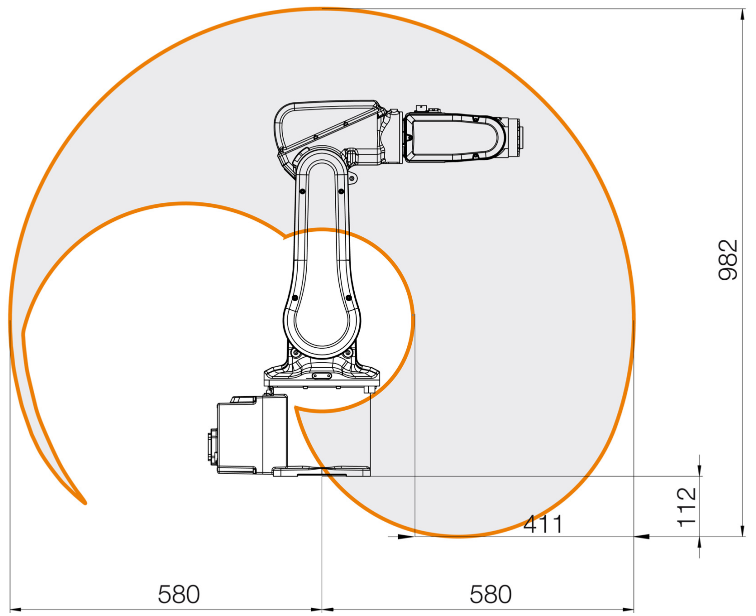
In cases where singularities and joint limits causes no problems, we just obtain the solution to $\boldsymbol{J}(\boldsymbol{q})\boldsymbol{\Delta q} = \boldsymbol{\Delta u}$ but otherwise we obtain a sensible solution as close to the desired displacement as possible.

Example

Test singularities

- Test and visualize if the given joint poses are in singularity.
- Integrate the provided python script with the example from last time calculate the jacobian and visualize the joint configuration with the provided example code from last time.

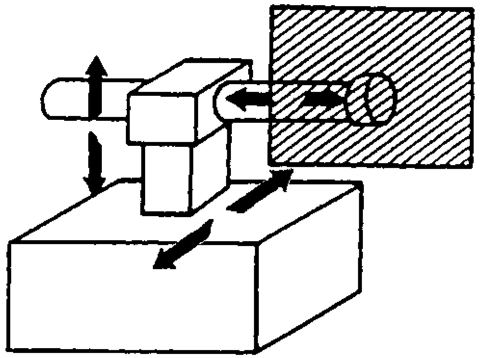
Robot workspace



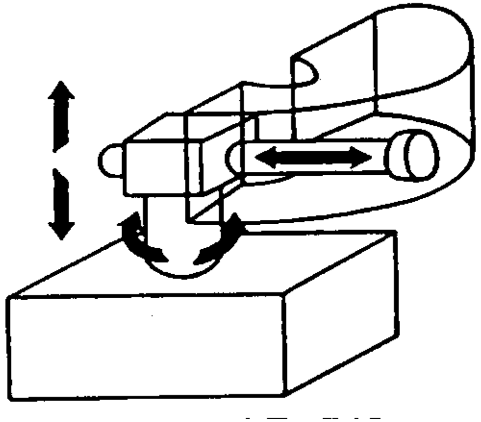
Workspace

- The robot workspace consists of all points that can be reached by the robot end-point.
- It plays an important role when selecting an industrial manipulator for an desired task.

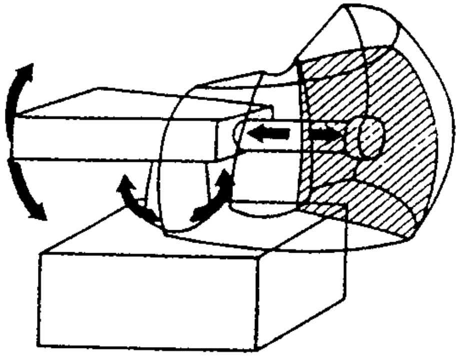
Workspace based on the configuration type



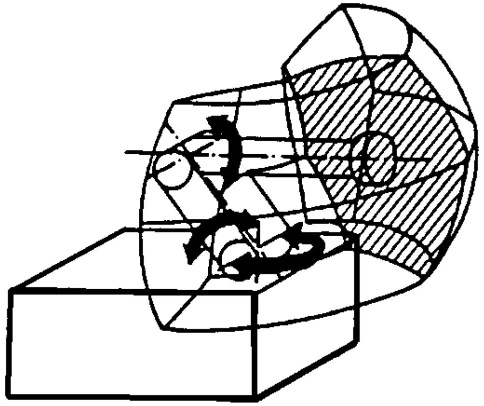
Rectangular



Cylindrical



Spherical



Revolute

Activity

In pairs find the workspace representation of the following robots discuss the difference between them,

