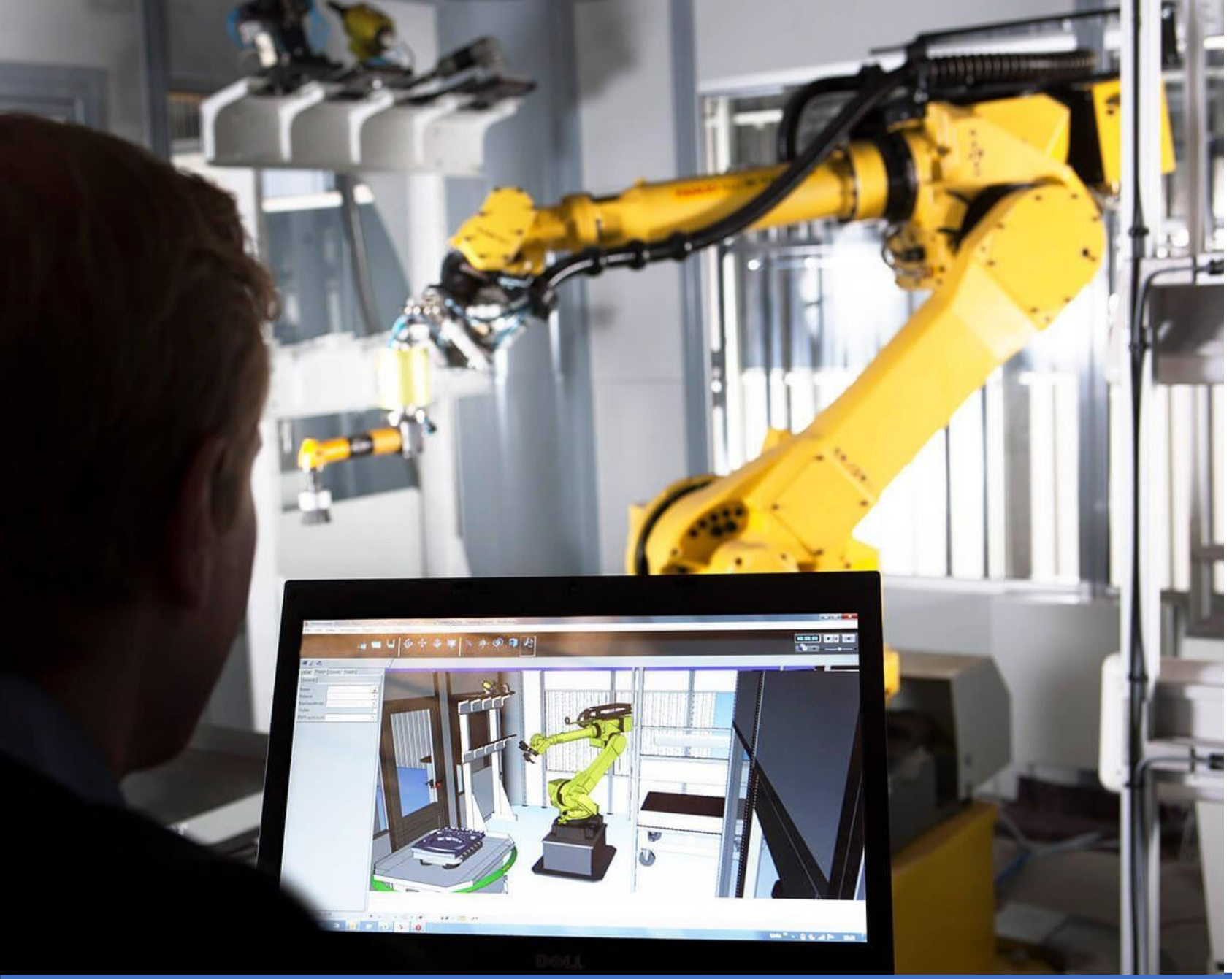
Module B – Programming robot motions

Robot Programming and Control
Academic Year 2021-2022

Diego Dall'Alba
diego.dallalba@univr.it
Department of Computer Science – University of Verona
Altair Robotics Lab

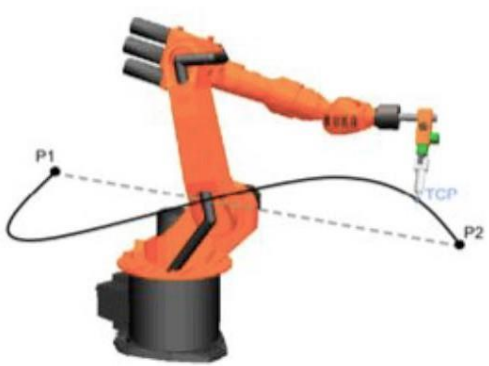
UNIVERSITÀ di VERONA
Dipartimento di INFORMATICA



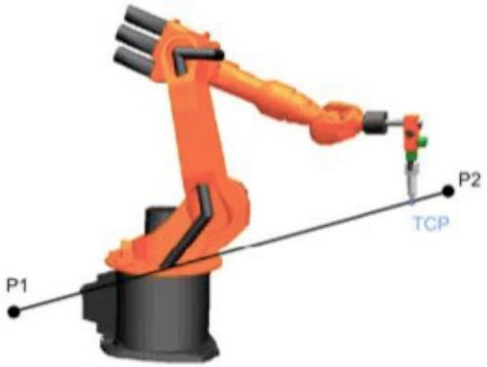
Overview

- Recap about motion commands supported by industrial robots
- Problem when programming robot motions: SINGULARITIES
- Singularities of a typical industrial robot
- Singularities of a typical cobot
- Work-cell design suggestions

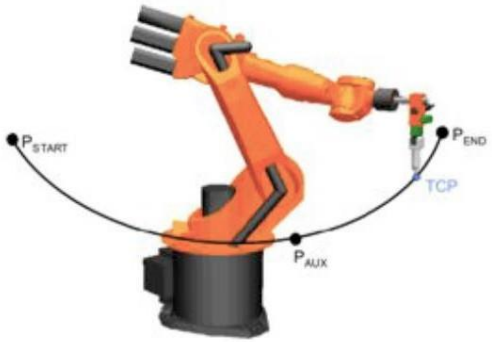
Recap from previous lessons



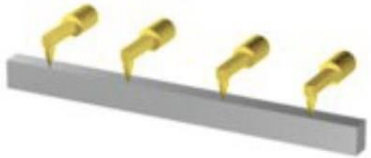
PTP motion
(point-to-point, linear in joint space)



LIN motion
(linear in Cartesian space)

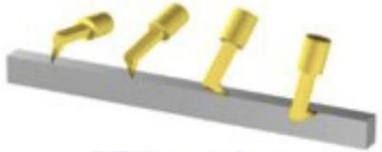


CIRC motion
(circular in Cartesian space)



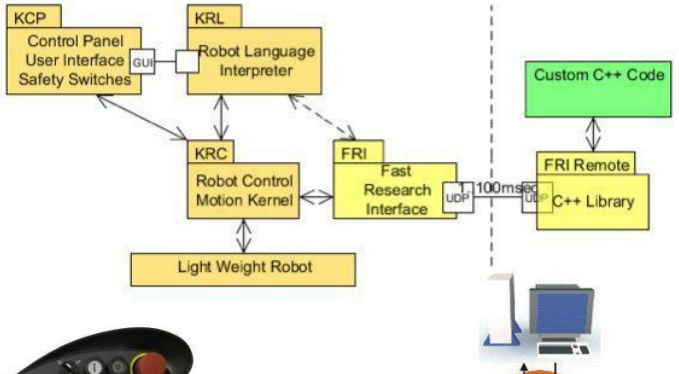
CONST orientation

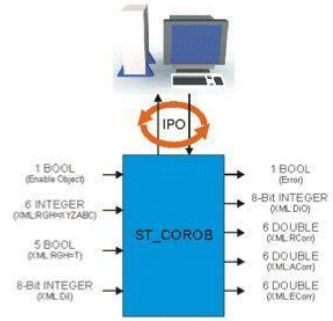
end-effector orientation



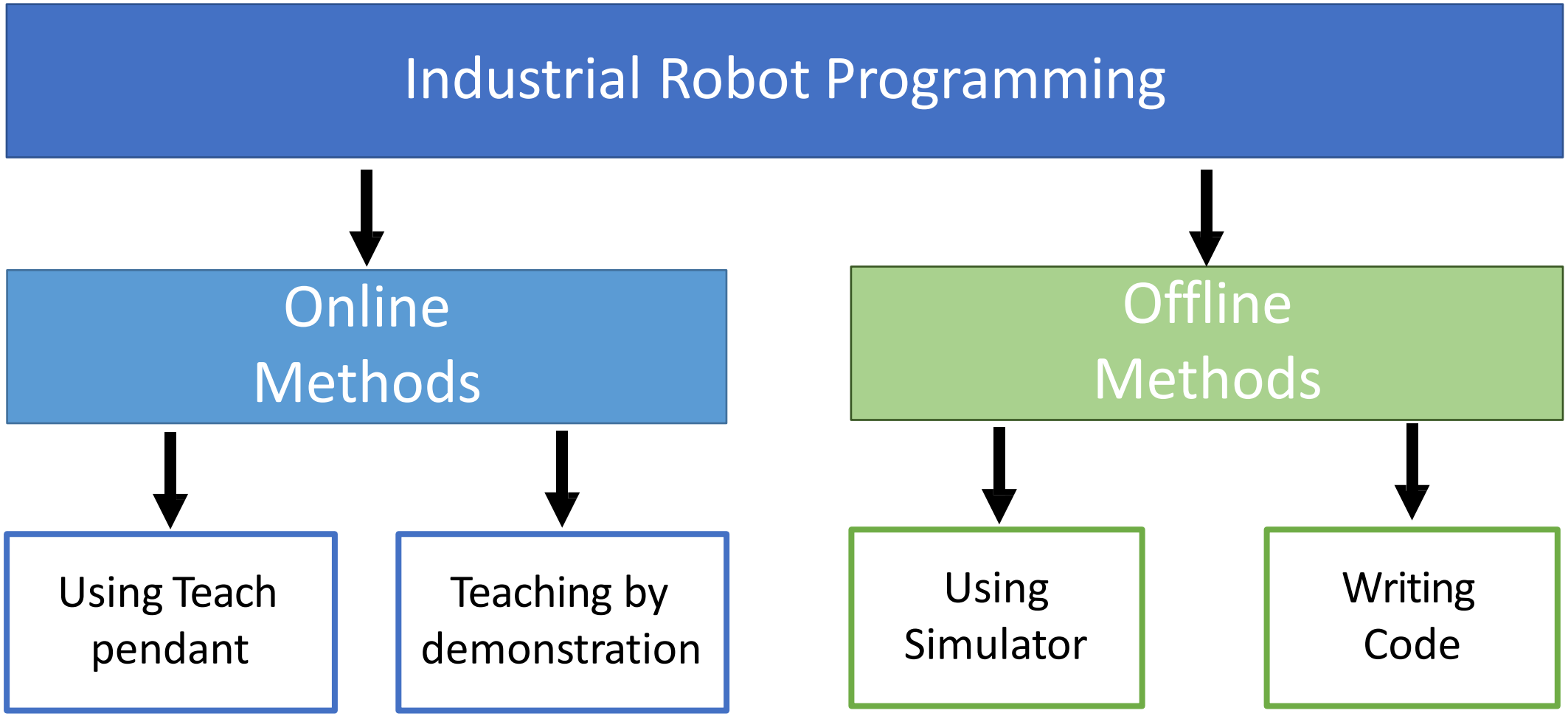
PTP motion
(linear in RPY angles)

- Common components in Robot programming
- Evolution of robot programming: from teaching to robot-oriented approaches
- Classification of programming methods





Classification of Programming methods



Industrial Robotic motions

An industrial robot can be controlled in two spaces**: joint space and Cartesian space**.

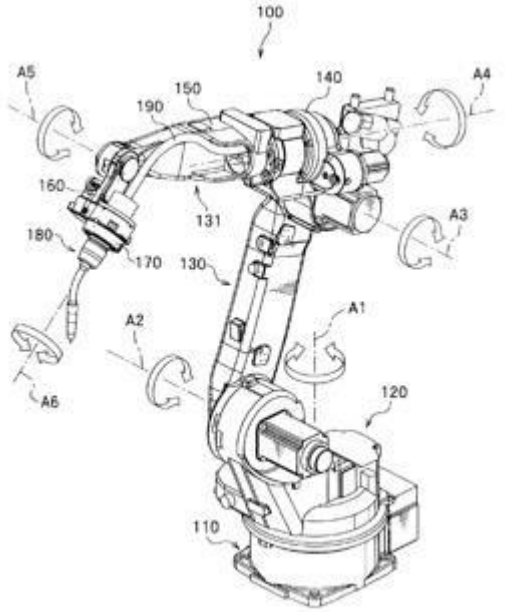
- For **joint-space** motion commands (sometimes incorrectly called point-to-point commands), the programmer specify a desired set of joint positions, and the robot moves by translating or rotating each joint to the desired joint position, simultaneously and in a linear fashion.
- For **Cartesian-space** motion commands, you specify a desired pose for the end-effector AND a desired Cartesian path (linear or circular).
  - To find the necessary joint positions along the desired Cartesian path, the robot controller must calculate the inverse position and velocity kinematics of the robot.
  - Singularities arise when this calculation fails and must therefore be avoided.