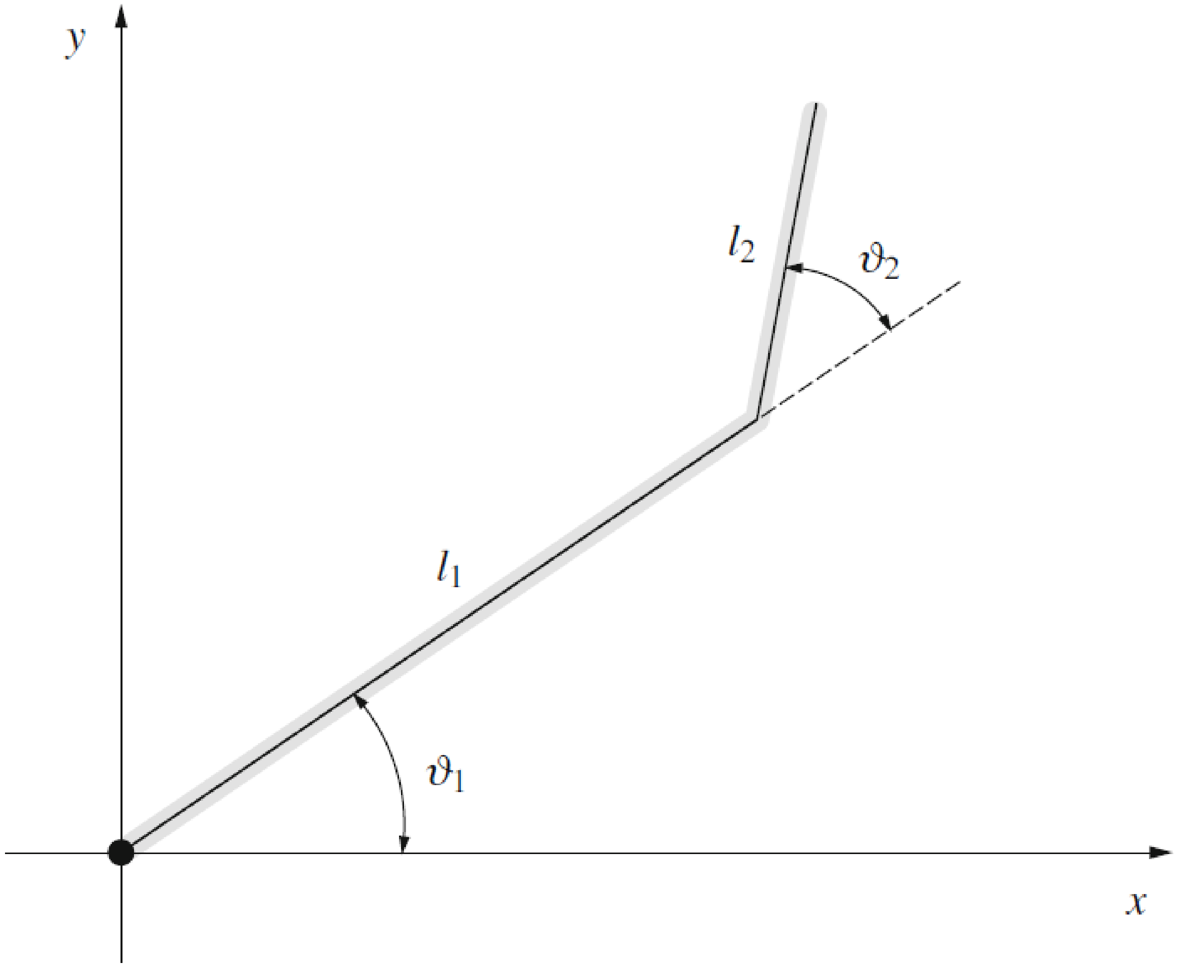
- UR5
- UR10
- KUKA iiwa
- Fanuc S-900iB
- IRB 2600

Let's derive the workspace representation for a 2 segment manipulator

- The rotational degrees of freedom are presented as ϑ_1 and ϑ_2 with link lengths presented as l_1 and l_2.
- The coordinates of the end point can be represented as:

$$x = l_1 \cos \vartheta_1 + l_2 \cos(\vartheta_1 + \vartheta_2)$$

$$y = l_1 \sin \vartheta_1 + l_2 \sin(\vartheta_1 + \vartheta_2).$$



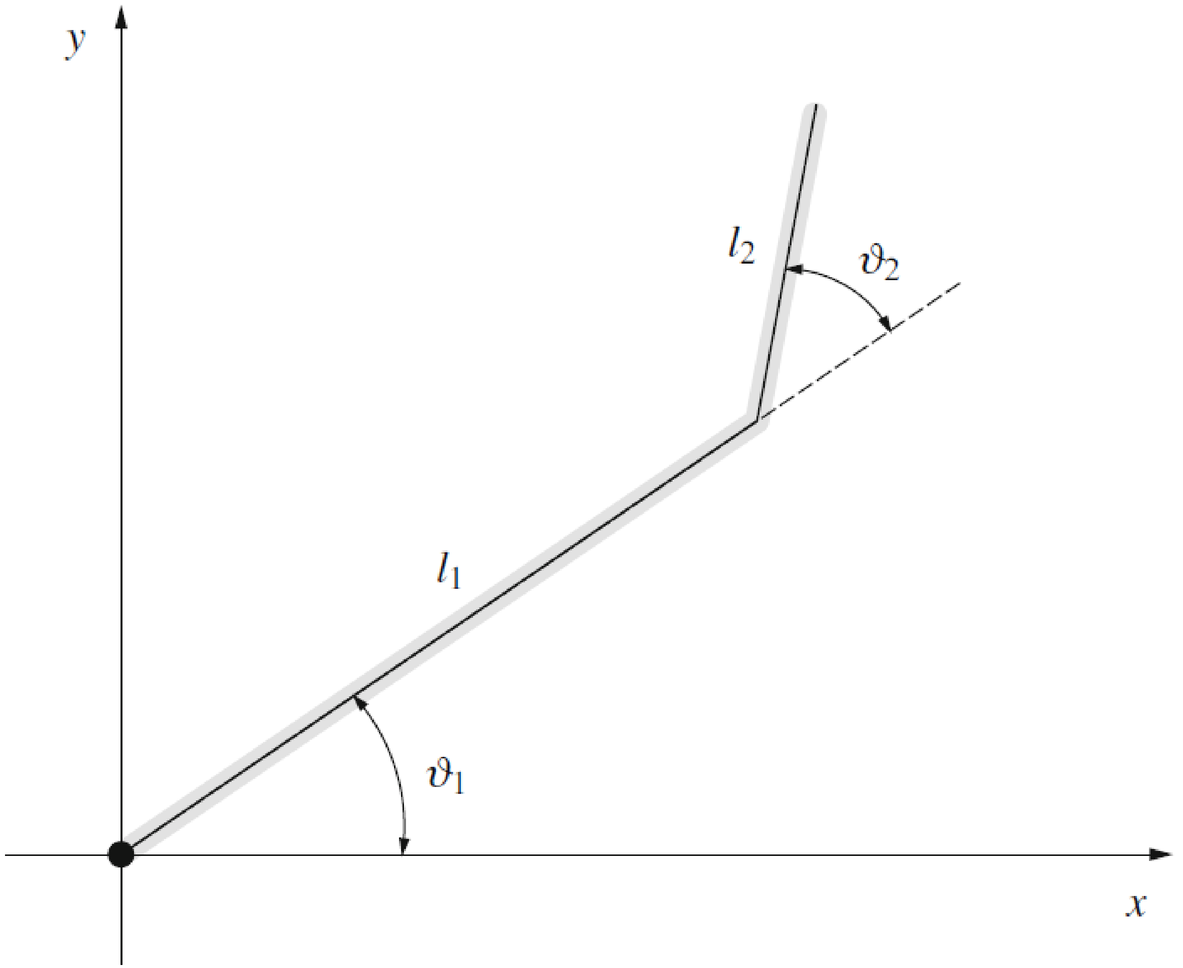
Workspace of a 2 segment manipulator

- Rearranging the equations, we get an equation representing a circle:

$$(x - l_1 \cos \vartheta_1)^2 + (y - l_1 \sin \vartheta_1)^2 = l_2^2$$

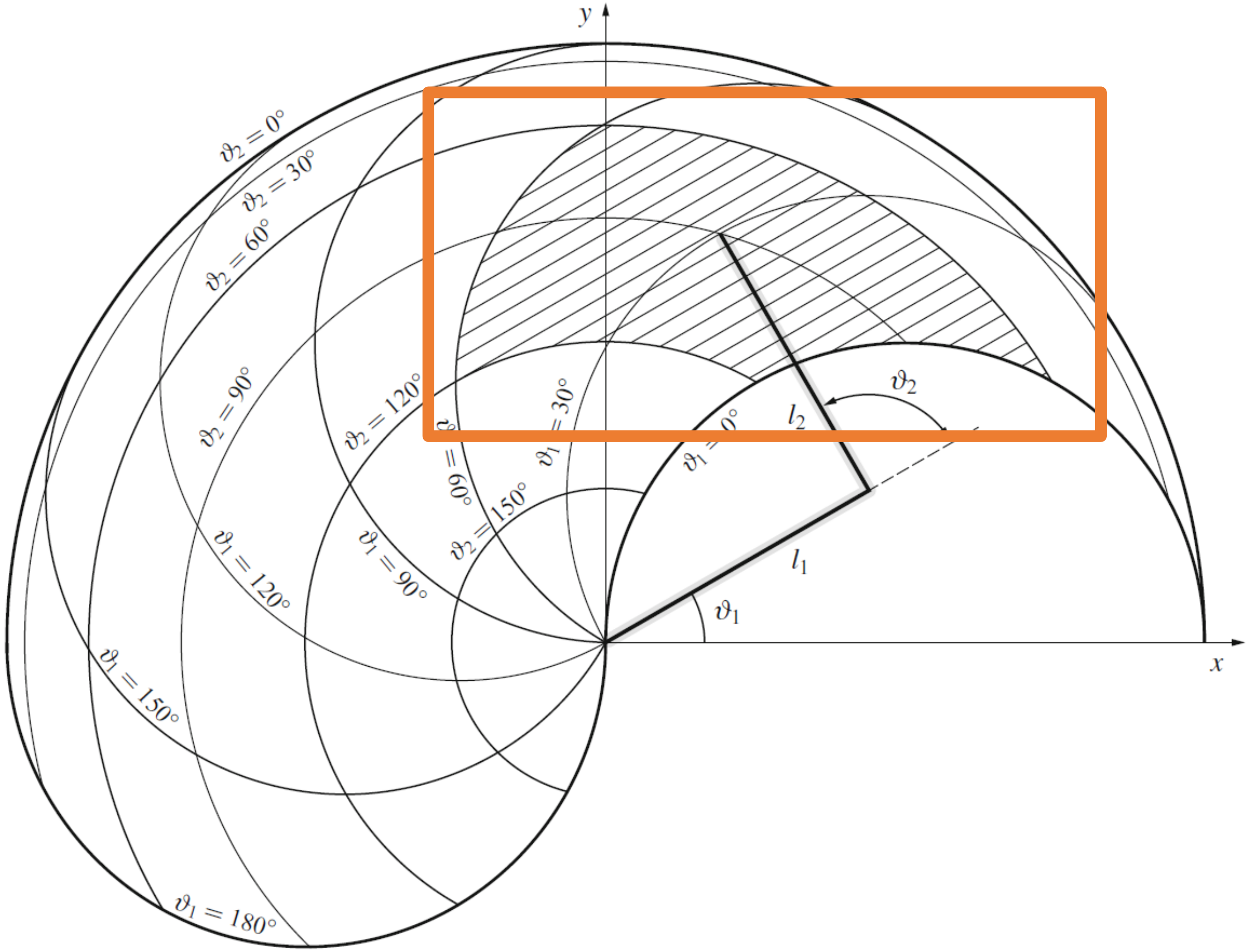
$$x^2 + y^2 = l_1^2 + l_2^2 + 2 l_1 l_2 \cos \vartheta_2.$$

- When we plot some solutions for some arbitrary angles we get.



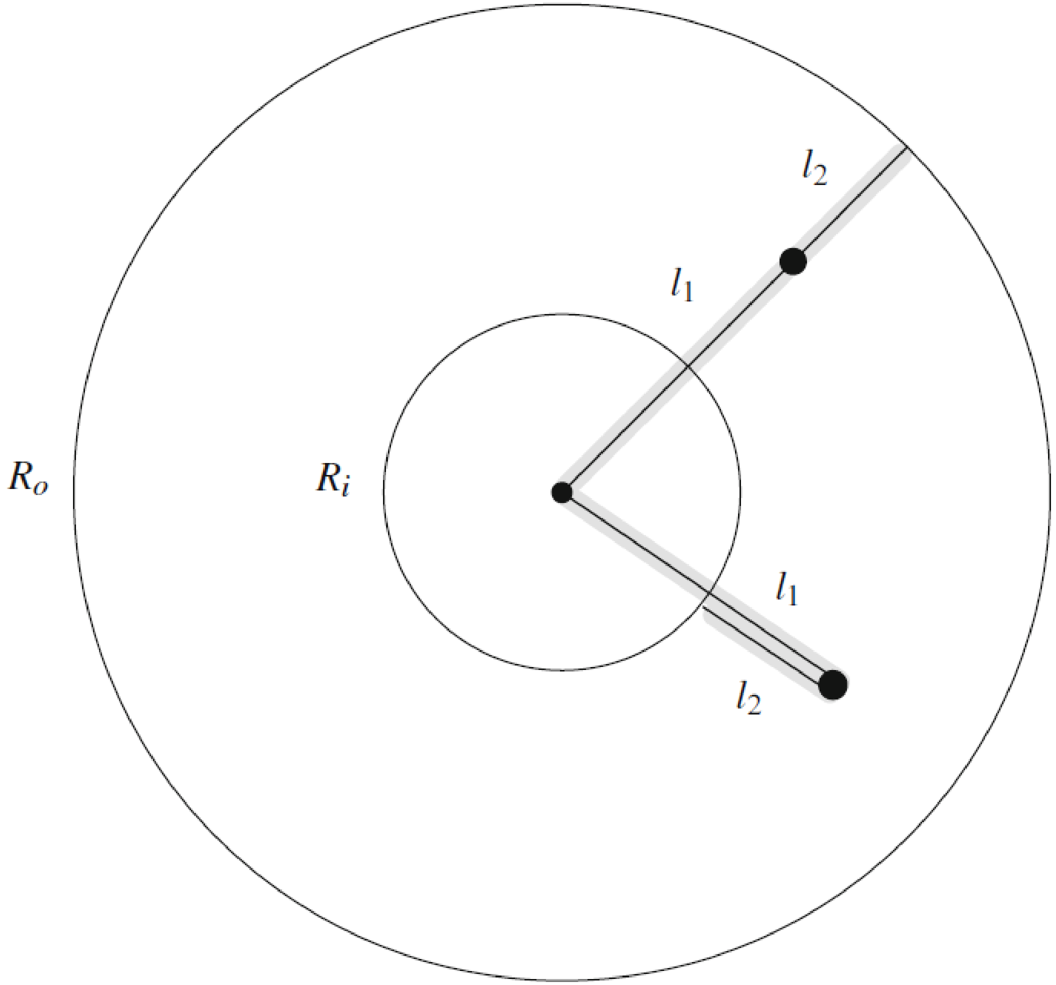
Workspace of a 2 segment manipulator

- The representation depends on both lengths of the links and angles between them.
- A more representable workspace is (e.g. for a SCARA manipulator), when we constrain the movement.



Workspace of a 2 segment manipulator

- In the previous example the two links were of the same lengths $l_1 = l_2$,
- lets consider a 2 segment manipulator where the links have different lengths $l_1 \neq l_2$.
- For a manipulator with this structure the workplace is comprised of two rings $R_i = l_1 - l_2$ (inner ring) and $R_o = l_1 + l_2$ (outer ring) R represents radious.



Workspace of a 2 segment manipulator

- The aim is find a ratio between l_1, l_2 so that it represents the largest working area.
- The work area can be represented as:

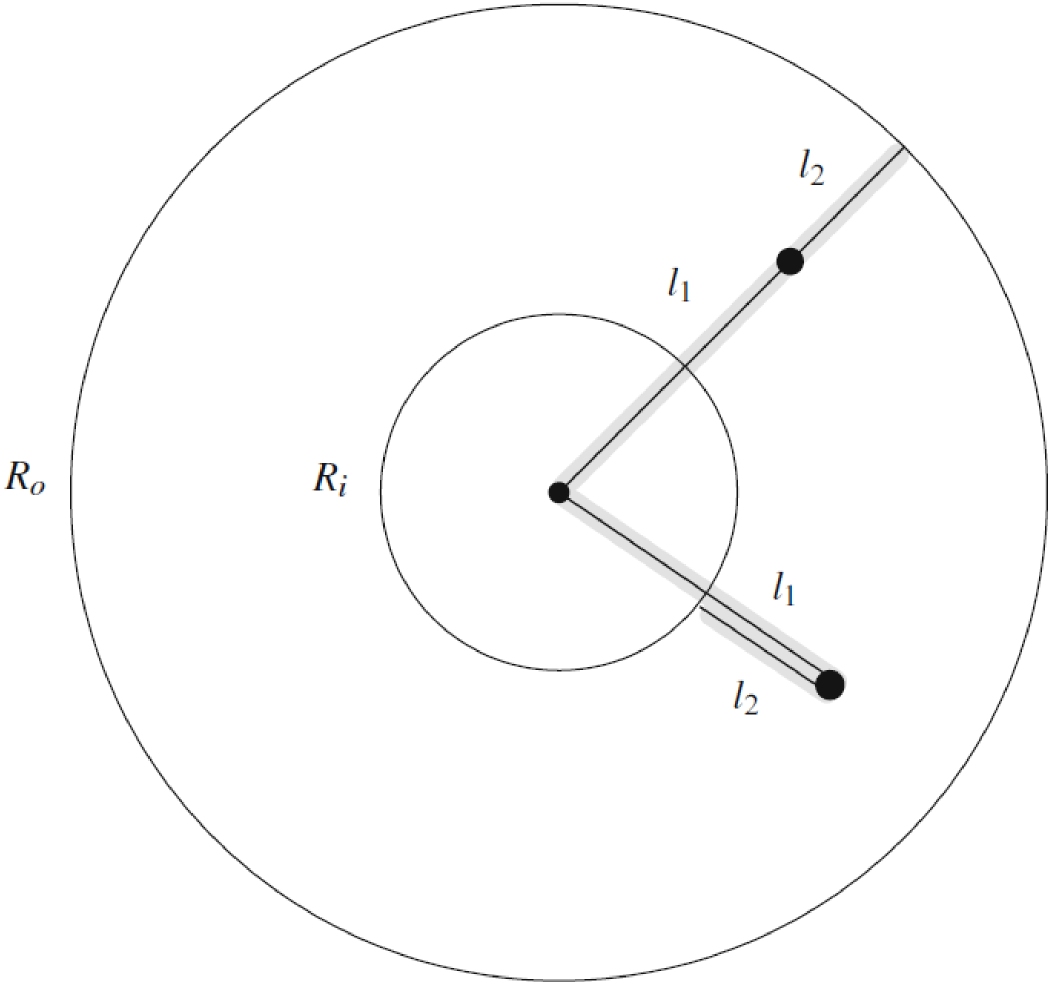
$$A = \pi R_o^2 - \pi R_i^2.$$

- by inserting the inner radius

$$R_i^2 = (l_1 - l_2)^2 = (2l_1 - R_o)^2$$