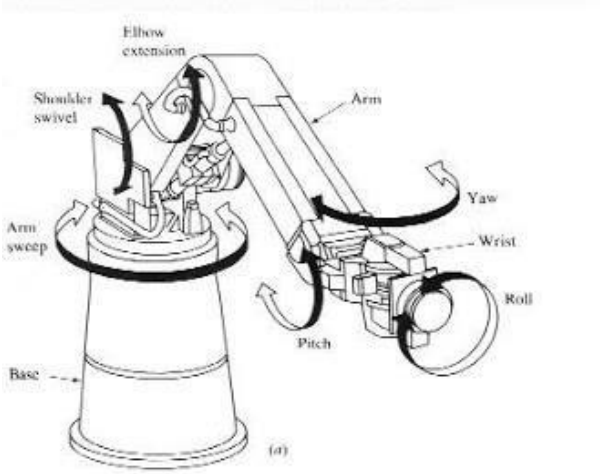
Limits in joint and Cartesian Space

- Try jogging a six-axis robot arm in joint space, and the only time the robot will stop is when a joint hits a limit or when there is a mechanical interference
- Try jogging the same robot in Cartesian space and the robot will frequently stop and refuse to go in certain directions, although it seems to be far from what you think is the workspace boundary.
- A robot singularity is a configuration in which the robot end-effector becomes blocked in certain directions.

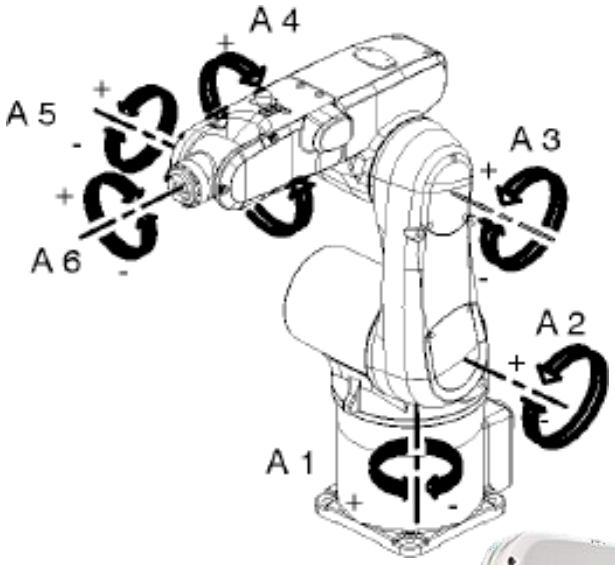
Limits in joint and Cartesian Space motions

- Any six-axis industrial robotic arm has singularities.
  - The correct term is six-degree-of-freedom serial manipulator , but let's stick to the popular, unscientific term six-axis used in Industry ☺.
- The complexity and types of singularity in a robot arm will depend on:
  - the number of joints,
  - their types (linear or revolute)
  - their geometric arrangement.
- Because singularities significantly deteriorate the performance of an industrial robot arm, <u>you must learn how to identify them and never move close to them, when using Cartesian-space motion commands</u>.





Typical structure of a 6 axis industrial manipulator



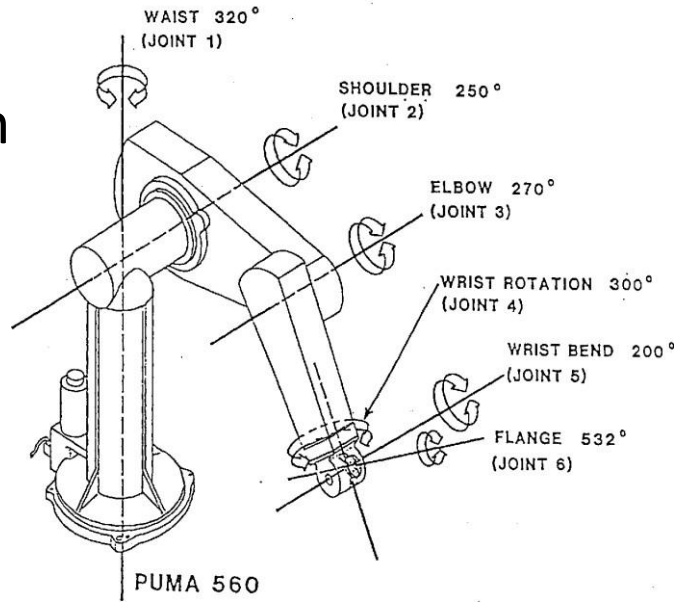
Example:
KUKA KR5 Sixx R650

Standard arrangement of joints axes:

- the axis of the first joint is vertical
- the axes of joints 2 and 3 are horizontal,
- the axis of joint 4 is normal to the axis of joint 3
- the axes of the last three joints typically intersect at one point (spherical or inline wrist)

Most of industrial manipulator have this structure, inherited from the father of all industrial manipulator**: Unimation PUMA**.

often bizarrely referred to as a *vertically articulated robot*



Other examples

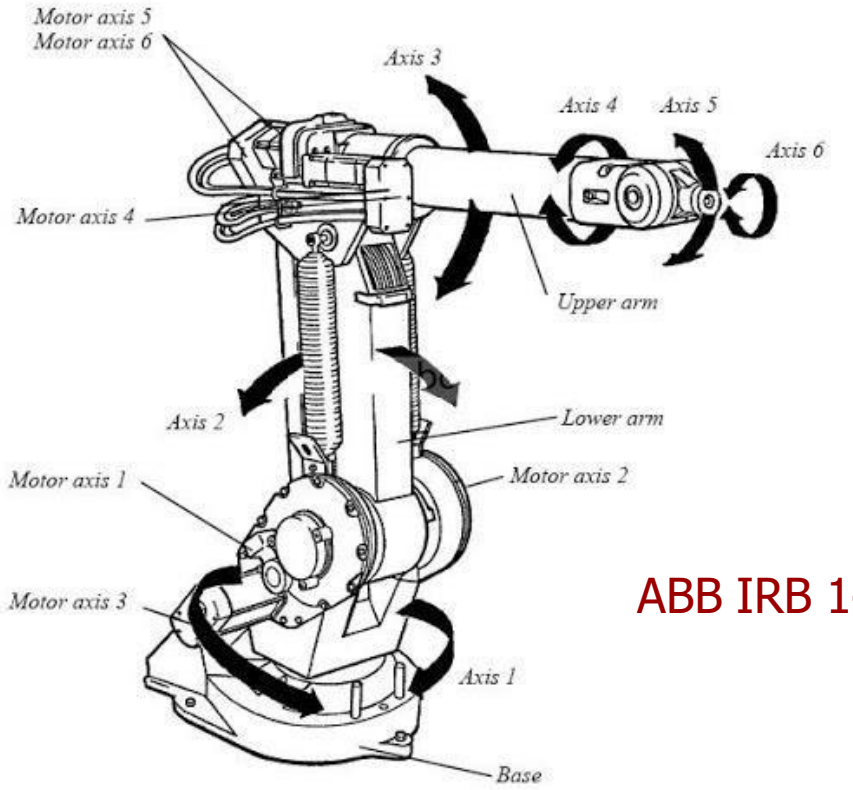
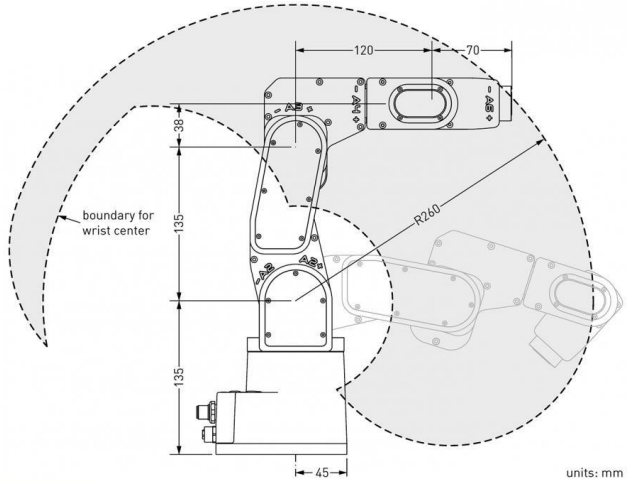


ABB IRB 1400



Mecademic M500

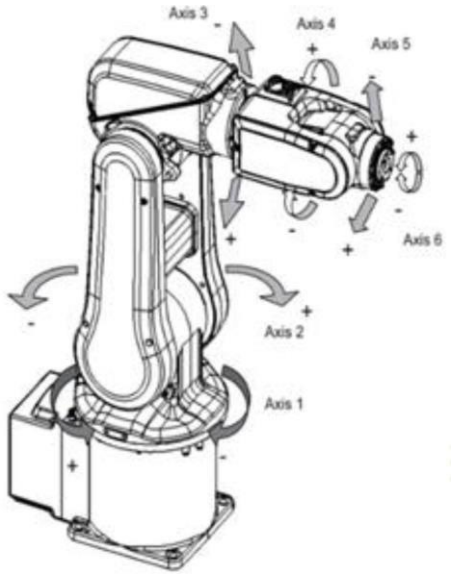


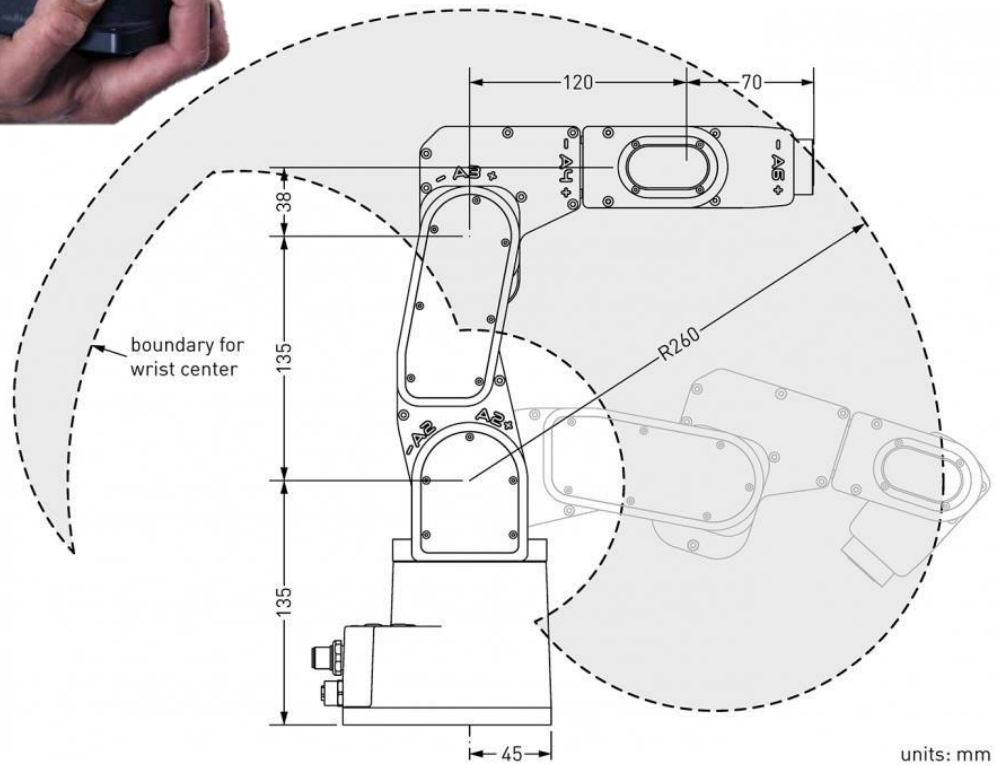
ABB IRB 120

There are also some exceptions, such as task specific manipulator (e.g. painting robots) and most of the collaborative robots.

Mecademic M500 - Specs

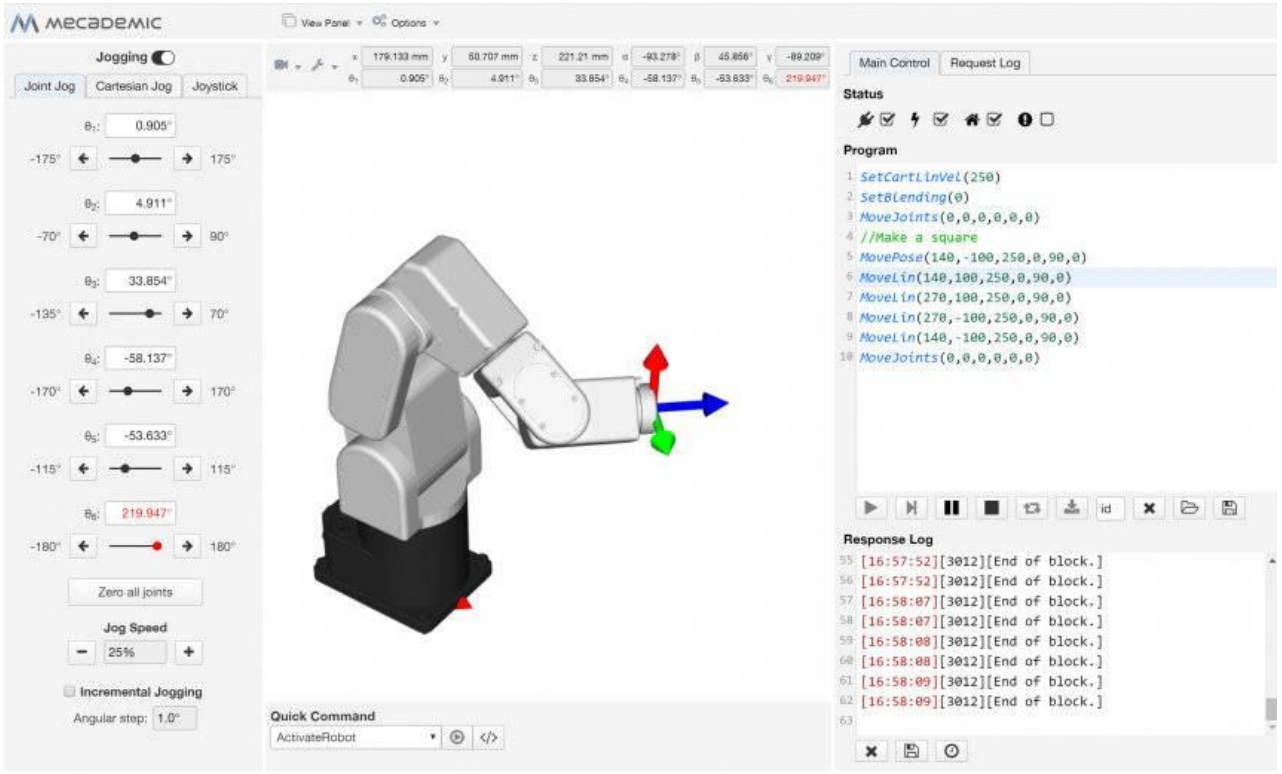
When fully extended, the length of the small industrial robot arm is about 330 mm.
Reach: 260 mm (see diagram below)
Payload: 0.5 kg rated (max. 1 kg)
Weight: 4.5 kg
Mounting: any orientation
Repeatability: 0.005 mm
Power Supply: 24V power supply that has an embedded safety module with an E-Stop and safety I/O interface.

The Meca500's controller is embedded in its palm-sized base. There is no bulky controller cabinet, no teach pendant, and no messy cables.



Mecademic M500 - Operation

- Simply connect the robot to any computing platform via the Ethernet cable provided and use a programming language of your choice (Java, C#, Python, etc.).
- The robot can also be controlled by most PLCs using either Ladder or Structured Text.
- Users may communicate with the robot via **EtherCAT**, ensuring guaranteed response times.



- Alternatively, users may send Mecademic's proprietary commands to the robot arm (as parsed strings) over **TCP/IP.** The commands are typical robot instructions such as MoveLin(x,y,z,α,β,γ).
- The robot's controller features an intuitive web interface for operation. It can be accessed via any web browser.

Mecademic M500 – Video

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

Robot singularities and motions (1)

- In a singularity, a robot cannot displace its end-effector along certain directions.
- At a singularity, a robotic arm loses one or more degrees of freedom.
- A robot singularity is a physical blockage, not some kind of abstract mathematical problem, although we have a simple mathematical explanation for it.
- Singularities of six-axis robot arms can be explained with the following inverse velocity kinematic equation:

$$\dot{q} = J^{-1}v,$$

- When this Jacobian matrix becomes singular (at certain joint positions), the above equation is not defined and finding joint velocities for certain Cartesian velocity vectors becomes impossible.

Robot singularities and motions (2)

- 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.

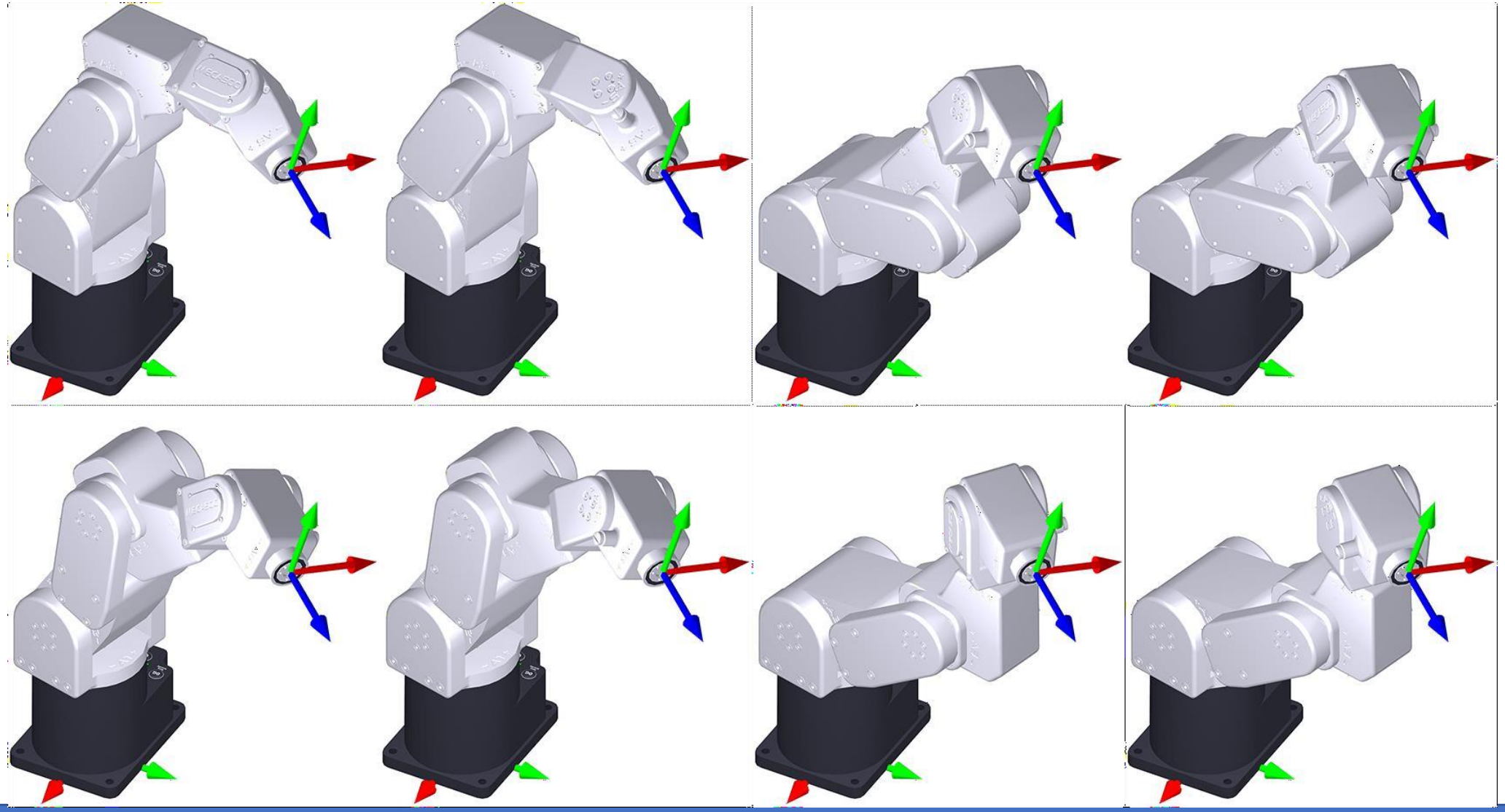
$$\dot{q} = J^{-1}v,$$

- 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.

Robot singularities as internal workspace boundaries

- Robot singularities are not only configurations in which the inverse velocity kinematics fail: in a singularity, the inverse position kinematic equations of a robot degenerate too.
- For a desired end-effector pose, six-axis robots can generally have up to eight different solutions for the joint positions, corresponding to the same end-effector pose.
  - Even more solutions are possible, if we consider that joint 6 is often unlimited, but let's keep it limited to one complete rotation for the sake of discussing singularities.
- These eight different solutions correspond to eight different configuration types. Changing a configuration type requires passing through a singularity.

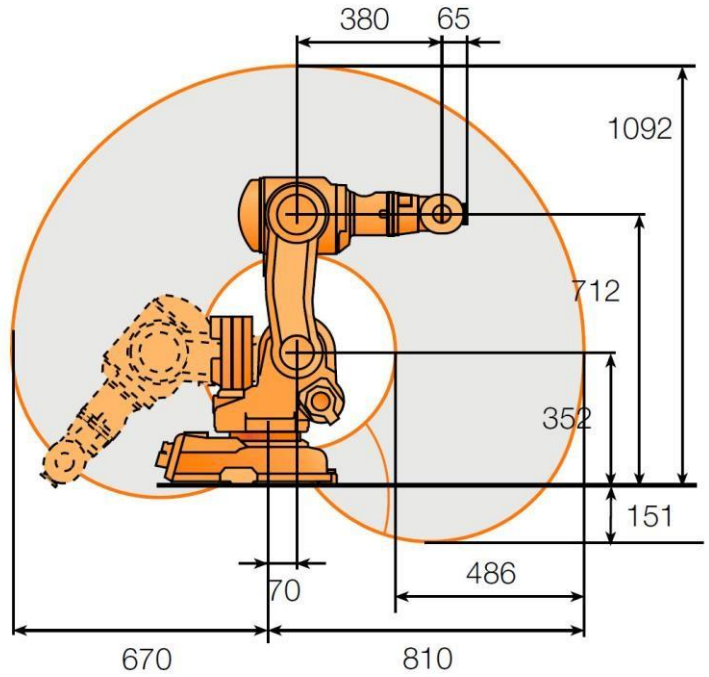
Fixed end-effector pose corresponding to eight different sets of joint positions

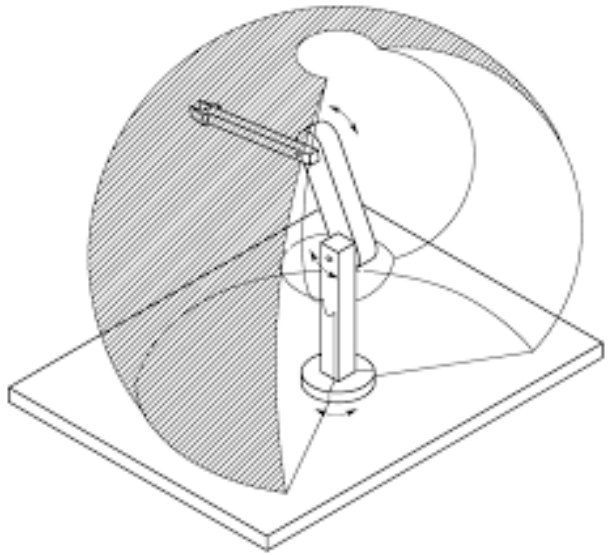


Workspace of a robotic manipulator

The workspace of a six-axis robot is the set of all poses (positions and orientations) attainable by a particular end-effector mounted on that robot.

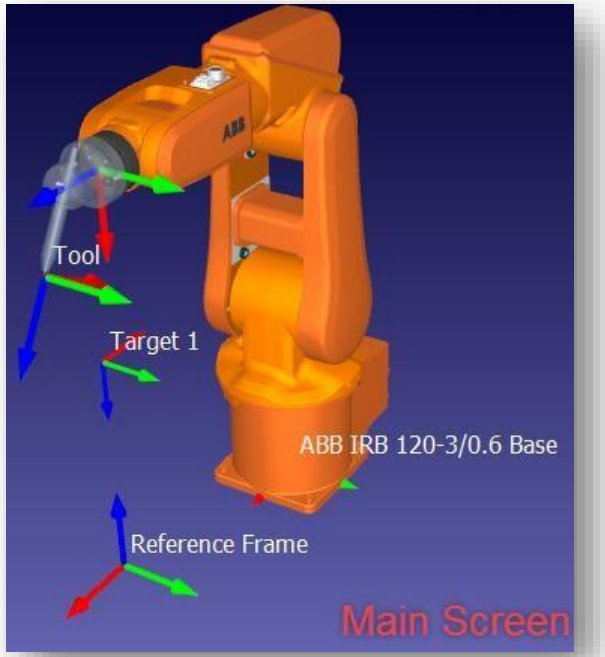
- no one can tell you what is the workspace of a specific six-axis robot arm, unless you specify what is the end-effector that you intend to use;
  - if you mount a relatively long end-effector, you won't be able to reach any position from this volume with the tip
- the workspace of a six-axis robot arm is highly heterogeneous, in terms of performance criteria such as speed and precision.
- the workspace is a six-dimensional entity that is generally impossible to represent graphically.