- and with rearranging we get

$$A = \pi R_o^2 - \pi (2l_1 - R_o)^2.$$



Workspace of a 2 segment manipulator

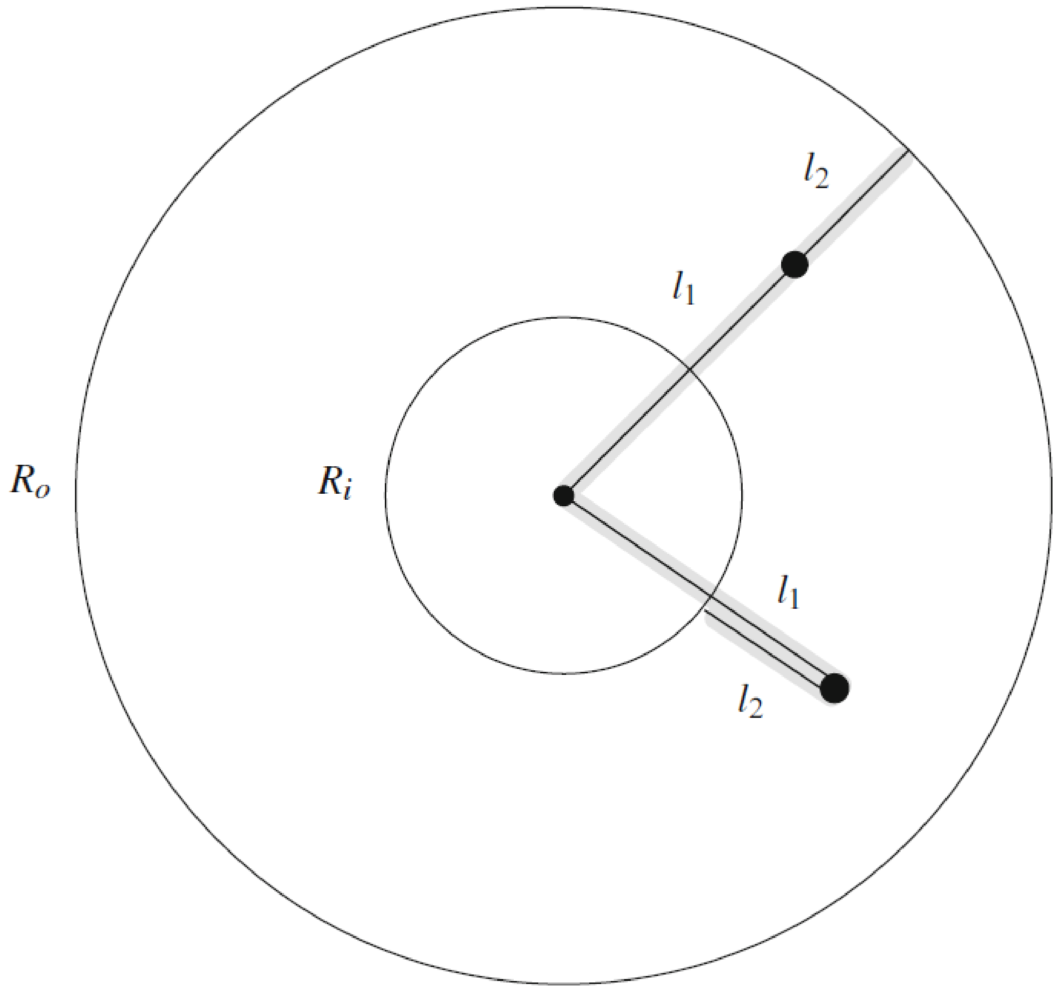
- To get the maximum area the derivative in respect to l_1 should be =0.

$$\frac{\partial A}{\partial l_1} = 2\pi(2l_1 - R_o) = 0$$

- and the solution is

$$l_1 = \frac{R_o}{2}$$

- given that $l_1 = l_2$.
- The largest working area of the two-segment mechanism occurs for equal lengths of both segments.

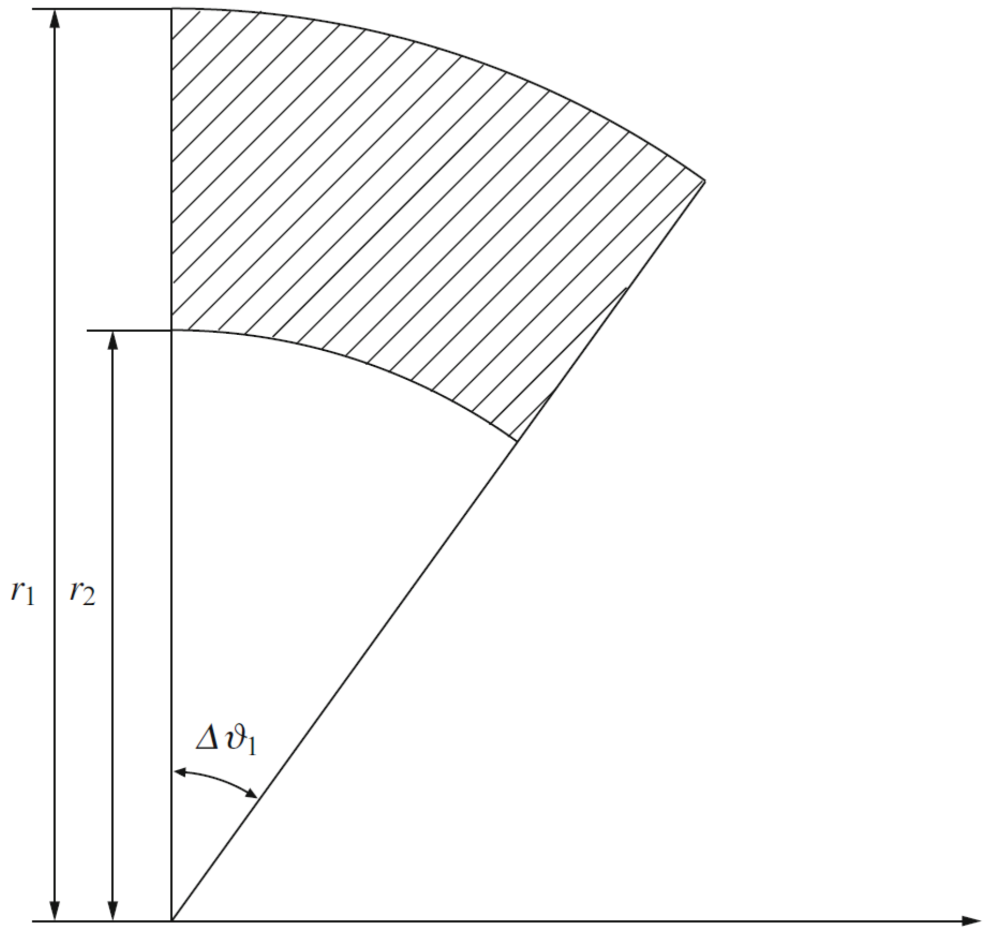


Workspace of a 2 segment manipulator

- The are depends on the lengths of the links and the min and max values of the angles.
- The working area refers to a segment of a ring.

$$\Delta\vartheta_1 = (\vartheta_{1_{max}} - \vartheta_{1_{min}})$$

$$A = \frac{\Delta\vartheta_1 \pi}{360}(r_1^2 - r_2^2)$$



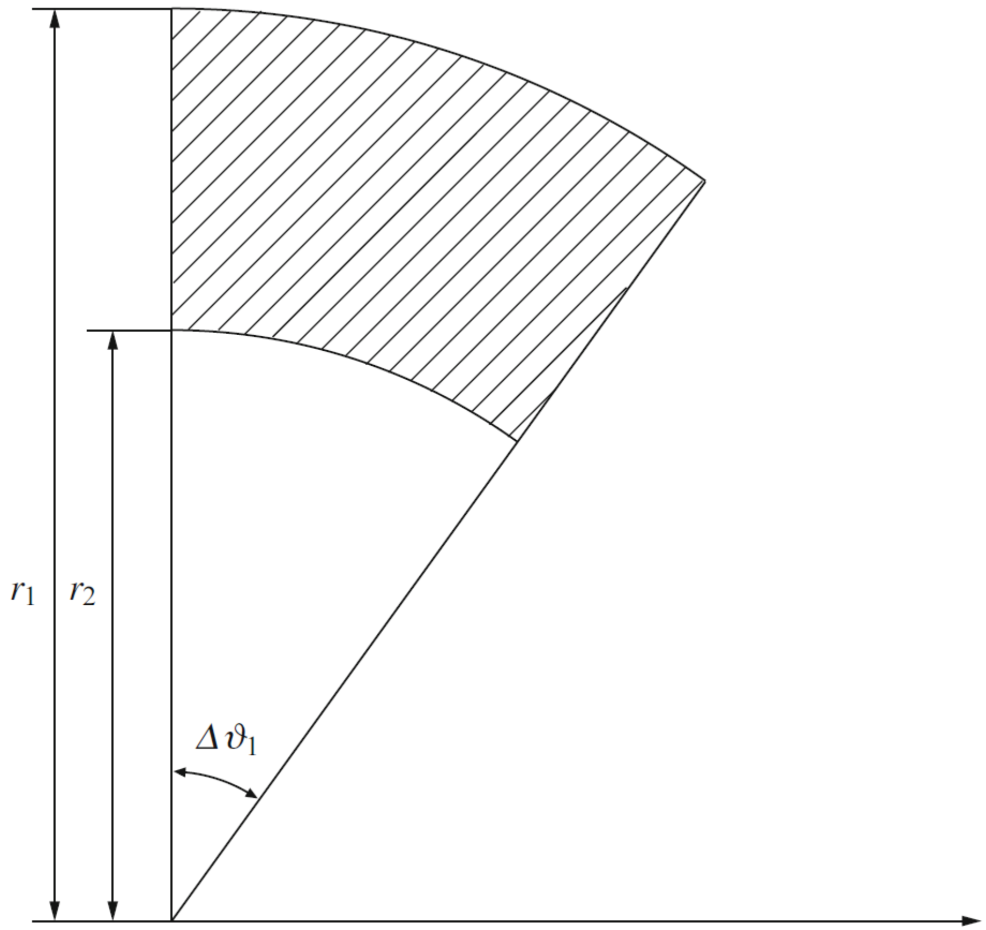
Workspace of a 2 segment manipulator

- The radios can be calculated as:

$$r_1 = \sqrt{l_1^2 + l_2^2 + 2l_1l_2\cos\vartheta_{2_{min}}}$$

$$r_2 = \sqrt{l_1^2 + l_2^2 + 2l_1l_2\cos\vartheta_{2_{max}}}$$

- The angle ϑ_1 determines the position of the working surface with respect to the reference frame and has no influence on its shape.



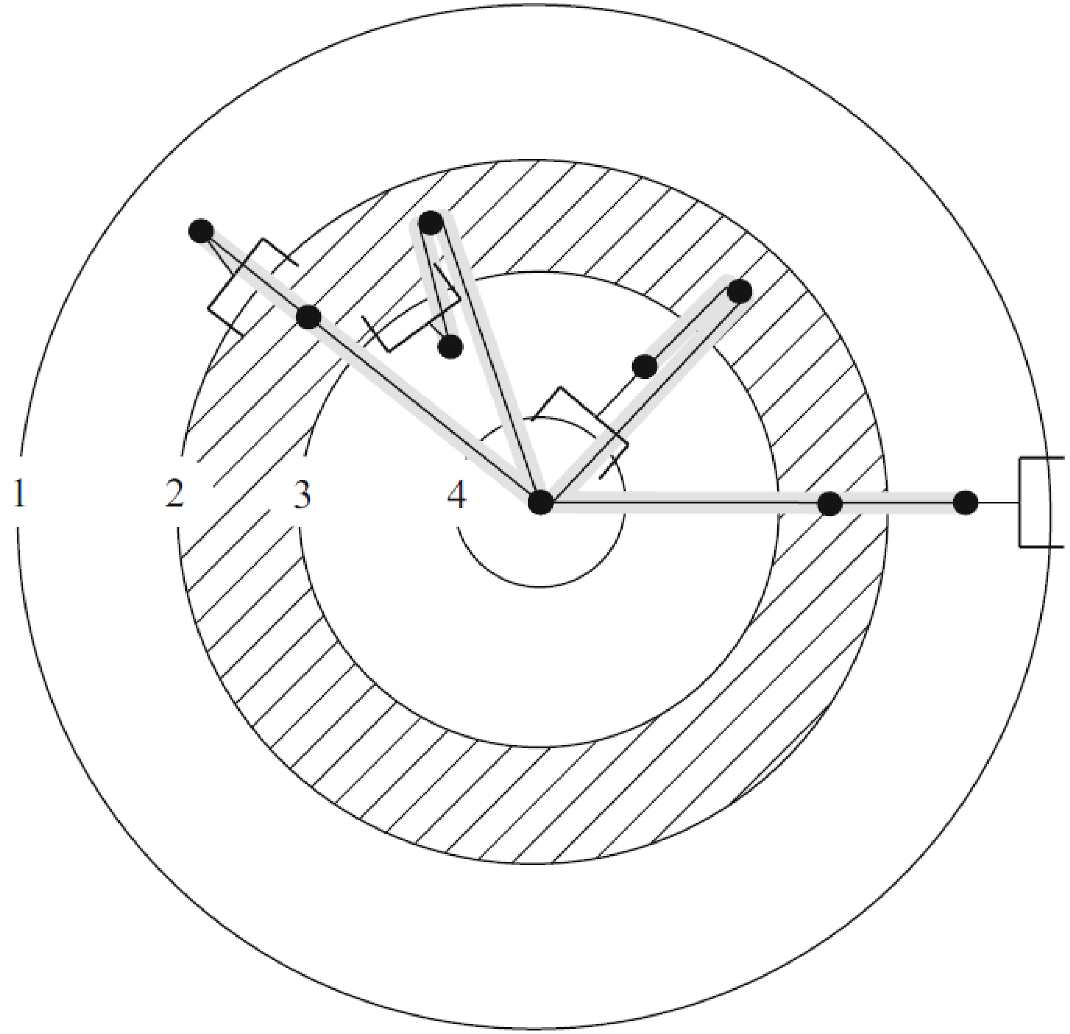
Workspace of a 2 segment manipulator

- The influence of the second angle ϑ_2 under different values we obtain a different shape of the work area.
- Until now, under the term workspace we were considering the so called reachable robot workspace. This includes all the points in the robot surroundings that can be reached by the robot end-point .
- Often this so-called ***dexterous workspace*** is of greater importance.