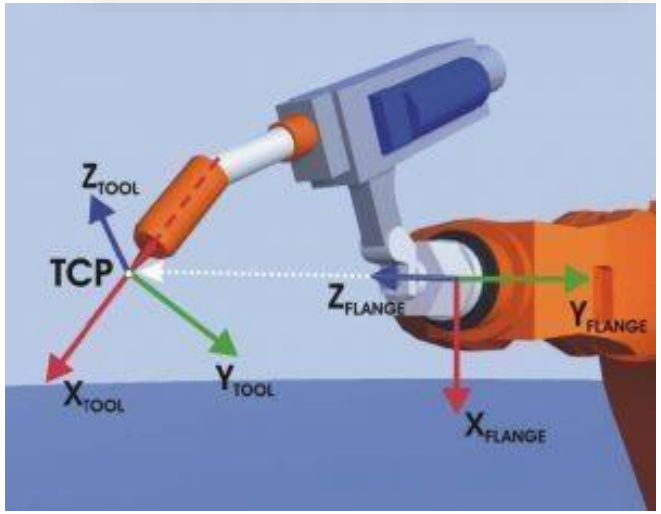




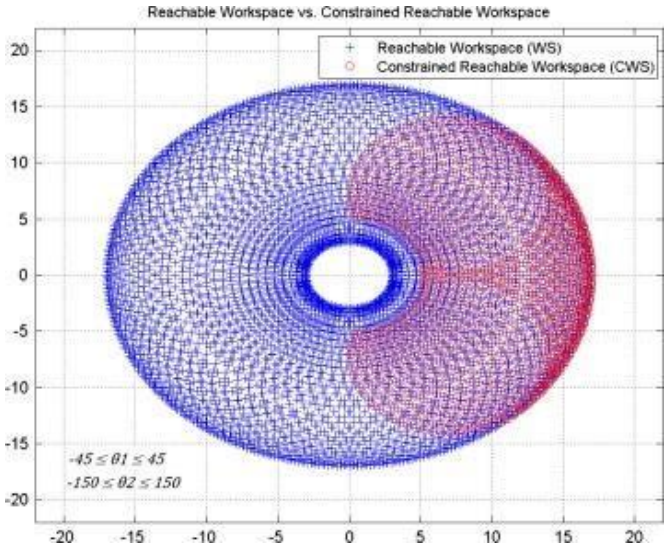
Workspace and EE tool

- Industrial robot arms are generally sold without any tooling
- an end-effector (such as a gripper or a touch probe) is eventually attached to the robot's flange.
- The user must then associate a tool reference frame, fixed to this end-effector.
- This tool reference frame is defined with respect to the flange reference frame.
- The origin of the tool reference frame is called the TCP (tool center point).
- The robot workspace is the set of poses attainable by the robot's tool reference frame.
- Whatever the end-effector, for each feasible position of the TCP, the range of attainable tool orientations is completely different, due to mechanical interferences, joint limits, and link length limitations.





Workspace and singularities

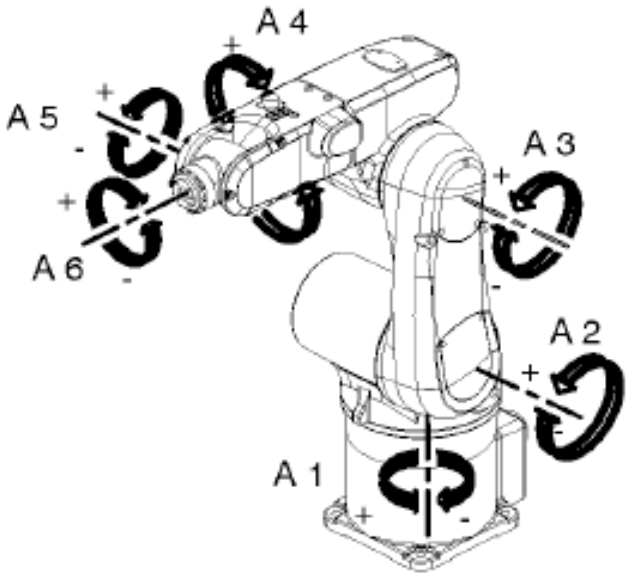


- The Cartesian workspace of the typical six-axis robot arm is composed of eight 6D entities.
- The boundaries between these entities correspond to singularities, while the remaining boundaries correspond to joint limits (and other mechanical interferences).
- Each of these singularity-free workspace subsets corresponds to a specific robot configuration.
- Note that if a given pose seems to be inaccessible with the current robot configuration, it may be accessible with another configuration.
- A common mistake is to consider only one of the singularity-free workspace subsets instead of the total workspace (i.e., working in only one of the eight configuration modes).
- Users typically teach robot positions by jogging, rather than by specifying a pose and leaving the robot controller in finding the best joints configuration

Typical singularities of a Industrial manipulator

Wrist singularity: axis A6 and axis A4 aligned

- The most frequently-encountered singularity industrial manipulator with spherical wrist.



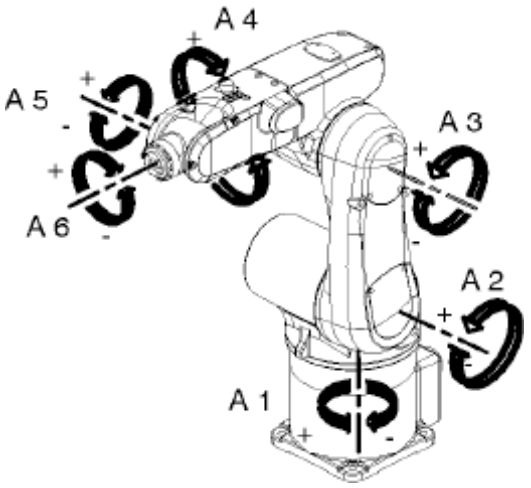
Elbow singularity: when center of wrist lies on the plane of A2 e A3

- This singularity is the least unexpected and is easy to avoid.

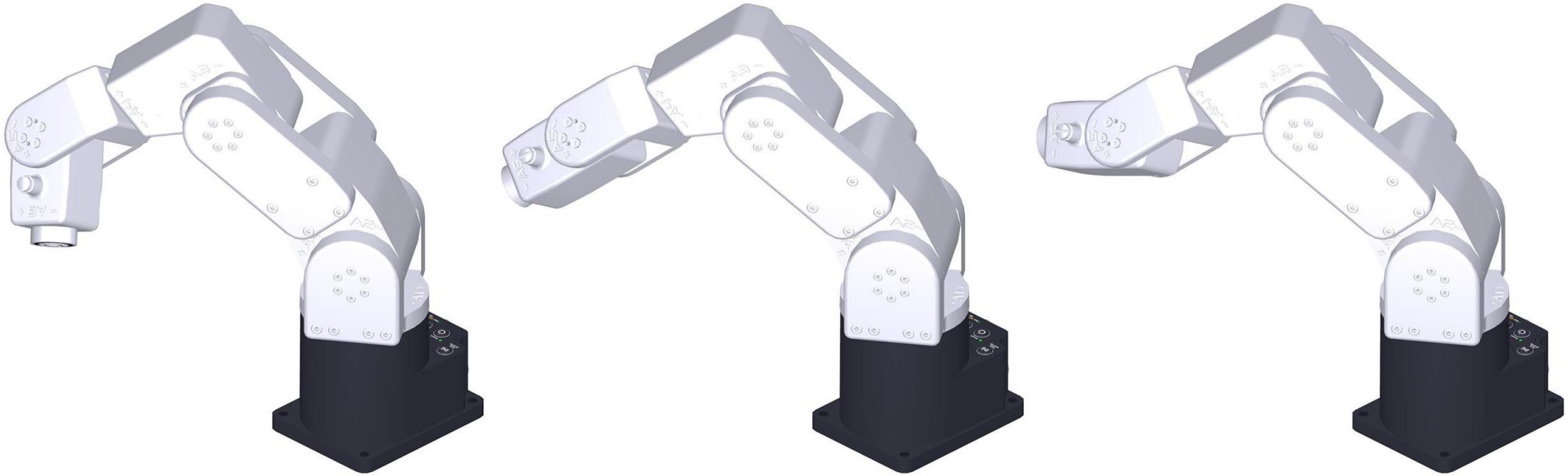
Shoulder singularity: when center of wrist lies on A1 axis

- This singularity is the most complex as it does not depend on a single joint position, as do the other two.

Wrist Singularity

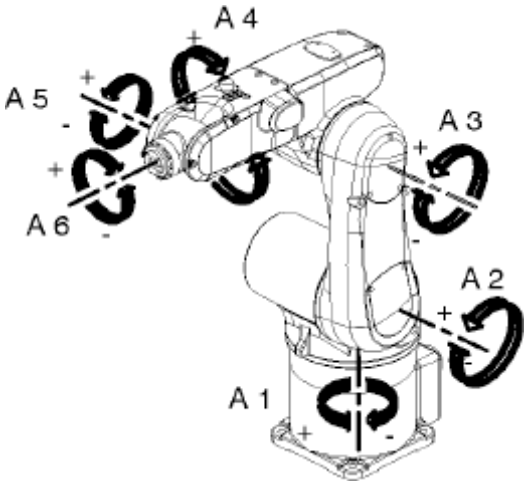


- In most robots, this condition corresponds to $\vartheta_5 = 0°$.
- In the figure below, the middle configuration corresponds to a wrist singularity whereas the other two correspond to two different sets of configuration types.
- In the left configuration, we have the so-called **no-flip** condition ($\vartheta_5 > 0°$) whereas, in the right configuration, we have the **flip** condition ($\vartheta_5 < 0°$).



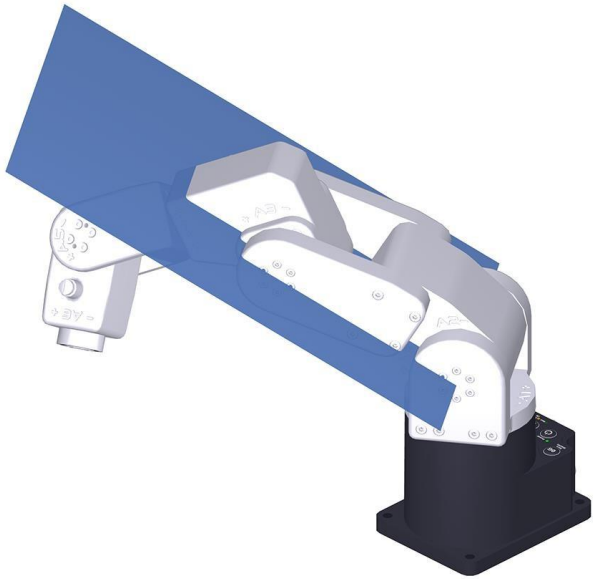
Wrist Singularity (2)

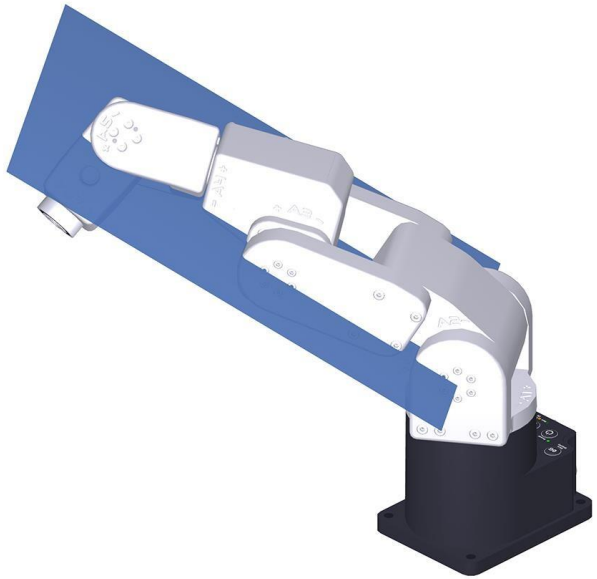
- In a wrist singularity, the robot cannot move in the direction of the axis of joint 5.
- In order for the TCP to follow a line through the singularity, joints 4 and 6 must simultaneously rotate 90° in opposite directions, at the singularity. Thus, crossing a wrist singularity while following a line is physically possible but, at the singularity, the end-effector remains motionless while joints 4 and 6 rotate.
- In a wrist singularity, there are infinite solutions to the inverse position kinematics of the robot.

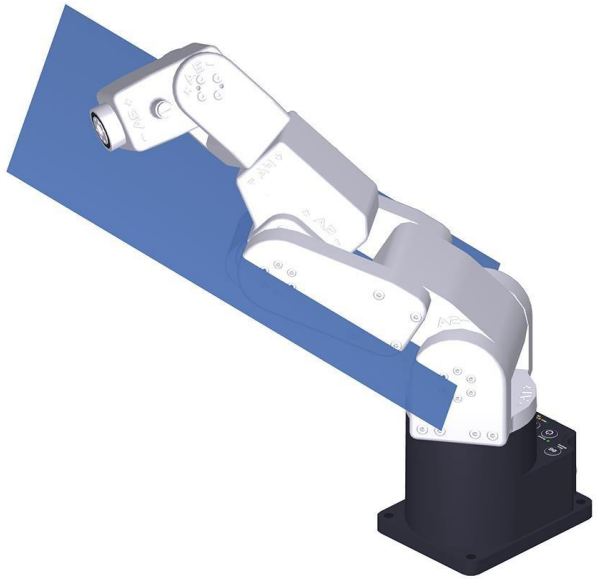


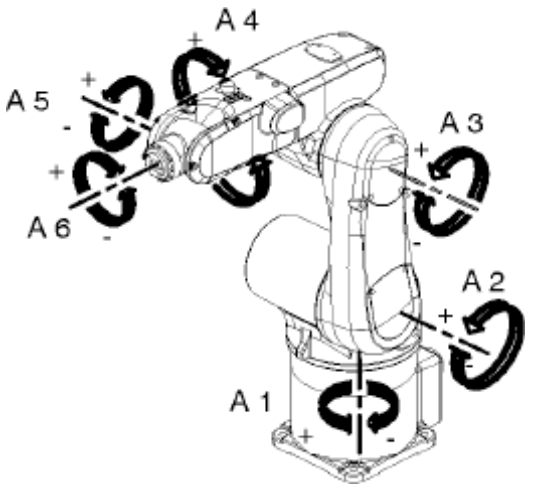
Elbow Singularity (1)

- It occurs when the wrist center (the point where the axes of joints 4, 5 and 6 intersect) lies on the plane passing through the axes of joints 2 and 3.
- We can say that, in an elbow singularity, the arm is fully stretched. (Due to mechanical interferences, most robot arms cannot be fully folded, which would be the other elbow singularity.)
- An elbow singularity is determined only by the position of joint 3, $\vartheta_3 = \vartheta_{singular}$.



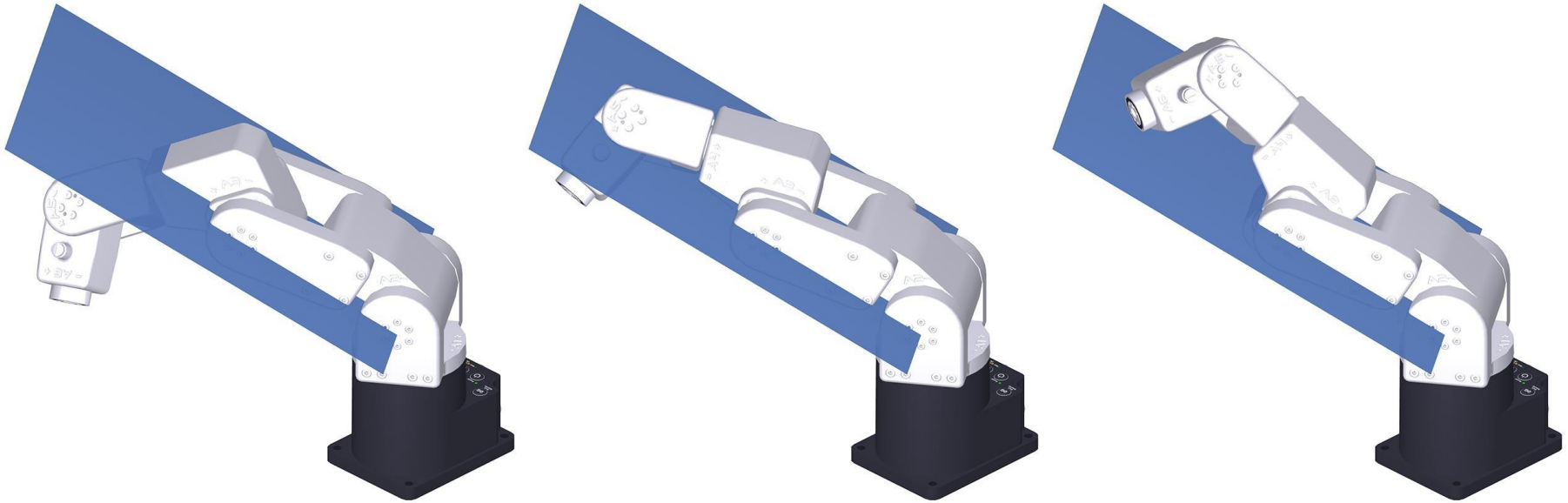






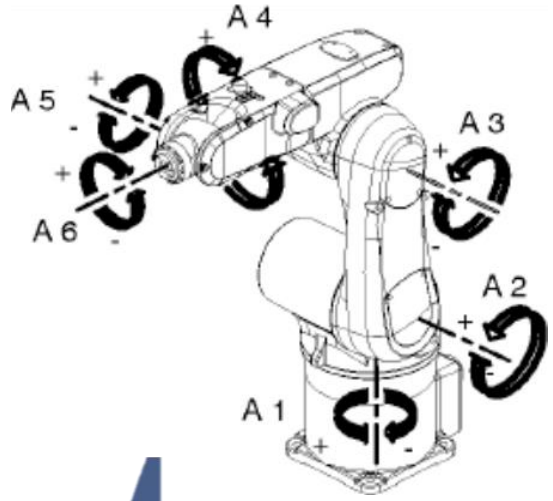
Elbow Singularity (2)

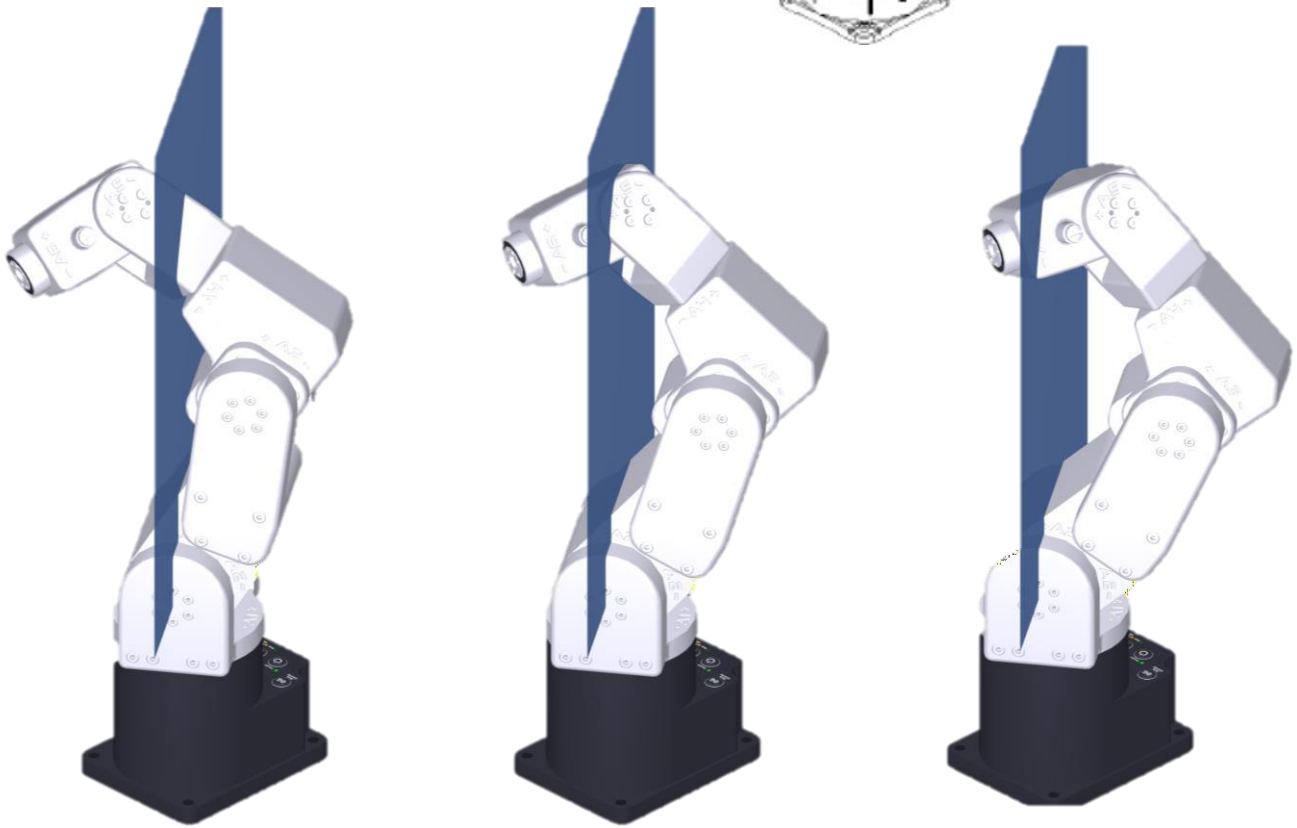
- The middle configuration corresponds to an elbow singularity ($\vartheta_3 = \vartheta_{singular}$) whereas the other two correspond to two different sets of configuration types.
- In the left configuration, we have the so-called **elbow-up** condition ($\vartheta_3 > \vartheta_{singular}$) whereas, in the right configuration, we have the **elbow-down** condition ($\vartheta_3 < \vartheta_{singular}$).
- In an elbow singularity, two sets of inverse position kinematic solutions degenerate into one. <u>This singularity is the least unexpected and is easy to avoid</u>.



Shoulder singularity

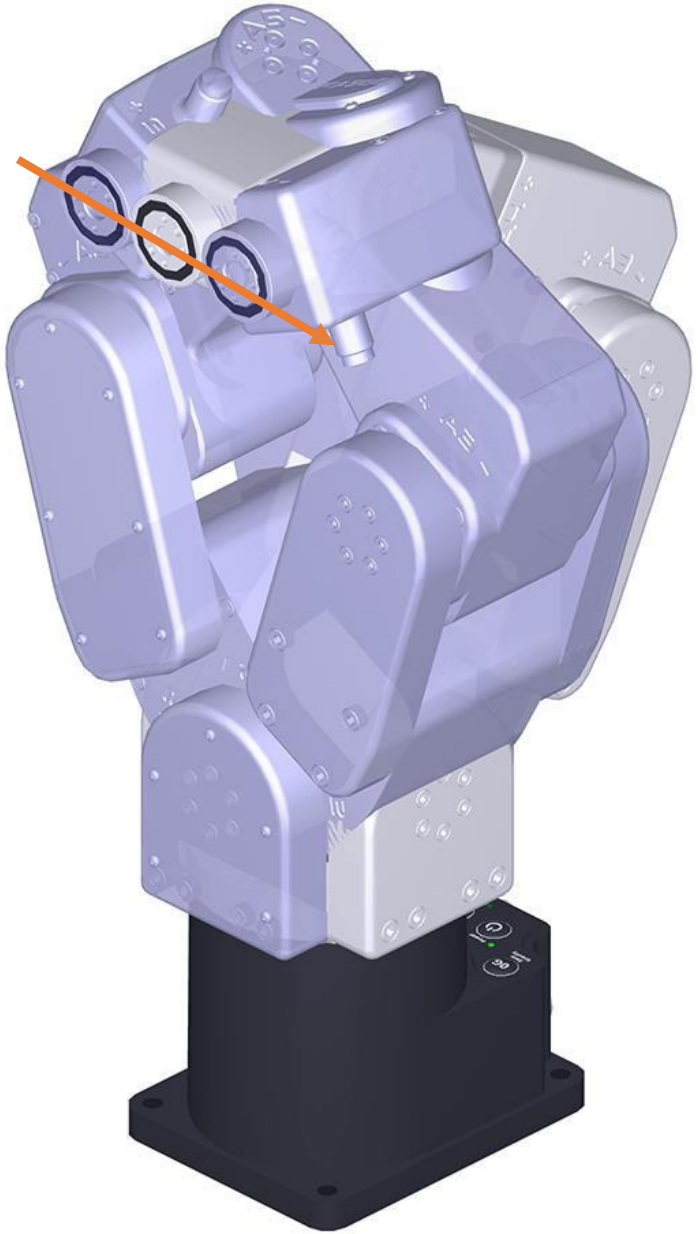
- It occurs when the center of the robot wrist lies in the plane passing through the axes of joints 1 and 2 (or through the axis of joint 1 and parallel to the axis of joint 2).
- This singularity is the most complex as it does not depend on a single joint position, as do the other two.
- the middle configuration corresponds to a shoulder singularity whereas the two others correspond to two different sets of configuration types. In the left configuration, we have the front condition whereas, in the right configuration, we have the back condition