$$
\begin{array}{rl}
0^\circ \leq \vartheta_2 \leq 30^\circ & A = 0.07 \\
30^\circ \leq \vartheta_2 \leq 60^\circ & A = 0.19 \\
60^\circ \leq \vartheta_2 \leq 90^\circ & A = 0.26 \\
90^\circ \leq \vartheta_2 \leq 120^\circ & A = 0.26 \\
120^\circ \leq \vartheta_2 \leq 150^\circ & A = 0.19 \\
150^\circ \leq \vartheta_2 \leq 180^\circ & A = 0.07.
\end{array}
$$

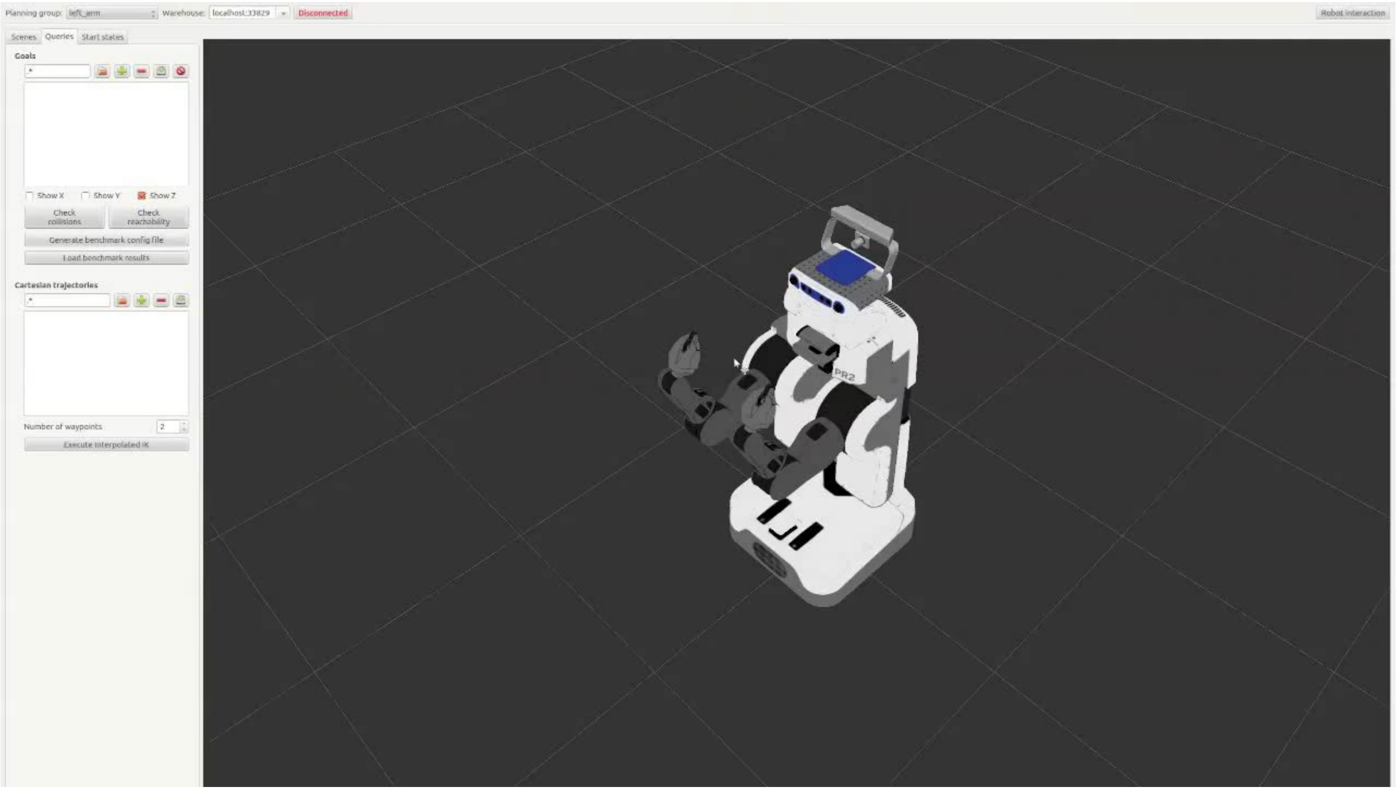
Workspace of a 2 segment manipulator

- The dexterous workspace comprises all the points that can be reached with any arbitrary orientation of the robot end-effector.
- This workspace is always smaller than the reachable workspace.
- The dexterous workspace is larger when the last segment (end-effector) is shorter.



Reachability

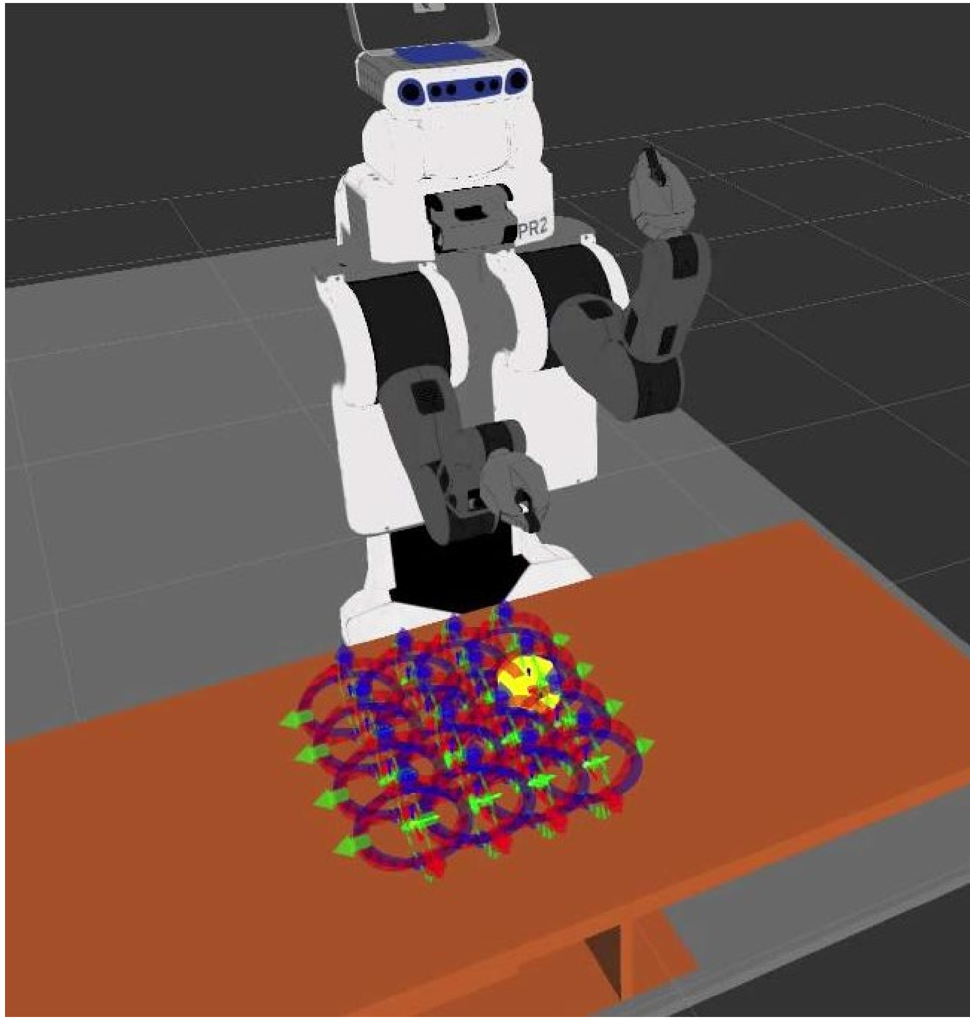
- The ability of a manipulator to reach a specific point in its workspace.
- Reachability is considered in:
  - Pick and Place task,
  - welding tasks…



https://www.youtube.com/watch?v=OglpJnElgu4

Reachability

- Iterative process to select the best position of the robot base for a specific task condition.
- The reachability of a desired pose is tested based on a number of collision free kinematic configuration that the robot can achieve.
- The outcome is the best position of the base in relation to the task space position of the workpiece.



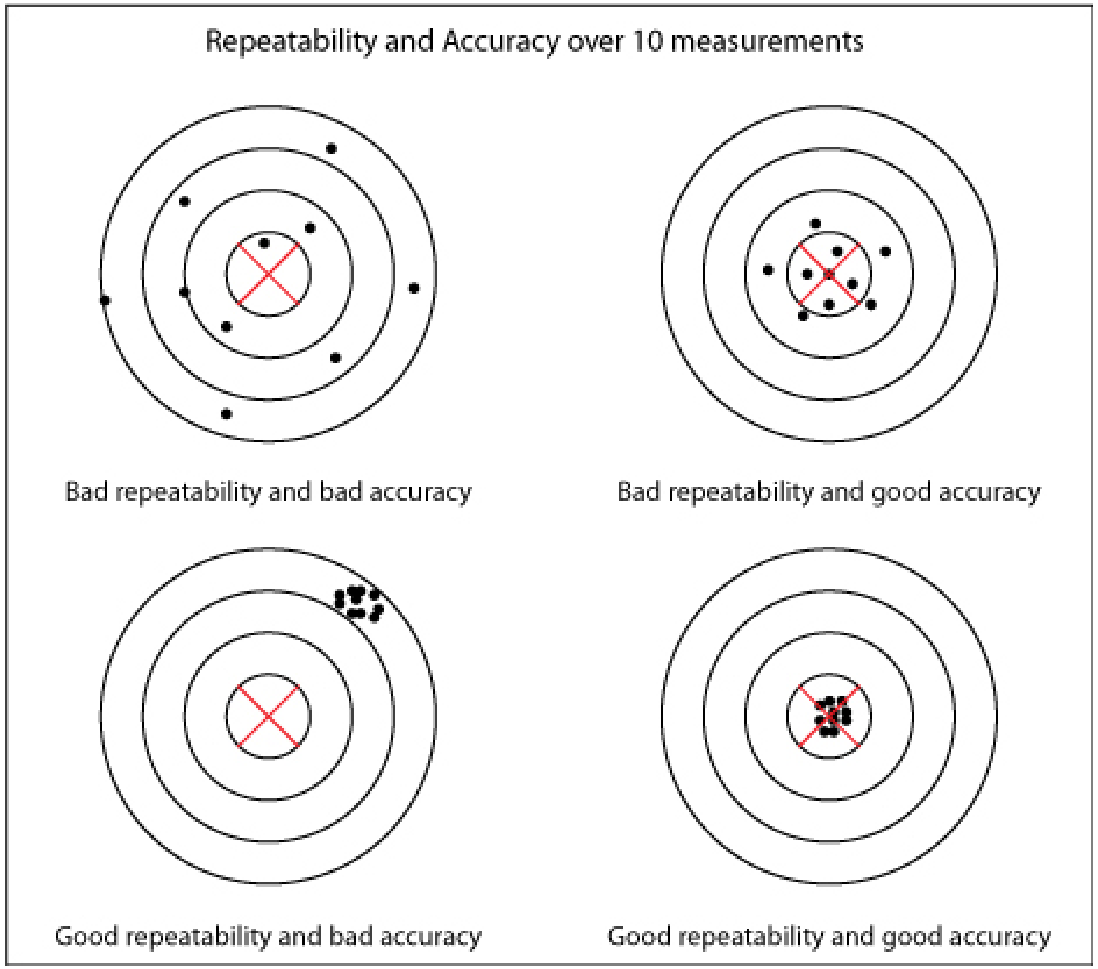
Accuracy

- If the mechanical dimensions of the structure differ from the corresponding DK implementation because of mechanical tolerances, a deviation arises between the position reached in the assigned posture and the position computed via direct kinematics.
- Such a deviation is defined as **accuracy**; this parameter attains typical values below one millimeter and depends on the structure as well as on manipulator dimensions.
- Accuracy varies with the end-effector position in the workspace.
- it is relevant when programming robots.

Repeatability & Accuracy

ISO 9283:1998 Norm for Industrial Robots:

- Repeatability: positional deviation from the average of displacement. (max speed and max payload).
- Accuracy: ability to position, at a desired target point within the work volume. (max speed and max payload).
- For a manipulator with a maximum reach of 1.5m, accuracy varies from 0.2 to 1mm in the workspace, while repeatability varies from 0.02 to 0.2mm.



Example

Workspace calculation

- Extend the workspace calculation to represent a 4 segment manipulator with the following parameters:
  - l1 = 2m
  - ang1 = $\mathrm{q}_{\min} = 0,°\ q_{max} = 130°$
  - l2 = 1.75m
  - ang2 = $\mathrm{q}_{\min} = 0,°\ q_{max} = 90°$
  - l3 = 1.25 m
  - ang3 = $\mathrm{q}_{\min} = 0,°\ q_{max} = 80°$
  - l4 = 0.75m
  - ang4 = $\mathrm{q}_{\min} = 0,°\ q_{max} = 40°$

Take home message

Singularities should be detected and avoided in the planning phase.

Robot’s workspace governs the task.

Reachability is the ability of a manipulator to reach a specific point in the workspace.

In the majority of the times we have to consider the dexterous workspace of the robot.