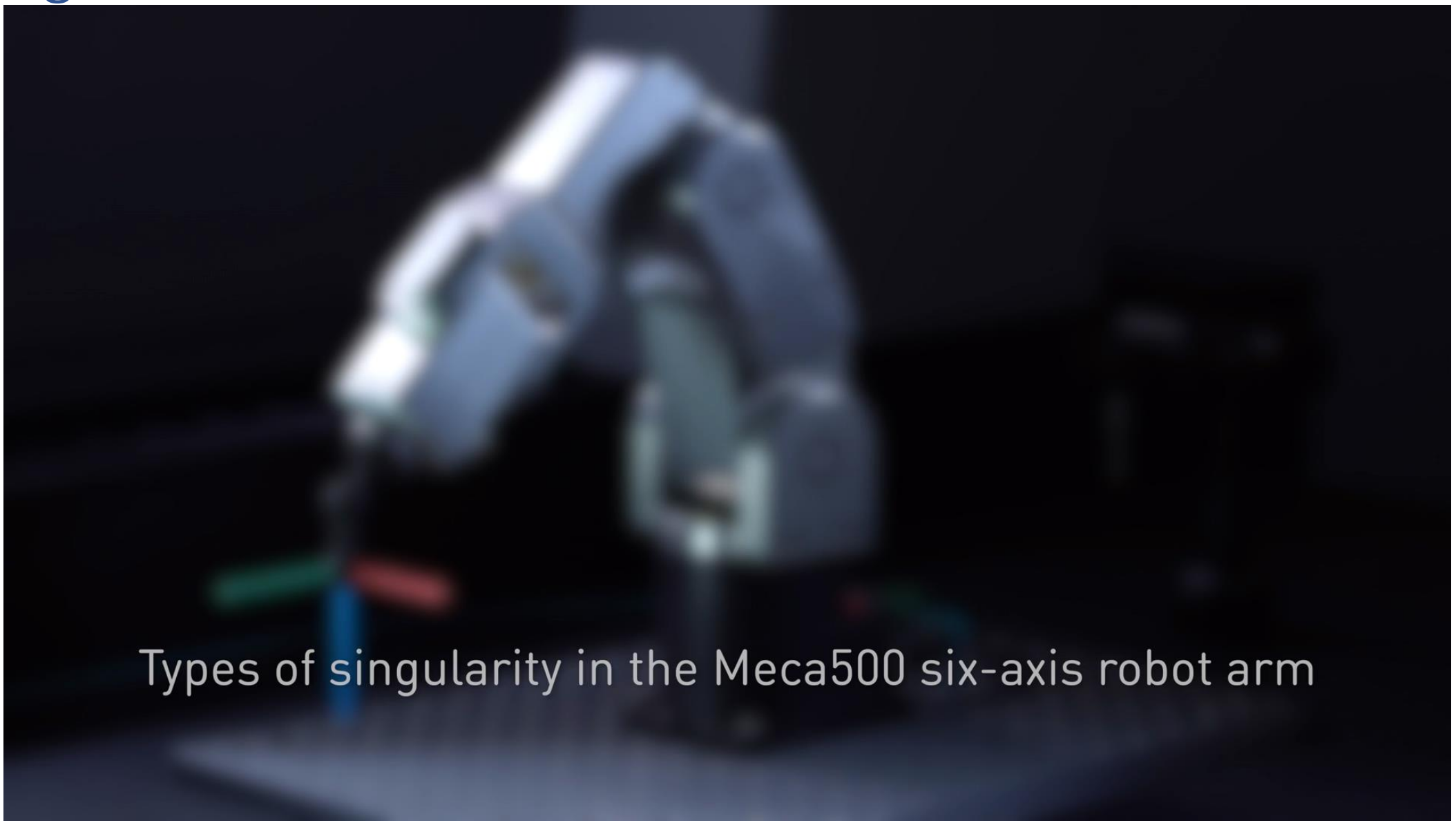
Crossing Shoulder singularity

- In a shoulder singularity, the robot cannot move in the direction of the axis of joint 2.
- In order for the TCP to follow a line through the singularity, joints 1 and 4 must simultaneously rotate 90° in opposite directions (other joints need to rotate too), while the end-effector remains stationary.
- Thus, it is physically possible to cross a shoulder singularity while performing a linear motion (i.e. following a line in Cartesian space), at the singularity, the end-effector remains motionless while some of the joints rotate.
- In other words, it is impossible to cross a shoulder singularity without having the end-effector stop.
- Due to numerical problems, it is impossible to crossing this singularity while jogging in Cartesian space.



Singularities video

https://www.youtube.com/watch?v=lD2HQcxeNoA



Industrial robot singularities handling

- Manufacturers program their robots so that singularities don't break the robot.
- However, in the past this just meant that if one joint was commanded to move too fast, the robot stopped completely with an error message.
- This was an effective but not very elegant solution.

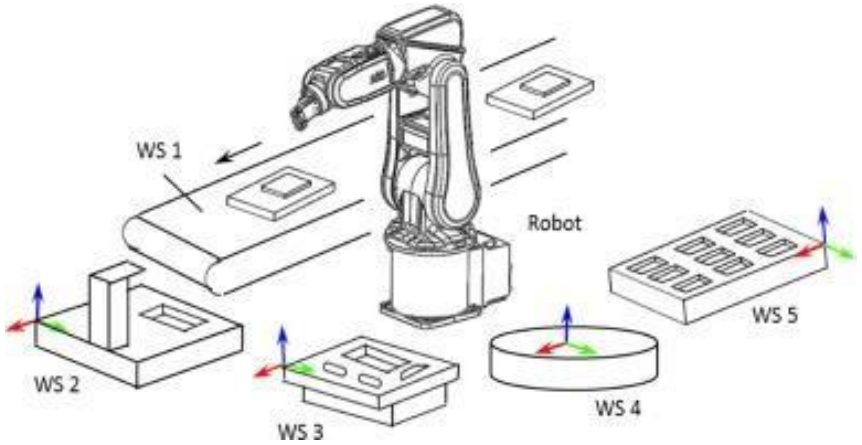
- These days, many robot manufacturers are improving their singularity avoidance.
- In the previous video , the robot had been programmed with a maximum speed for each joint.
- The robot slowed down when it reached the middle of the line. When it passed the singularity, it was able to continue doing the rest of the line at the correct velocity.
- In many case this provide bad task performance (e.g., painting), but the robot nevertheless functions correctly and doesn't get stuck.

Robot singularities and configurations

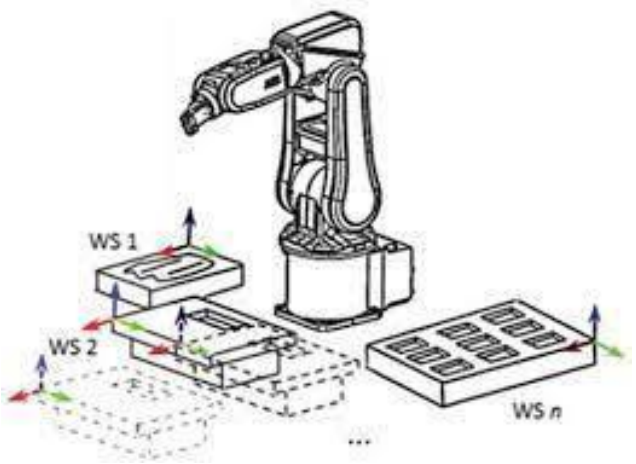
- The configuration type is defined by the set of three configuration conditions described above, namely {flip/no-flip, elbow-up/elbow-down, front/back}.
- Passing from one configuration type to another requires crossing a singularity.
- It is therefore imperative that when using Cartesian-space commands, **you specify the desired configuration type**, not merely the desired end-effector pose.

flip no-flip	elbow-up elbow-down	Front back
no	up	front
no	up	back
no	down	front
no	down	back
yes	up	front
yes	up	back
yes	down	front
yes	down	back

Good practice in robotic cell design to avoid singularities



- Keep robot singularities in mind while designing your robot cell.
- Pay particular attention to the design of the adaptor plate for your end-effector.
- If you do not need six degrees of freedom to position and orient your end-effector (e.g. if laser cutting or inserting round pins), take advantage of your redundant degree of freedom to swing away from singularities.
- Consider the use of an offline programming and simulation software, which can be of great assistance in checking for robot singularities and in testing different cell design



Robotic cell design guidelines (1)

The English saying goes “Measure twice, cut once,” but when it comes to robot cell design, opt for the Russian version “Measure seven times, cut once.”

- Limit the movement in Cartesian Space, use it when you need to follow a specific path
- Move the task in another zone of the workspace, i.e. consider a different robot config.
- Fix your robot on an available support
  - Many industrial robots can be attached in any orientation (e.g., upside-down, or on a wall). Start by exploring available support surfaces before designing a dedicated support that will inevitably eat up valuable space.
- Robot performances are not constant over the entire workspace, for example the more stretched the robot arm, the less its precision (the more flexible it is) and the more time it takes to suppress vibrations at the tooltip.
- Take advantage of the robot's redundancy if you don't need six degrees of freedom (or consider a 7 axis manipulator or mounting your 6 axis manipulator on a linear guide)

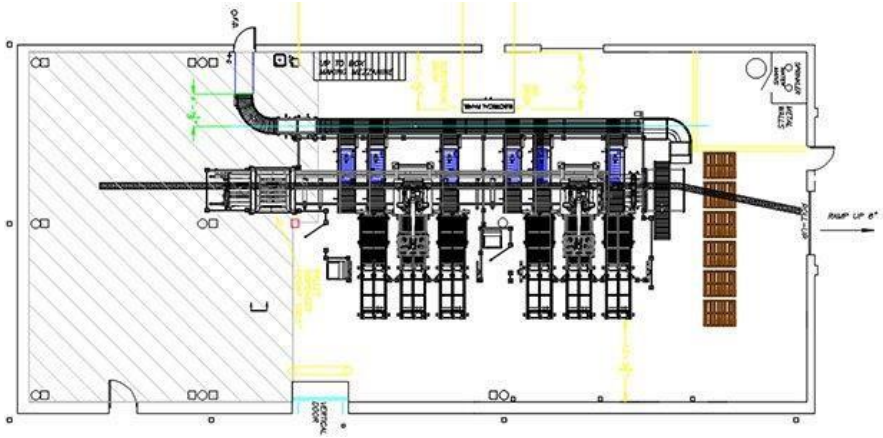
Robotic cell design guidelines (2)

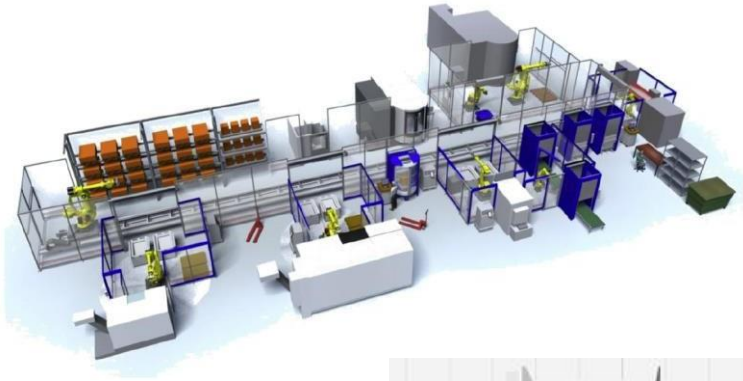
Carefully design or choose your EE tool:

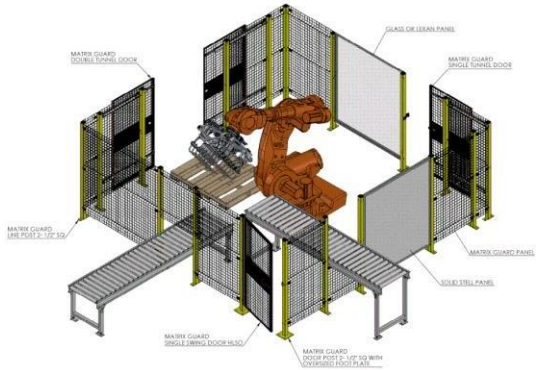
- you could ask to introduce an adapted plate between flange and EE to introduce a rigid transformation. Adding a small angles (5 to 15 degrees) to the tooling could reduce the chance that the robot moves into a singularity.
- Pay attention to EE cabling and mechanical dimensions, limit the motions of six axes; Whenever possible, at a desired pose, try to keep joint 6 as close as possible to zero degrees.
- If you must use a wide range of orientations (e.g., optical quality inspections) place the tool reference frame as close as possible to the flange reference frame.

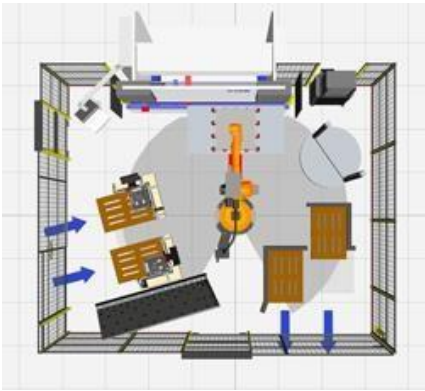
Final considerations

- In conclusion, unless your application involves only a few repetitive movements, you must fully understand the workspace of your robot and spend a considerable time designing your robot cell.
- Robot cell design is not only about being able to perform all robot movements; it's also about optimizing cycle time and other performance criteria (precision, energy consumption).



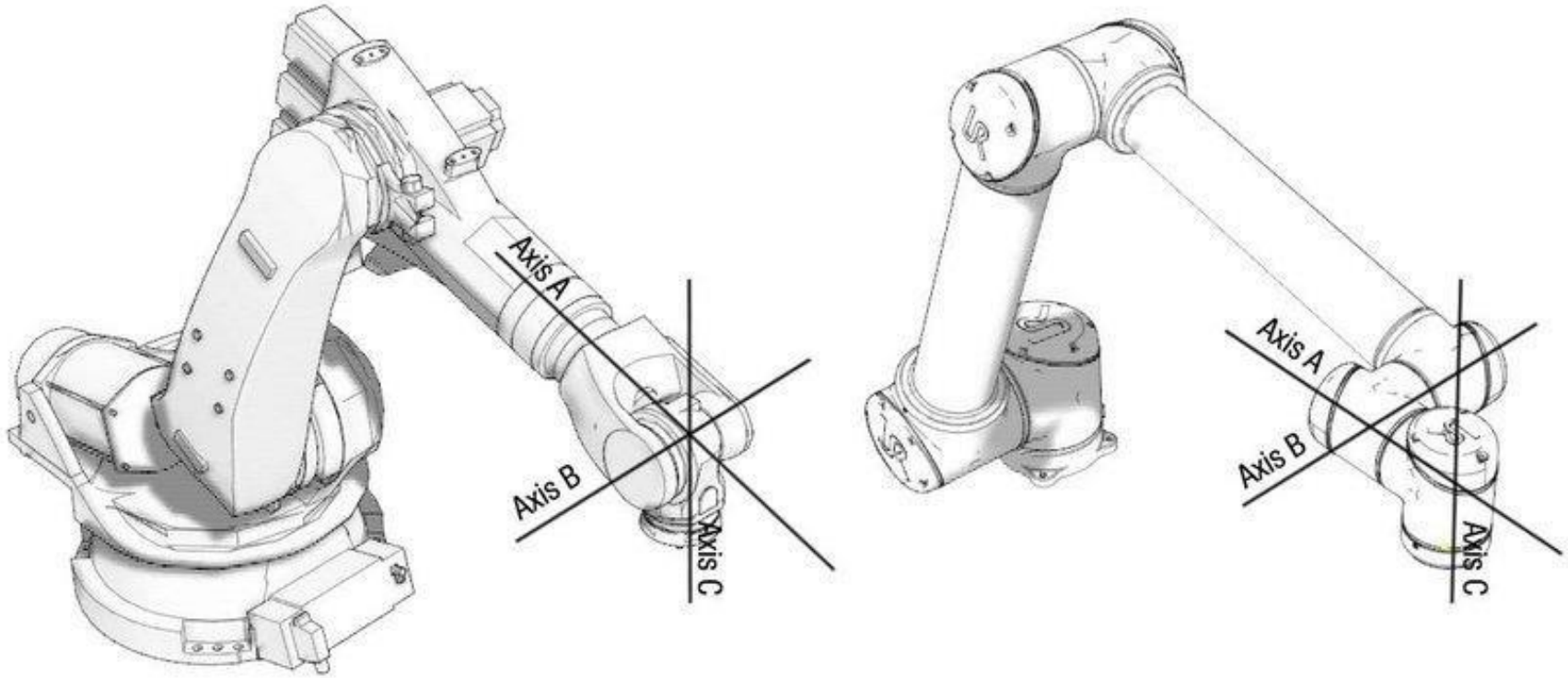






Singularities of a typical collaborative robot

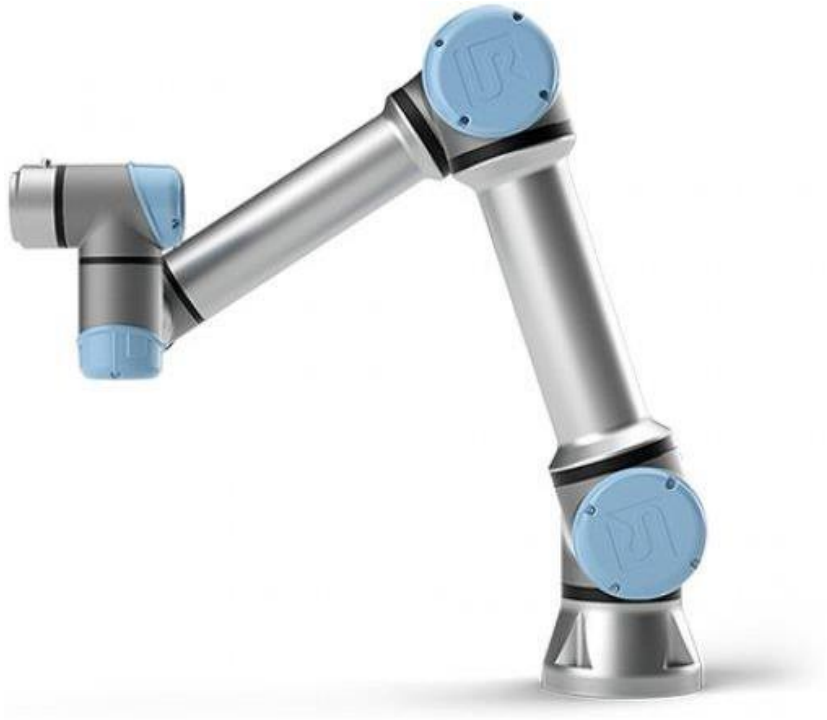
Collaborative robots do not have a PUMA-type architecture with a spherical wrist



Example of a typical collaborative robot: Universal Robot UR5 Robotic Manipulator

6-axis robot arm with a working radius of 850 mm / 33.5 in

Weight	18.4 kg / 40.6 lbs
Payload	5 kg / 11 lbs
Reach:	850 mm / 33.5 in
Joint ranges:	+/- 360° on all joints
Speed:	Joint: Max 180°/sec. Tool: Approx. 1 m/sec. / Approx. 39.4 in/sec.
Repeatability:	+/- 0.1 mm / +/- 0.0039 in (4 mil)
Footprint:	Ø149 mm / 5.9 in
Degrees of freedom:	6 rotating joints
Control box size (WxHxD):	475 mm x 423 mm x 268 mm / 18.7 x 16.7 x 10.6 in
I/O ports:	10 digital in, 10 digital out, 4 analogue in, 2 analogue out
I/O power supply:	24 V 1200 mA in control box and 12 V/24 V 600 mA in tool
Communication:	TCP/IP 100 Mbit: IEEE 802.3u, 100BASE-TX Ethernet socket & Modbus TCP
Programming:	Polyscope graphical user interface on 12 inch touchscreen with mounting
Noise:	Comparatively noiseless
IP classification:	IP54
Power consumption:	Approx. 200 watts using a typical program
Collaboration operation:	Tested in accordance with sections 5.10.1 and 5.10.5 of EN ISO 10218-1:2006
Materials:	Aluminium, ABS plastic
Temperature:	The robot can work in a temperature range of 0-50°C
Power supply:	100-240 VAC, 50-60 Hz
Calculated Operating Life:	35,000 Hours



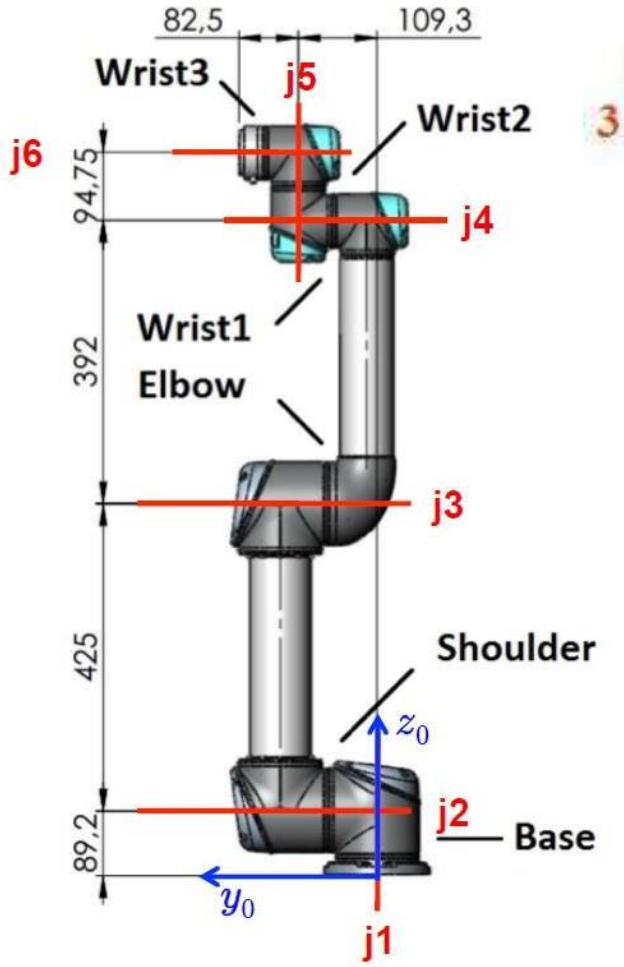
Applications of UR5 video

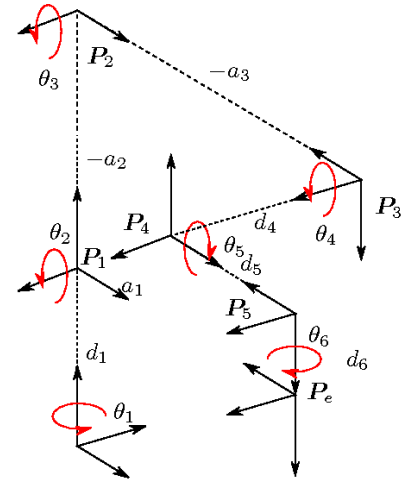
https://www.youtube.com/watch?v=pIcxOGo7ieU

UR5 Kinematic structure

Indeed, the vast majority of six-axis cobots on the market have the same arrangement of six revolute joints as the UR5 by Universal Robots:

- the axes of joints 2, 3, and 4 are parallel
- the axis of joint 1 intersects and is normal to the axis of joint 2
- the axis of joint 5 intersects and is normal to the axes of joints 4 and 6.

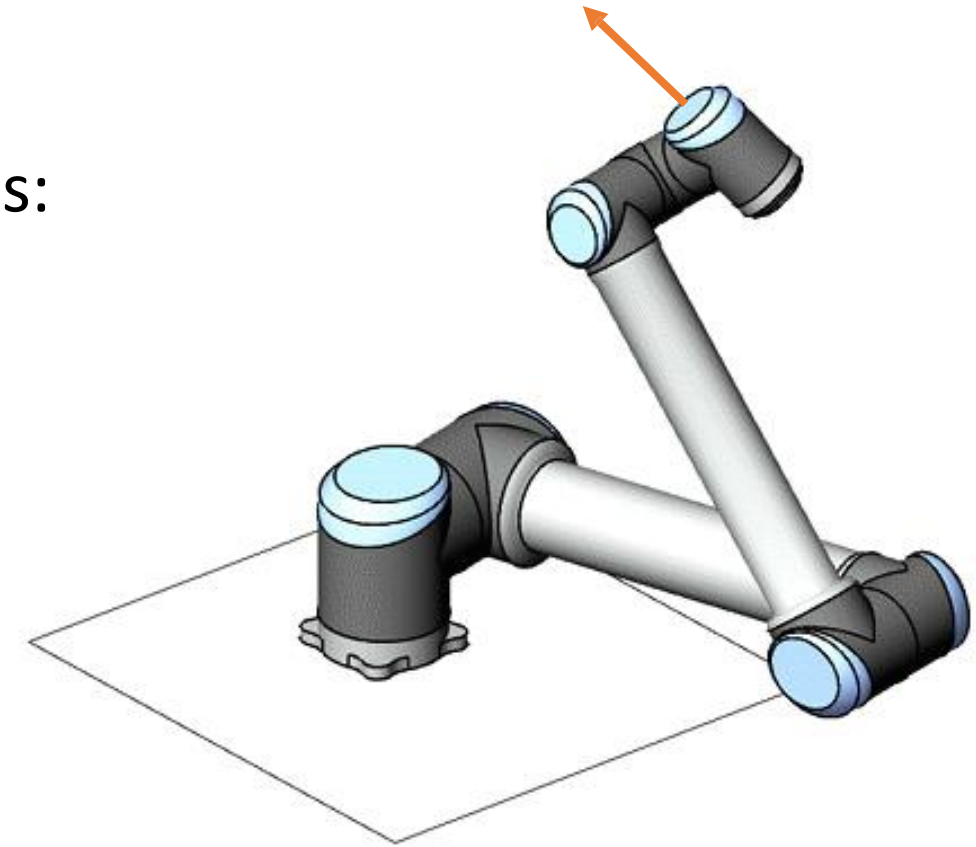




(b) The joint frames of the UR5 robot. The join position is $\boldsymbol{q} = [0, -\frac{\pi}{2}, -\frac{\pi}{2}, -\frac{\pi}{2}, \frac{\pi}{2}, 0]^{\mathrm{T}}$

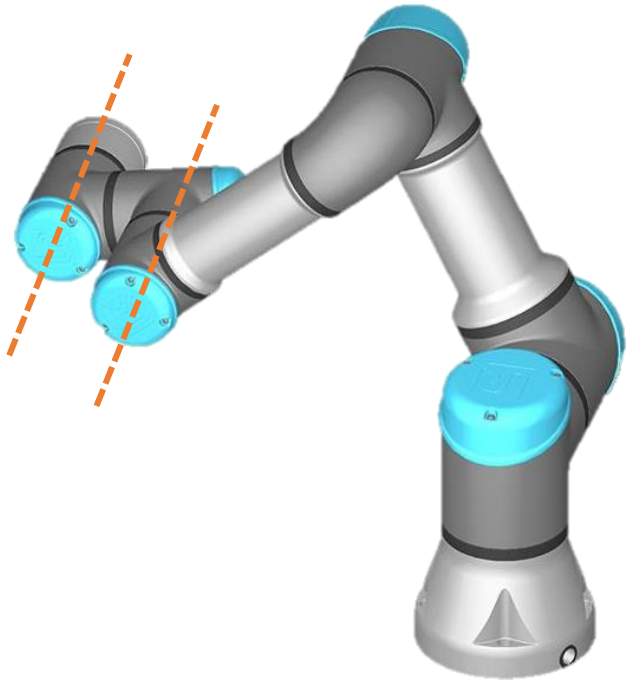
Inverse kinematic solutions and singularities

- These cobots have a simple inverse position kinematic problem allowing up to eight different solutions.
- They have also 3 types of singularities:
  - Wrist singularity
  - Elbow singularity
  - Shoulder singularity
- These singularities are slightly different from previously described industrial robot singularities.



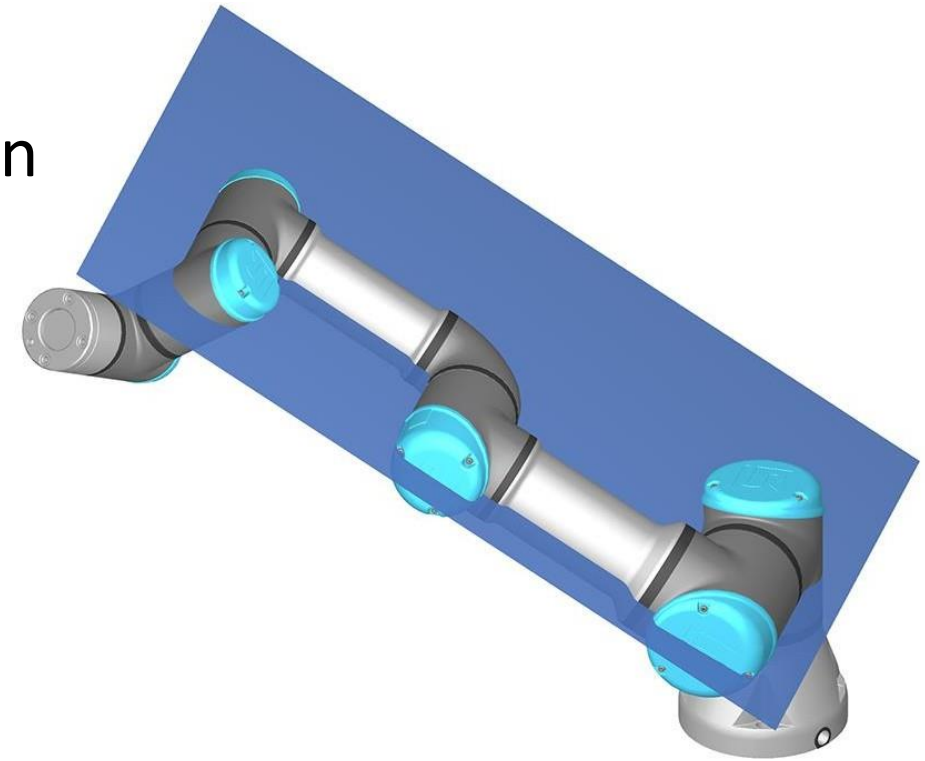
Wrist singularity

- The most troublesome of them is the wrist singularity.
- It happens when the axes of joints 4 and 6 are parallel, when:
  - **θ5° = 0, θ5 = ±180°, or θ5 = ±360°.**
- Of course, problems occur even when these two axes are close to parallel.
- Furthermore, in a wrist singularity, the mechanism consisting of joints 2, 3, 4, and 5, can move while the end-effector remains stationary.



Elbow singularity

The elbow singularity occurs when the arm is fully stretched.
Theoretically, this singularity can also occur when the arm is fully folded, but even though joint 3 can rotate ±360°, the robot links collide each other well before reaching the fully folded configuration.
More precisely, the elbow singularity occurs when the axes of joints 2, 3 and 4 are coplanar, i.e., when $\theta 3 = 0°$.
This singularity is by far the least problematic one and the simplest to prevent.

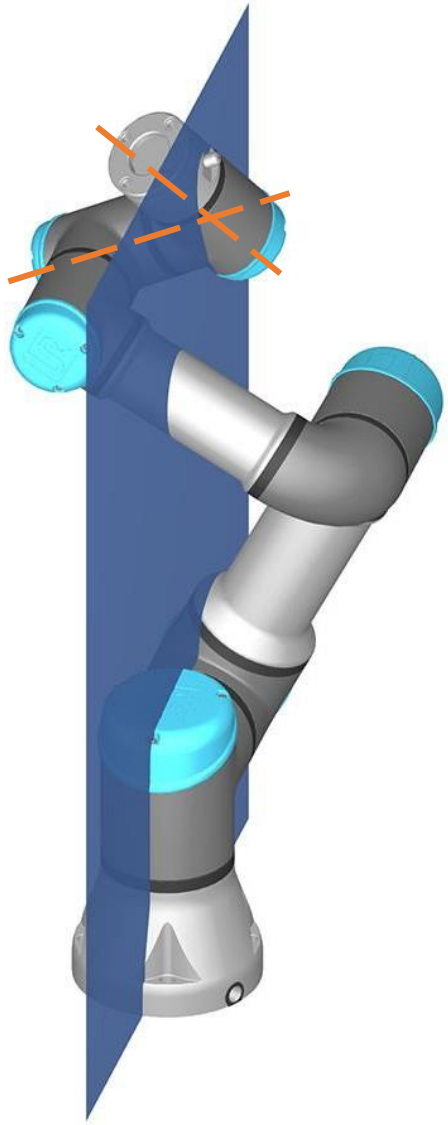


Shoulder singularity

Shoulder singularity occurs when the intersection point of the axes of joints 5 and 6 lie in one plane with the axes of joints 1 and 2.

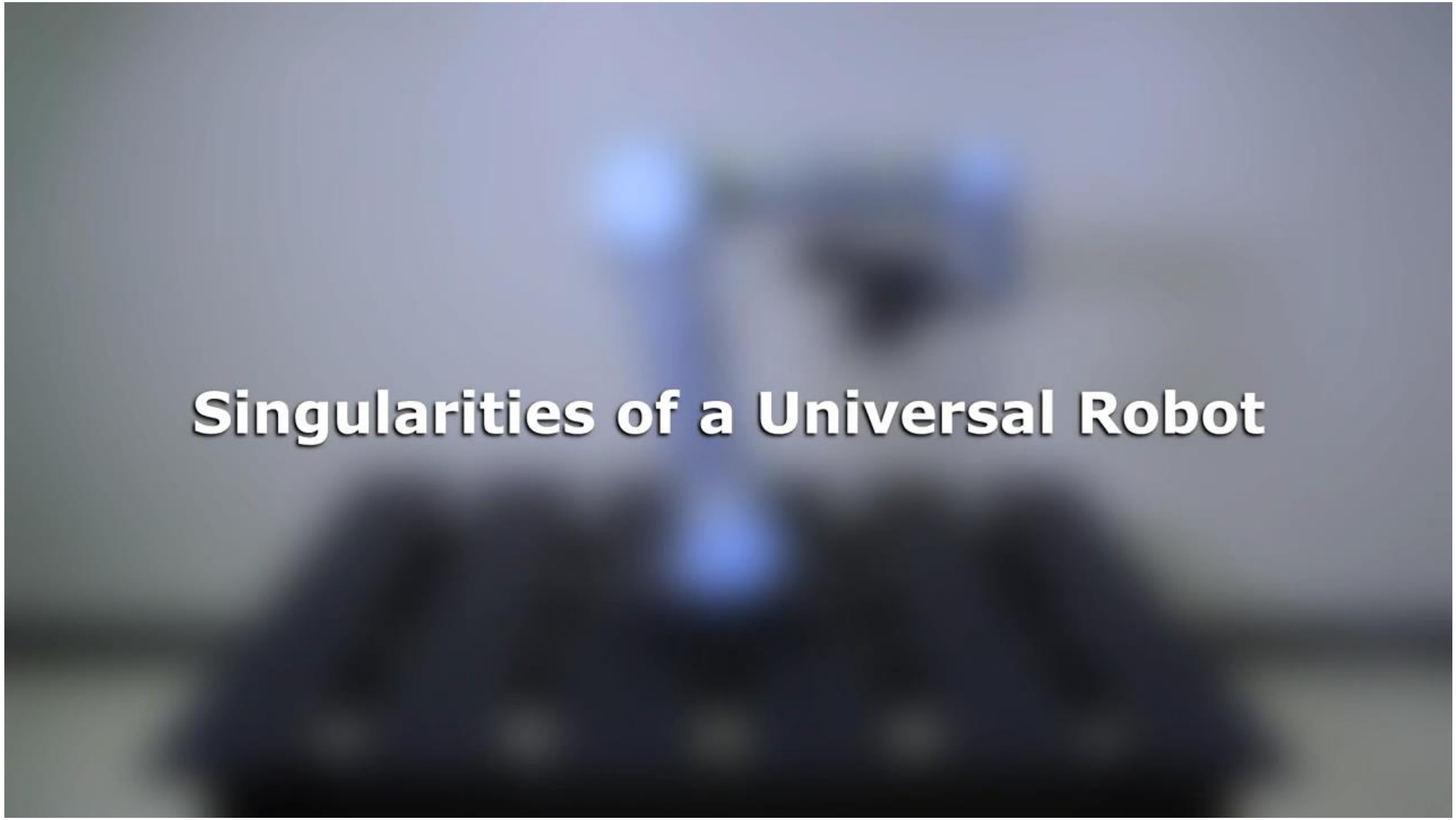
In a shoulder singularity, the two possible solutions for ϑ_1 coalesce.

This singularity is relatively easy to prevent. It must, however, be considered in every risk assessment for a cobot installation, due to the sudden rapid motion produced by joint 1, when passing close to a shoulder singularity.

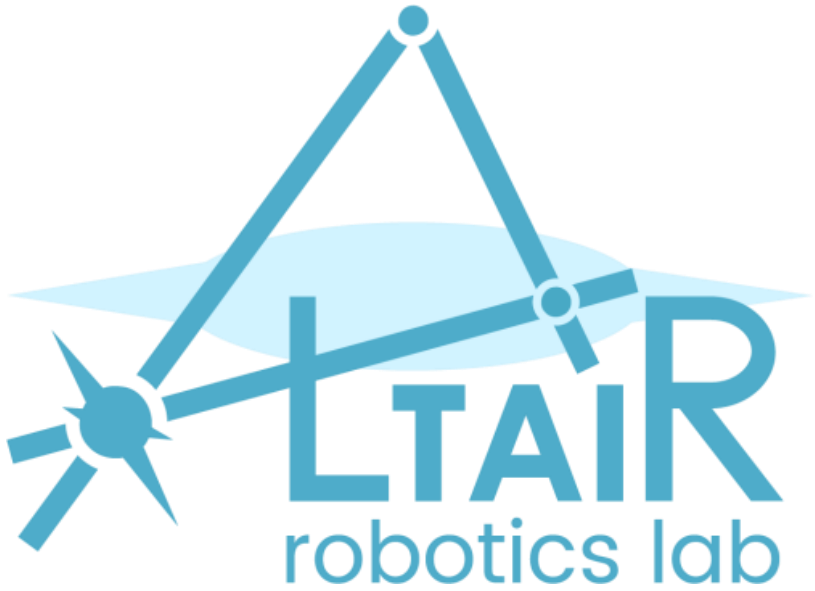


UR5 Singularities

https://www.youtube.com/watch?v=6Wmw4lUHlX8



Questions?



The contents of these slides are partially based on: