INDUSTRIAL ROBOTICS

Prof. Dr.-Ing. Thomas Wich

3 - KINEMATIK

Objectives of the chapter

TECHNISCHE HOCHSCHULE LÜBECK

- We want:
  - Calculate where a robot moves when we set certain joint angles (forward transformation, forward kinematics)
  - Calculate how we need to adjust joint angles to reach a certain position (backward transformation, inverse kinematics)

Bildquelle: https://github.com/jonabalzer/braccio_description

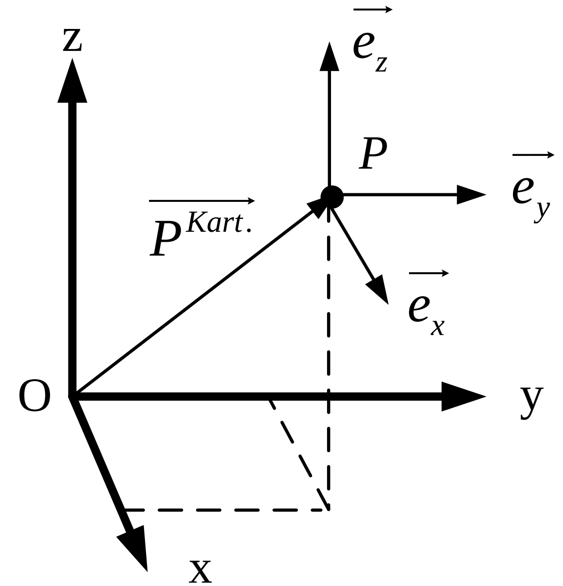
Contents of the lecture

TECHNISCHE HOCHSCHULE LÜBECK

- Structure
    - Coordinate systems
    - Reference systems
    - Transformations between reference systems
    - The Denavit-Hardenberg process
    - Inverse Kinematics / Backward Transformation
    -

Coordinate systems

- Origin
  - Reference point to which the coordinates refer
  - Coordinates there take the value 0
- Unit vectors
  - Vectors of length 1 tangential to the coordinate axes
  - n vectors for n-dimensional KS
- Vectors
  - Linear combination of unit vectors

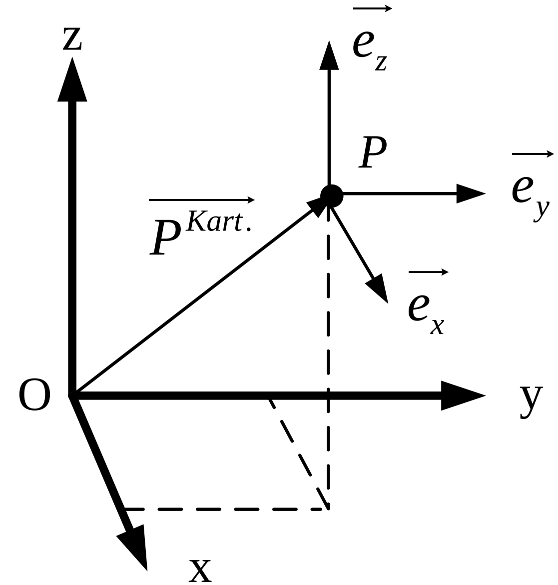


$$\overrightarrow{P^{Kart.}} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = x\overrightarrow{e_x} + y\overrightarrow{e_y} + z\overrightarrow{e_z}$$

TECHNISCHE HOCHSCHULE LÜBECK

Koordinatensysteme

- Cartesian coordinate system
- 



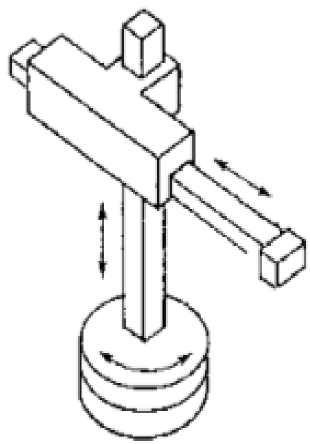
$$\vec{P}^{Kart.} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = x\vec{e}_x + y\vec{e}_y + z\vec{e}_z$$

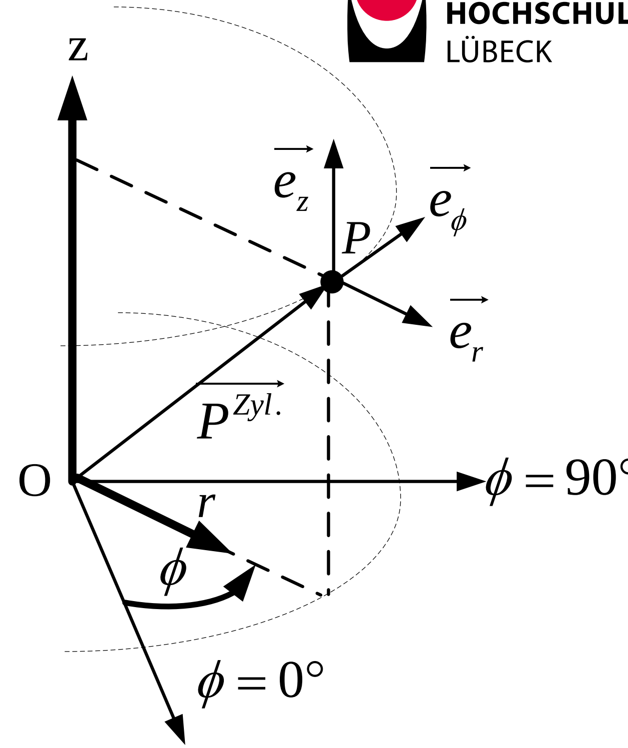
CS	Dist.	Angles	Unit vectors	coord.
Cartesian	3	0	$\vec{e}_x,\ \vec{e}_y,\ \vec{e}_z$	x, y, z

TECHNISCHE HOCHSCHULE LÜBECK

Coordinate systems

- Cylinder coordinate system





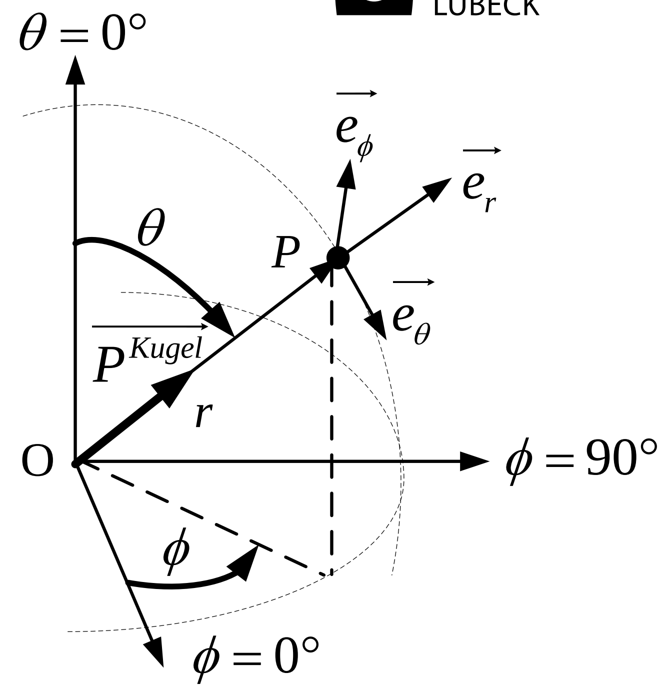
$$\overrightarrow{P^{Zyl.}} = \begin{pmatrix} r \\ \phi \\ z \end{pmatrix} = r\overrightarrow{e_r} + \phi\overrightarrow{e_\phi} + z\overrightarrow{e_z}$$

CS	Dist.	Angles	Unit vectors	coord.
Cylinder	2	1	$\overrightarrow{e_r}, \overrightarrow{e_\phi}, \overrightarrow{e_z}$	r, Φ, z

TECHNISCHE HOCHSCHULE LÜBECK

Coordinate systems

- Spherical Coordinate System



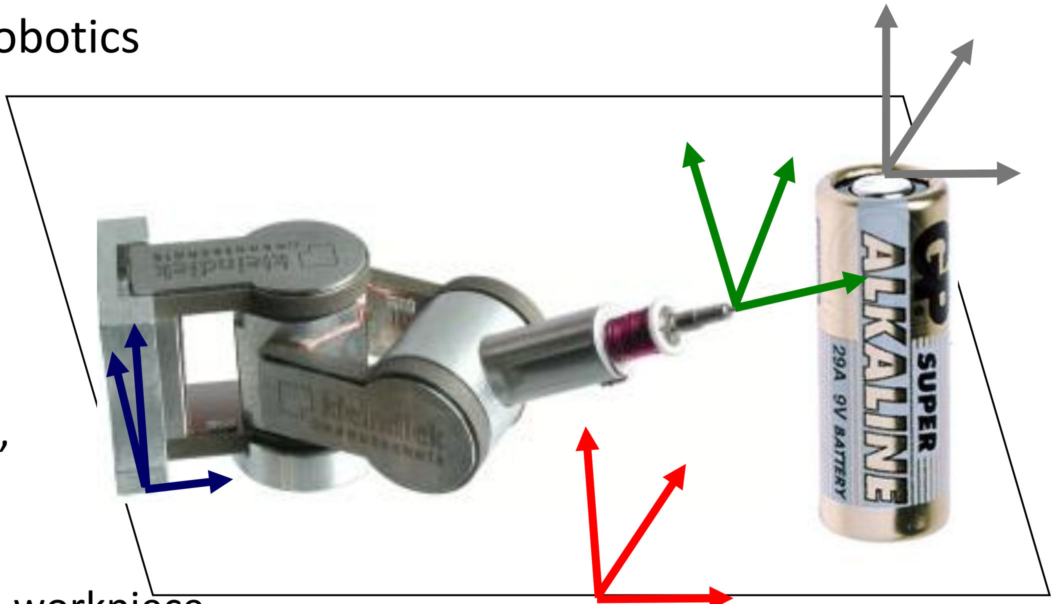
$$\overrightarrow{P^{Kugel}} = \begin{pmatrix} r \\ \phi \\ \theta \end{pmatrix} = r\overrightarrow{e_r} + \phi\overrightarrow{e_\phi} + \theta\overrightarrow{e_\theta}$$

CS	Dist.	Angles	Unit vectors	coord.
spheric al	1	2	$\overrightarrow{e_r}, \overrightarrow{e_\phi}, \overrightarrow{e_\theta}$	r, Φ, θ

Reference systems

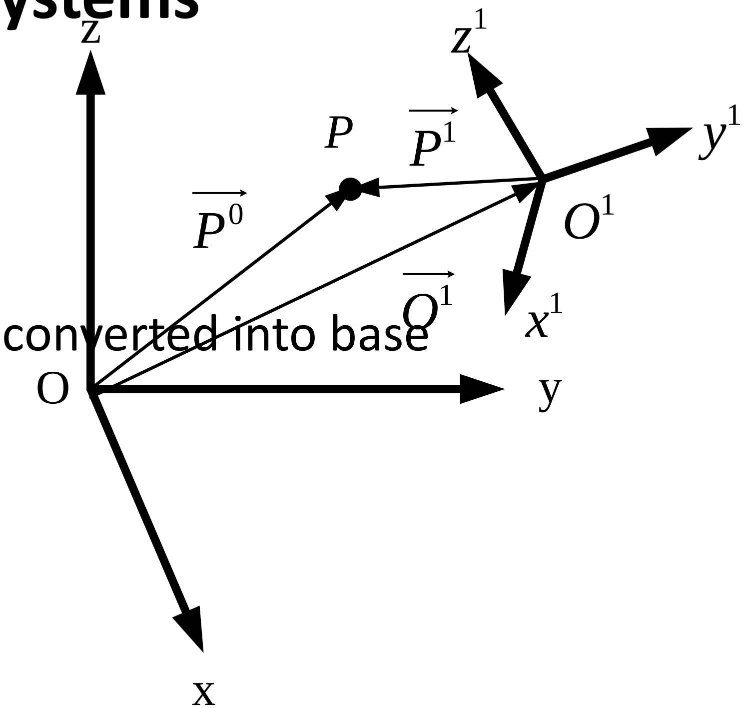
TECHNISCHE
HOCHSCHULE
LÜBECK

- Common reference coordinates in robotics
  - World coordinates
    - Firmly connected to the surroundings
  - Base coordinates of the robot
  - Hand flange coordinates (Tool Connector Point, TCP)
    - connected to the hand flange,
    - are also moved
  - Workpiece coordinates
    - Coordinates connected to the workpiece
    - Workpiece and thus also its CS can be movable!
  - Convention: When working with coordinates and systems, always specify the reference system, e.g.:
  -



$$\overrightarrow{P^{TCP}},\ \overrightarrow{P^{Welt}},\ \overrightarrow{P^{WST}}$$

Transformations between reference systems



- Transformations between reference Coordinate systems
  - How can a location in the workpiece KS be converted into base coordinates?
  - Transformation consists of:
  - Translation:
    - Shift of KS 1 compared to KS 0
    - Description by a vector
  - Rotation
    - Rotation of the CS 1 compared to the CS 0
    - Description by a rotation matrix

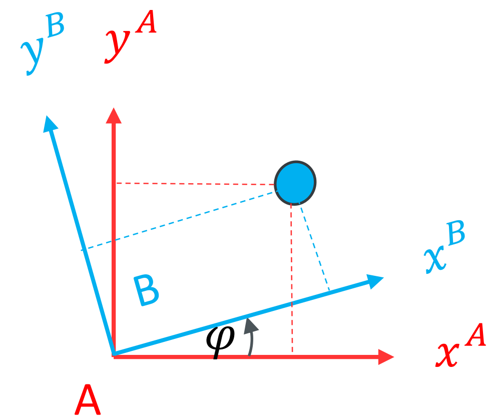
$$\overrightarrow{P^o} = \overrightarrow{O^1} + \mathbf{R_1^0}\overrightarrow{P^1}$$

- Goal:
  Development of a transformation matrix that combines translation and rotation.

Transformations between reference systems

- 2D rotation
  - The transformation of a CS rotated by the angle ϕ can be represented by a simple 2 x 2 rotation matrix:
  - This allows a point P^B to be transferred to CS A:

$$T_B^A = R(\varphi) = \begin{pmatrix} cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix}$$



$$p^A = T_B^A\, p^B = \begin{pmatrix} cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix} \begin{pmatrix} x_P^B \\ y_P^B \end{pmatrix}$$

TECHNISCHE HOCHSCHULE LÜBECK

Transformations between reference systems

- 2D rotation
  - Reverse transformation from CS A to CS B:

$$T_A^B = R(\varphi)^T = \begin{pmatrix} cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{pmatrix} \qquad p^B = T_A^B\, p^A = \begin{pmatrix} cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{pmatrix} \begin{pmatrix} x_P^A \\ y_P^A \end{pmatrix}$$

- The rotation matrices $R(\phi)^T$ and $R(\phi)$ are orthonormal to each other, thus:

$$R(\varphi)^T\, R(\varphi) = I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

- The determinant is 1.

TECHNISCHE HOCHSCHULE LÜBECK

Transformations between reference systems

- 3D rotation around an elementary axis
  - Similar to 2D rotation, the 3D rotation matrices result in simple rotations (elementary rotations)
  - Rotation around the x-axis:

$$T_B^A = R(x, \propto) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos \propto & -sin \propto \\ 0 & sin \propto & cos \propto \end{pmatrix}$$

  -
  - Rotation around the y-axis:

$$T_B^A = R(y, \beta) = \begin{pmatrix} cos\beta & 0 & sin\beta \\ 0 & 1 & 0 \\ -sin\beta & 0 & cos\beta \end{pmatrix}$$

  -
  - Rotation around the z-axis:
  -

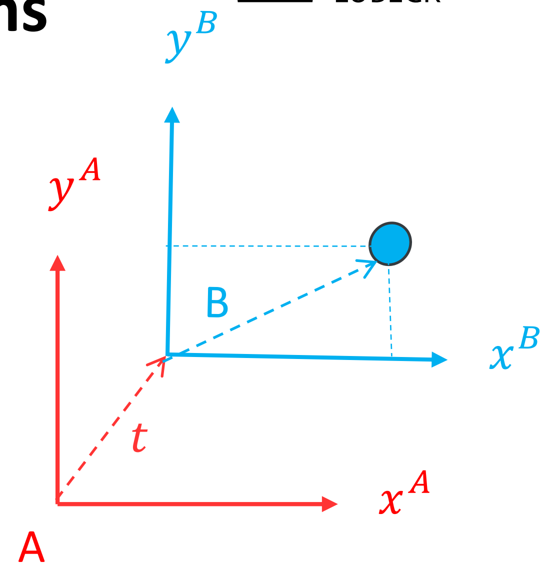
$$T_B^A = R(z, \theta) = \begin{pmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Transformations between reference systems

- 3D rotation around any axis
  - Rotation matrix
  - $$T_B^A = R(x, \propto) = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}$$
  - Coordinate transformation of a point p from KS B to KS A:
  - $$p^A = T_B^A\, p^B \quad = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{pmatrix} x_P^B \\ y_P^B \\ z_P^B \end{pmatrix}$$
  - Back transformation:
  - $$p^B = {T_B^A}^T\, p^A = T_A^B \quad p^A$$

TECHNISCHE HOCHSCHULE LÜBECK

Transformations between reference systems



- Translation of coordinate systems
  - The translation corresponds to a simple addition of the displacement vector t:

$$p^A = p^B + t$$

  -
  - Problem:
    - Rotation transformation is achieved by matrix multiplication
    - Translation transformation is achieved by vector addition
    - requires several arithmetic operations
    - Reverse transformation is time-consuming to calculate
  - Solution: Use of homogeneous coordinates for the closed calculation of transformations
  -

TECHNISCHE HOCHSCHULE LÜBECK

Transformations between reference systems

- Homogeneous 3D translation
  - Approach: Another "virtual" coordinate is added to the matrices
  - Example: 2D Translation
  - $$p^A = p^B + t = \begin{pmatrix} x_P^B \\ y_P^B \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \quad \Rightarrow \quad \tilde{p}^A = \widetilde{Tl}_B^A \, \tilde{p}^B = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_P^B \\ y_P^B \\ 1 \end{pmatrix}$$

Identity matrix

t

$$\boxed{\tilde{p}^A} = \widetilde{Tl}_B^A \, \boxed{\tilde{p}^B} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \boxed{\begin{pmatrix} x_P^B \\ y_P^B \\ 1 \end{pmatrix}}$$

Homogeneous coordinates

TECHNISCHE HOCHSCHULE LÜBECK

Transformations between reference systems

- Homogeneous 3D translation
  - General wording for translation (3D)
  -

$$\begin{pmatrix} p^A \\ 1 \end{pmatrix} = \begin{pmatrix} I_{3x3} & t \\ 0_{1x2} & 1 \end{pmatrix} \begin{pmatrix} p^B \\ 1 \end{pmatrix}$$

  - Homogeneous coordinates
  -

$$\tilde{p}^A = \begin{pmatrix} I_{3x3} & t \\ 0_{1x3} & 1 \end{pmatrix} \tilde{p}^B$$

  - Translation matrix with the translation vector in homogeneous coordinates:
  -

$$\widetilde{Tl}(t) = \begin{pmatrix} I_{3x3} & t \\ 0_{1x3} & 1 \end{pmatrix}$$

TECHNISCHE HOCHSCHULE LÜBECK

Transformations between reference systems

- Homogeneous 3D rotation
  - Approach: the translation vector is zero, the unit matrix is replaced by the rotation matrix.

$$\tilde{R} = \begin{pmatrix} R & 0_{3x1} \\ 0_{1x3} & 1 \end{pmatrix}$$

  -
  - Rotation with homogeneous coordinates:
  -

$$\begin{pmatrix} p^A \\ 1 \end{pmatrix} = \begin{pmatrix} R & 0_{3x1} \\ 0_{1x3} & 1 \end{pmatrix} \begin{pmatrix} p^B \\ 1 \end{pmatrix} = \begin{pmatrix} Rp^B \\ 1 \end{pmatrix}$$

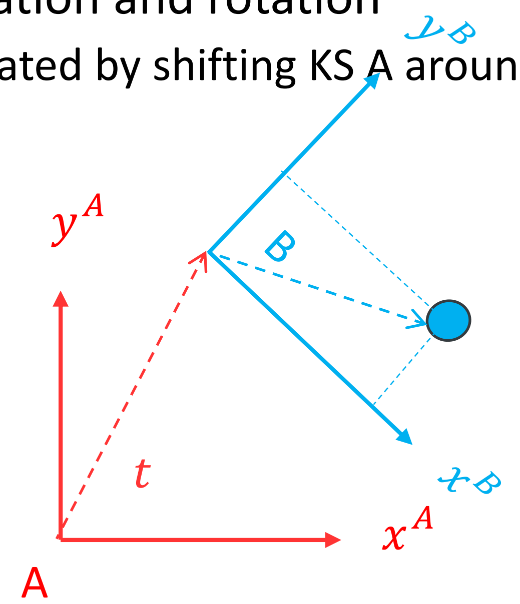
$$\tilde{p}^A = \begin{pmatrix} R & 0_{3x1} \\ 0_{1x3} & 1 \end{pmatrix} \tilde{p}^B = \tilde{R}\,\tilde{p}^B$$

TECHNISCHE HOCHSCHULE LÜBECK

Transformations between reference systems

- Homogeneous 3D transformation
  - Concatenation of transformations – translation and rotation
    - Intermediate coordinate system KS* is created by shifting KS A around the vector t:

$$T_*^A = \widetilde{Tl}(t)$$



    -
    - By rotating the intermediate coordinate system KS* is created by KS B:
    - $$T_B^* = \tilde{R}$$

    - This can be summarized to:
    - $$T_B^A = T_*^A \, T_B^* = \widetilde{Tl}(t)\tilde{R} = \begin{pmatrix} I_{3x3} & t \\ 0_{1x3} & 1 \end{pmatrix} \begin{pmatrix} R & 0_{3x1} \\ 0_{1x3} & 1 \end{pmatrix} = \begin{pmatrix} R & t \\ 0_{1x3} & 1 \end{pmatrix}$$

TECHNISCHE HOCHSCHULE LÜBECK

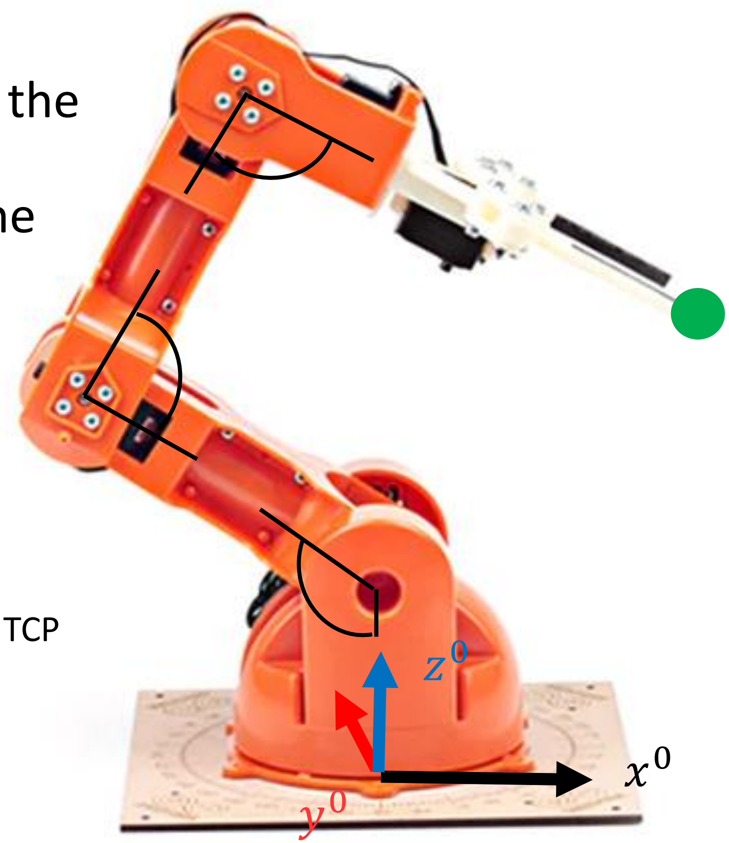
Transformations between reference systems

- Homogeneous 3D Transformation Summary
  - Homogeneous transformation matrices have the form:
  - $$T = \begin{pmatrix} R & t \\ 0_{1x3} & 1 \end{pmatrix}$$
  - Inverse transformation matrix:
    $$T^{-1} = \begin{pmatrix} R^T & -R^T t \\ 0_{1x3} & 1 \end{pmatrix}$$
  - $$T\,T^{-1} = \begin{pmatrix} R & t \\ 0_{1x3} & 1 \end{pmatrix} \begin{pmatrix} R^T & -R^T t \\ 0_{1x3} & 1 \end{pmatrix} = \begin{pmatrix} I_{3x3} & 0_{3x1} \\ 0_{1x3} & 1 \end{pmatrix} = I_{4x4}$$
  - Several transformations can be combined by multiplying the homogeneous transformation matrices.
    This saves computing time in the robot controller!
  - $$T_C^A = T_B^A\, T_A^B$$

TECHNISCHE HOCHSCHULE LÜBECK

The Denavit-Hardenberg-Transformation

- Problem formulation:
  We are looking for a method with which the position of the TCP in the base or world coordinate system can be calculated if the joint angles of the robot are known.
- This is called the so-called forward transformation or forward kinematics.
-
- Notes:
  - In the picture, the gripper tip is used instead of the TCP
  - The axes bear the CS number in the supersciptum
  -

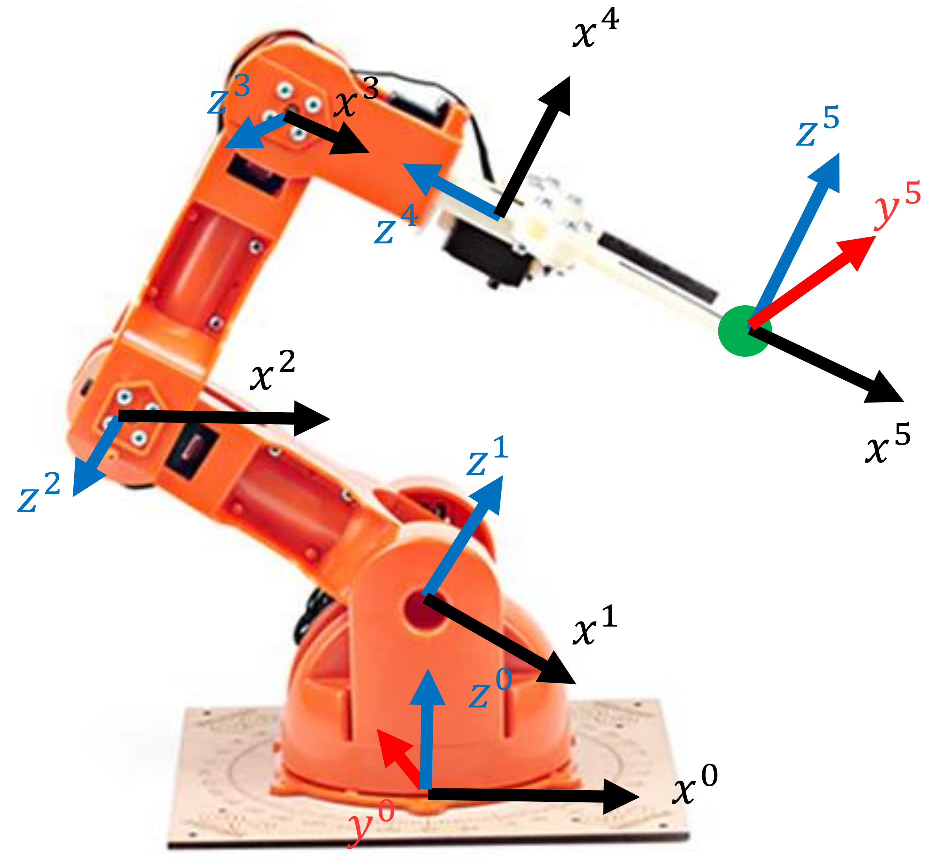


Bildquelle: https://store.arduino.cc/products/tinkerkit-braccio-robot

TECHNISCHE HOCHSCHULE LÜBECK

The Denavit-Hardenberg-Transformation

- Approach (I):
- Each joint is provided with its own coordinate system
  - Thus, each member has 2 coordinate systems
- Hint:
  - the basic coordinate system and the tcp coordinate system are usually predefined
  -

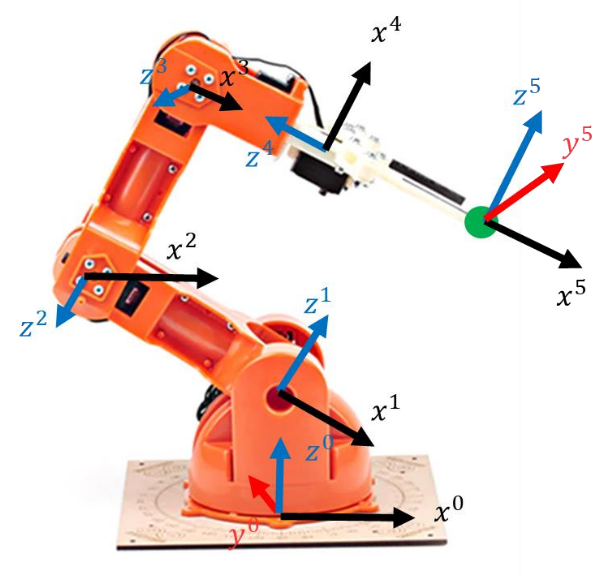


Bildquelle: https://store.arduino.cc/products/tinkerkit-braccio-robot

TECHNISCHE HOCHSCHULE LÜBECK

The Denavit-Hardenberg-Transformation

- Approach (II):
  - For each member, the DH Transformation parameters are determined to transform from the i-th to the i-1-th joint:
    - Translation along the z-axis: d
    - Rotation around the z-axis: δ
    - Translation along the x-axis: a
    - Rotation around the x-axis: α



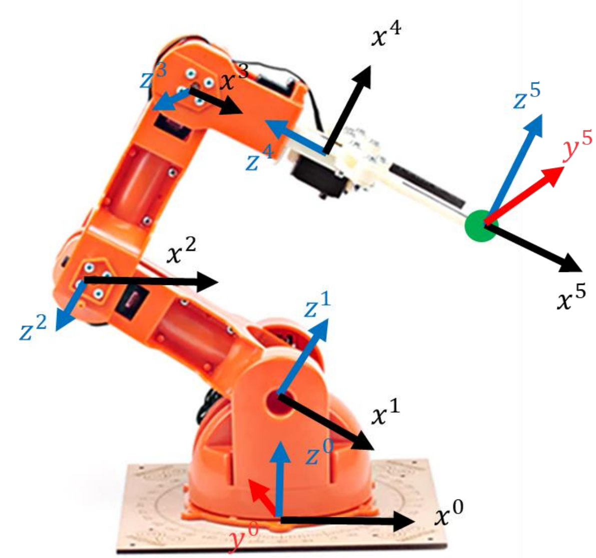
  - This gives the DH transformation matrix for one member:
  -

$$T_i^{i-1} = \widetilde{Tl}(z,d)^* \tilde{R}(z,\delta) * \widetilde{Tl}(x,a) * \tilde{R}(x,\alpha)$$

Bildquelle: https://store.arduino.cc/products/tinkerkit-braccio-robot

TECHNISCHE HOCHSCHULE LÜBECK

The Denavit-Hardenberg-Transformation



- Approach (III):
  - The DH Transformation Matrix:

$$T_i^{i-1} = \widetilde{Tl}(z,d)^* \, \tilde{R}(z,\delta) * \widetilde{Tl}(x,a) * \tilde{R}(x,\alpha)$$

  - Can be easily converted into the following form:

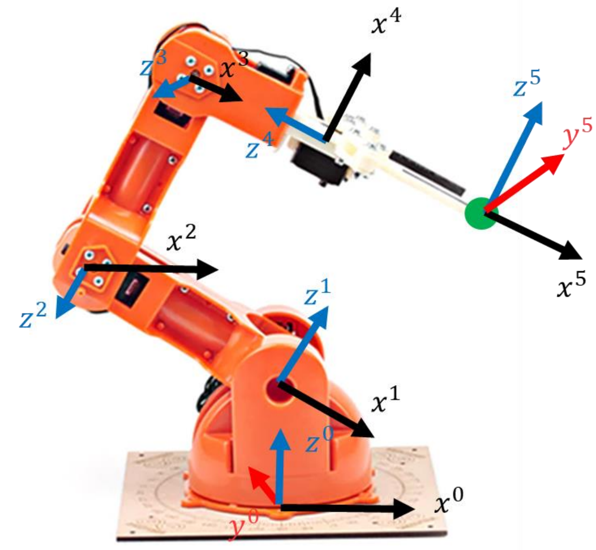
$$T_i^{i-1} = \begin{pmatrix} \cos\delta_i & -\sin\delta_i\cos\alpha_i & \sin\delta_i\sin\alpha_i & a_i\cos\delta_i \\ \sin\delta_i & \cos\delta_i\cos\alpha_i & -\cos\delta_i\sin\alpha_i & a_i\sin\delta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Bildquelle: https://store.arduino.cc/products/tinkerkit-braccio-robot

TECHNISCHE HOCHSCHULE LÜBECK

The Denavit-Hardenberg-Transformation

- Approach (IV):
  - In total, there are four DH parameters per CS: $\boldsymbol{a_i}$, $\boldsymbol{\alpha_i}$, $\boldsymbol{d_i}$ und $\boldsymbol{\delta_i}$
  - Two of the four DH parameters are always constants: $\boldsymbol{a_i}$, $\boldsymbol{\alpha_i}$
  - Only one of the other parameters $\boldsymbol{d_i}$ and $\boldsymbol{\delta_i}$ is a variable, this depends on the type of movement at the joint i:
    - Translation: $\boldsymbol{d_i}$ variable and $\boldsymbol{\delta_i}$ constant
    - Rotation: $\boldsymbol{d_i}$ constant und $\boldsymbol{\delta_i}$ variable
  - This is the result of the DH analysis: List of individual parameters in a table:



Glied	d_i	δ_i	a_i	α_i
1	…	…	…	…
2	…	…	…	…
3	…	…	…	…
4	…	…	…	…
5	…	…	…	…

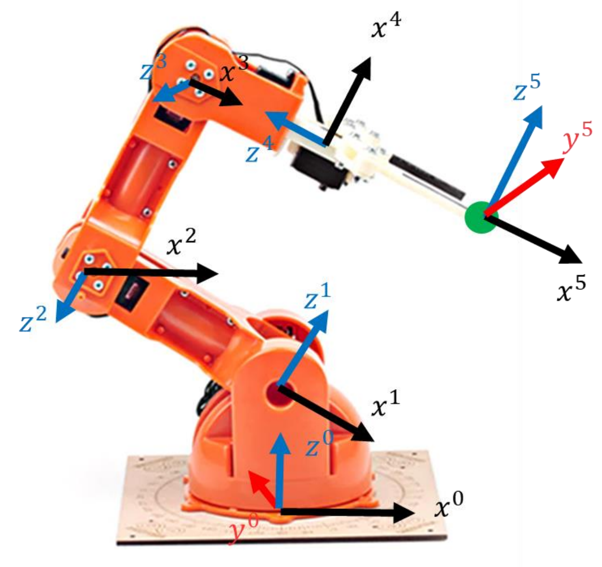
Bildquelle: https://store.arduino.cc/products/tinkerkit-braccio-robot

The Denavit-Hardenberg-Transformation

- Approach (V):
  - The multiplication of the transformation matrices results in the overall transformation rule:

$$\begin{pmatrix} x^0 \\ y^0 \\ z^0 \\ 1 \end{pmatrix} = T_1^0 * T_2^1 * \ldots * T_n^{n-1} * \begin{pmatrix} x^n \\ y^n \\ z^n \\ 1 \end{pmatrix}$$

  - Dies ist die:
    - Forward transformation or
    - Forward kinematics
    - The DH parameters are also required for backward transformation / inverse kinematics.

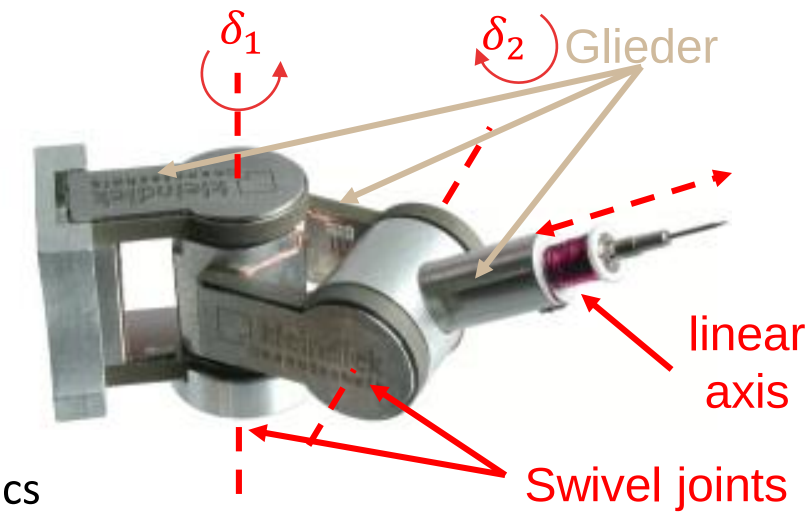


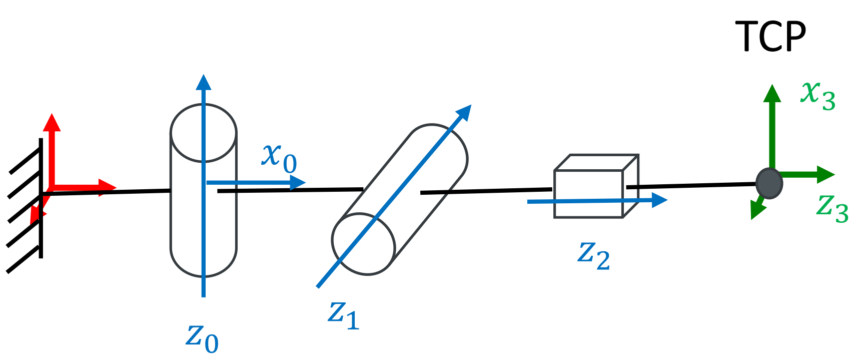
Bildquelle: https://store.arduino.cc/products/tinkerkit-braccio-robot

TECHNISCHE HOCHSCHULE LÜBECK

The Denavit-Hardenberg-Transformation

- The DH Process – Step 1
- Analysis of the kinematic chain
    - Links are represented by strokes and by joints (T/R) together connected.
    - If necessary, "straighten" the kinematics
    -
- Each joint has its own coordinate system
    - Z-axis points in the direction of positive movement/rotation
    - X-axis will be set later (if not specified)
    - Y-axis can be omitted!

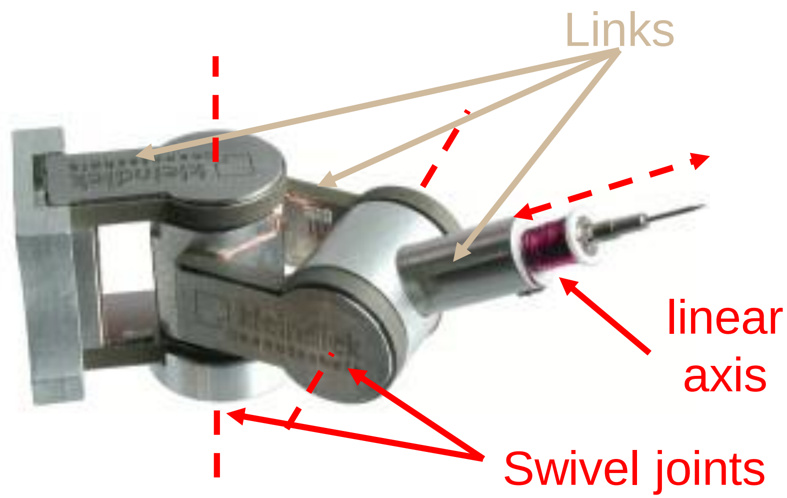


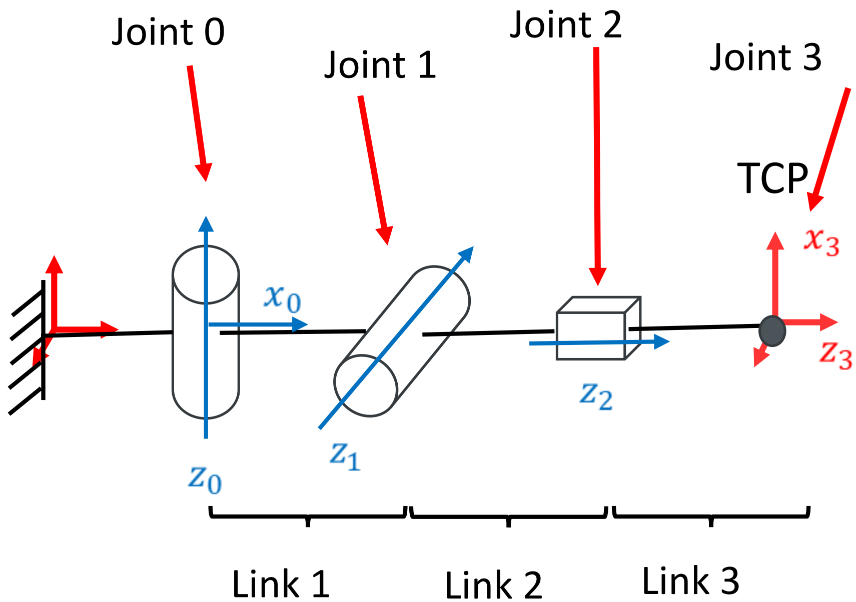


TECHNISCHE HOCHSCHULE LÜBECK

The Denavit-Hardenberg-Transformation

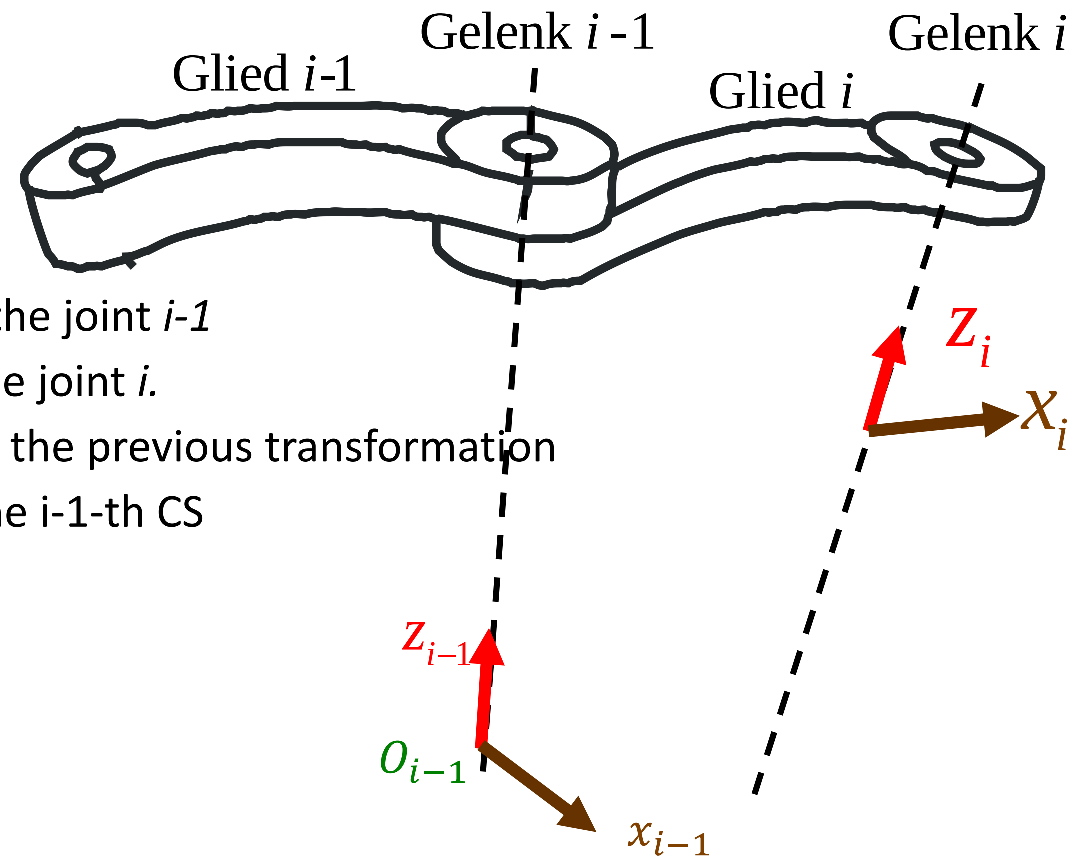
- The DH Process – Step 1
  - Benennung der Gelenke und Glieder
  - Red World KS is not taken into account here
  - Enter angles and distances if necessary





The Denavit-Hardenberg-Transformation

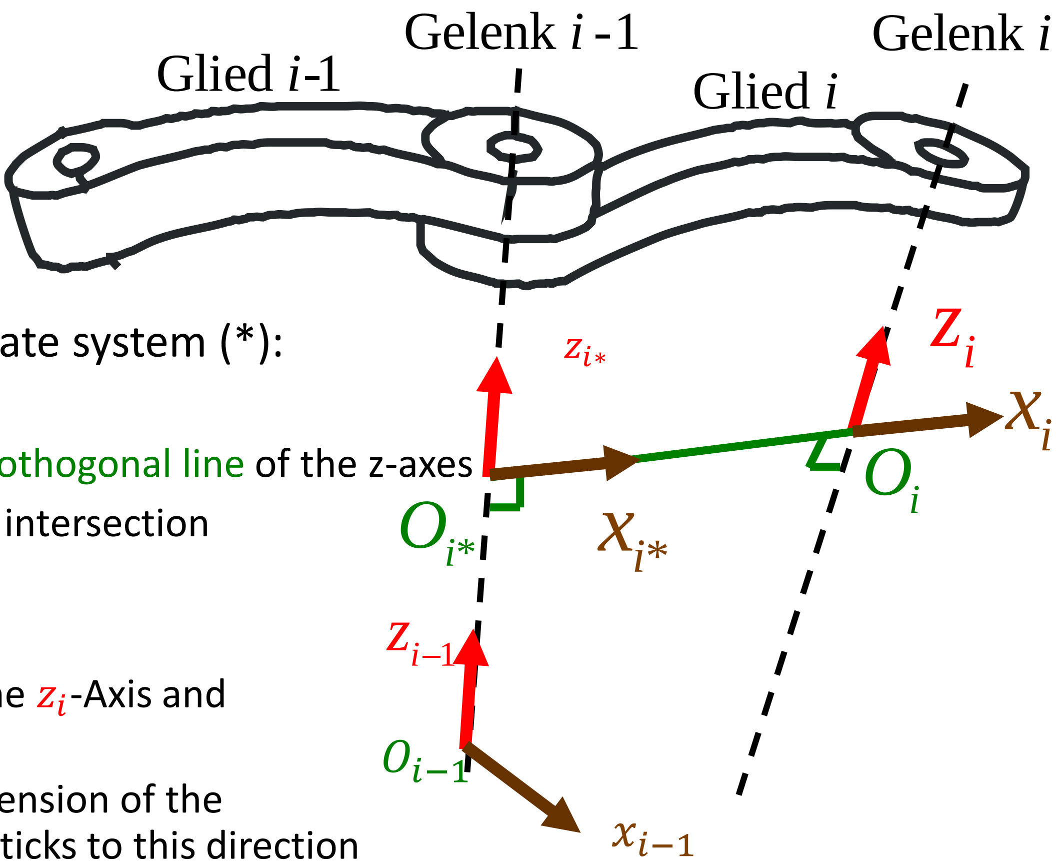
- The DH Process – Step 2
  - Naming of the links and joints:
  - On the link i are the joints i-1 and i
  -
  - The axis $\boldsymbol{z_{i-1}}$ lies on the axis of the joint *i-1*
  - *The axis* $\boldsymbol{z_i}$ lies on the axis of the joint *i.*
  - The axis x_{i-1} is determined by the previous transformation
  - Thus also the origin O_{i-1} of the i-1-th CS
  -
  - The position of the axis x_i is determined in the next step.



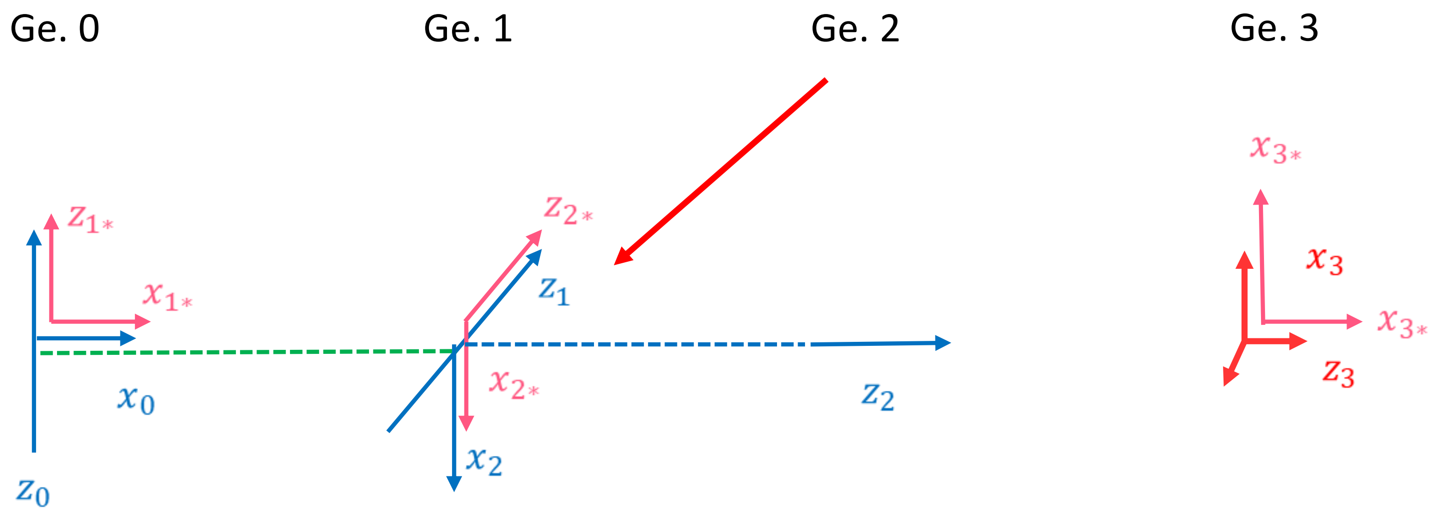
TECHNISCHE HOCHSCHULE LÜBECK

The Denavit-Hardenberg-Transformation

- The DH Process – Step 2
  Definition of coordinate systems:
  - The axles x_i and x_{i*} lie on the common othogonal line of axles z_{i-1} and z_i .
  - Specifying the auxiliary coordinate system (*):
    - z_{i*} is located on the z_{i-1} –axis
    - x_{i*} is located on the common othogonal line of the z-axes
    - The origin of KS O_{i*} lies at the intersection of z_{i*} and x_{i*}
  - Definition of the i-th CS:
    - O_i lies at the intersection of the z_i-Axis and the common othogonal line
    - The axis x_i is based on the extension of the common othogonal line and sticks to this direction



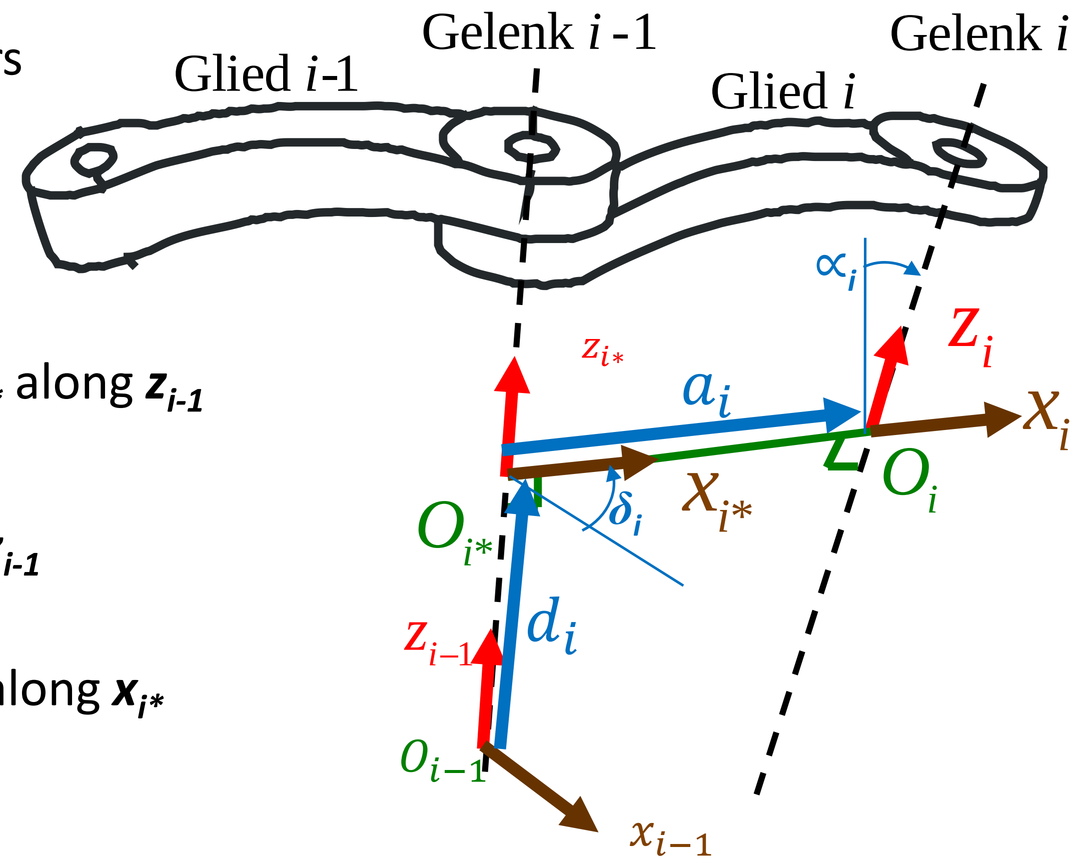
The Denavit-Hardenberg-Transformation



- The DH Process – Step 2
  Definition of coordinate systems:
  - Origin of joint 2 migrates to the origin of joint 1 (special case)
  -

The Denavit-Hardenberg-Transformation

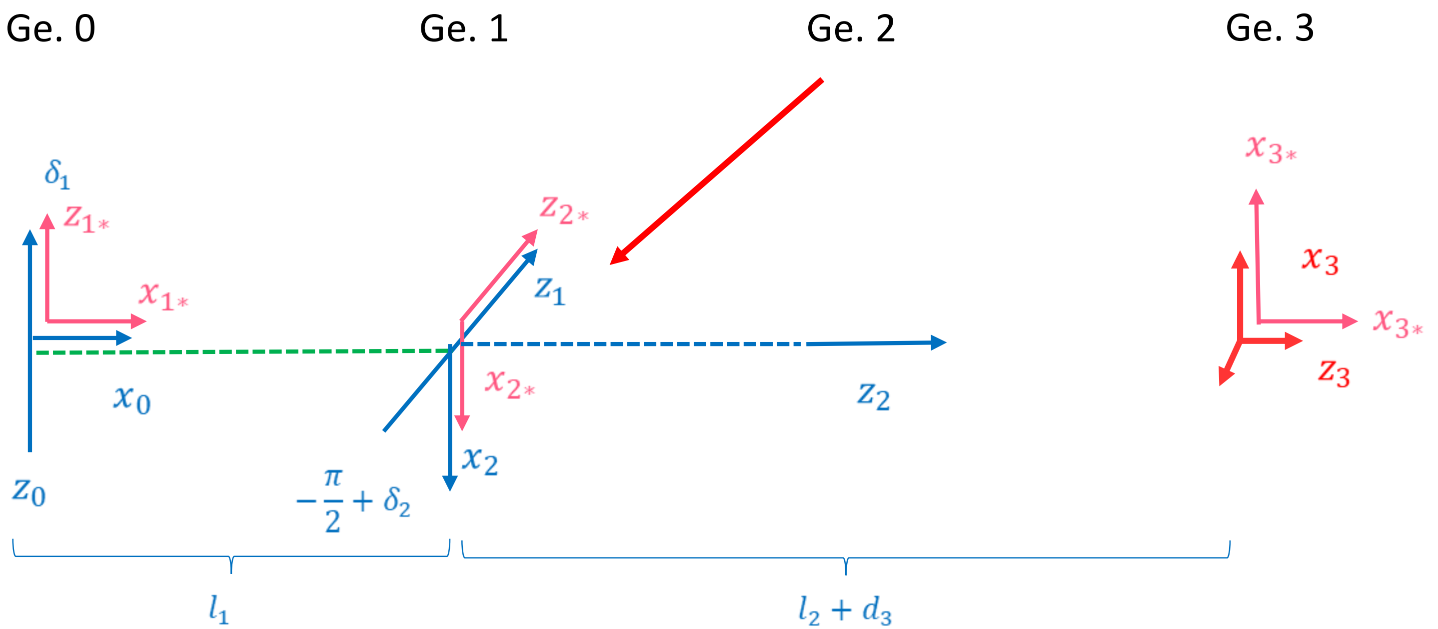
- The DH Process – Step 3
  Determination of DH parameters
-



- $\boldsymbol{d_i}$: Distance from $\boldsymbol{O_{i-1}}$ to $\boldsymbol{O_{i*}}$ along $\boldsymbol{z_{i-1}}$
- $\boldsymbol{\delta_i}$: Angle of $\boldsymbol{x_{i-1}}$ to $\boldsymbol{x_{i*}}$ over $\boldsymbol{z_{i-1}}$
- $\boldsymbol{a_i}$: Distance from $\boldsymbol{O_{i*}}$ to $\boldsymbol{O_i}$ along $\boldsymbol{x_{i*}}$
- $\propto_i$: Angle of $\boldsymbol{z_{i*}}$ to $\boldsymbol{z_i}$ over $\boldsymbol{x_i}$

TECHNISCHE HOCHSCHULE LÜBECK

The Denavit-Hardenberg-Transformation



- The DH Process – Step 3
  Determination of DH parameters:

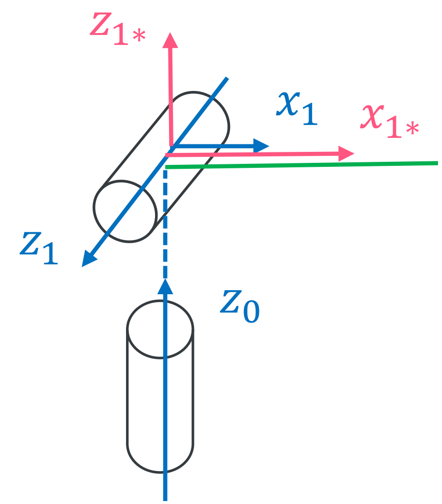
Glied	d_i	δ_i	a_i	α_i
1	0	δ_1	l_1	$-\pi/2$
2	0	$+\pi/2 + \delta_2$	0	$+\pi/2$
3	l_2+d_3	0	0	0

The Denavit-Hardenberg-Transformation

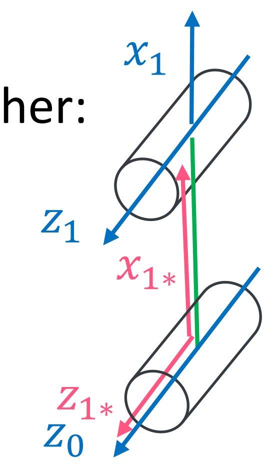
TECHNISCHE HOCHSCHULE LÜBECK

- The DH Process – Step 2
  Definition of coordinate systems: special cases

  - orthogonal z-axes
  -



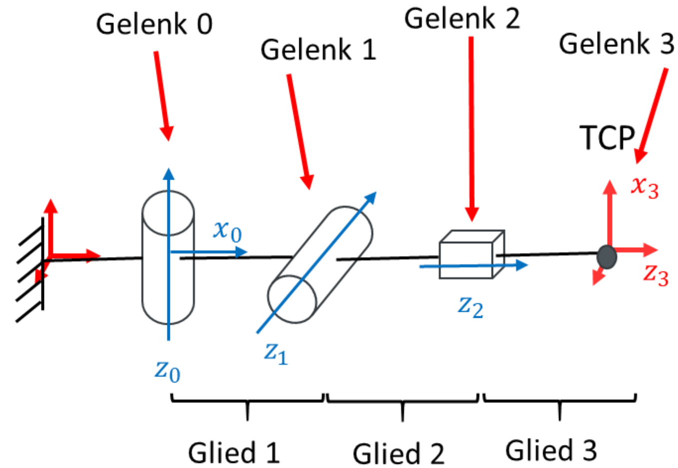
  - Parallel z-axes
  - Extreme case: parallel z-axes lying on top of each other:
    - O_1 Suitably chosen
    - O_{1*} ins neue Koordinatensystem O_1

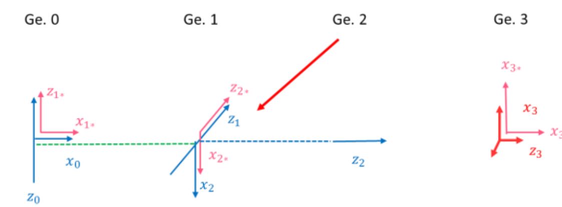


TECHNISCHE HOCHSCHULE LÜBECK

The Denavit-Hardenberg-Transformation

- Summary:
  - Step 1: Sketch with links, joints and z-axes
  - Step 2: Definition of the (help) coordinate systems in another sketch
  - Step 3: Determination of DH parameters, summary in a table
  - Now the transformation matrix can be set up.





$$T_3^0 = T_1^0(\delta_1) * T_2^1(\delta_2) * T_3^2(d_3)$$

Glied	d_i	δ_i	a_i	α_i
1	0	δ_1	l_1	$-\pi/2$
2	0	$+\pi/2 + \delta_2$	0	$+\pi/2$
3	$l_2 + d_3$	0	0	0

TECHNISCHE HOCHSCHULE LÜBECK

The Denavit-Hardenberg-Transformation

Direct kinematics / forward kinematics:

- clearly defined (for open chains)
- to calculate analytically with the help of matrices

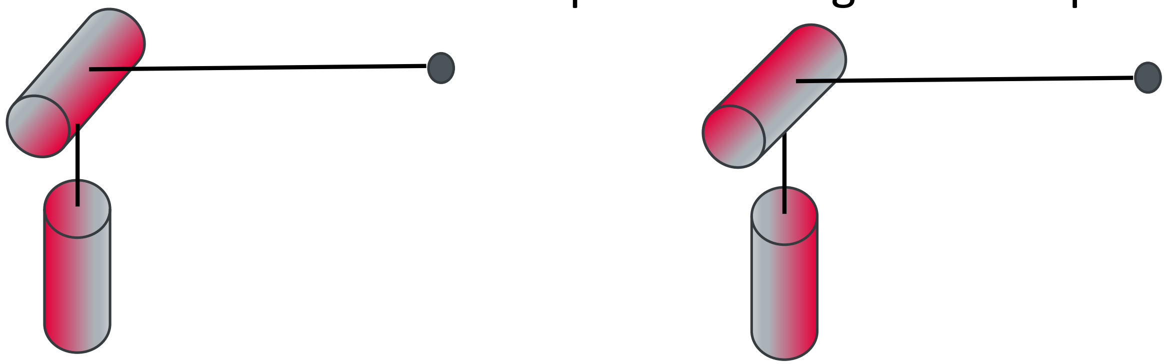
$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}^{Welt} = T_0^{Welt} \cdot T_3^0(\delta_1, \delta_2, d_3) \cdot T_{Wzg}^3 \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}^{TCP}$$

Indirect kinematics / backward kineamtics:

- non-linear system of equations → closed solutions are not always possible
- Multiple solutions may be possible
- There can be an infinite number of solutions (e.g. redundant kinematics)
- There can be no acceptable solutions
- therefore:
    - Introduction of boundary conditions
    - possibly numerical solutions

Inverse Kinematics / Backward Transformation

TECHNISCHE
HOCHSCHULE
LÜBECK

- Initial problem: Inverse kinematics explained using the example:
- 
- Mathematically speaking:
  - We are looking for the "Tool Configuration Vector" w in base coordinates, which is composed of the desired position p and the orientation R of the TCP.
  - The transfer function f depends on the joint variables q of the robot.

$$w(q) = f(q_1, q_2, \dots, q_n) = f(\theta_1, \theta_2, \dots, \theta_n) = \binom{p}{R}$$

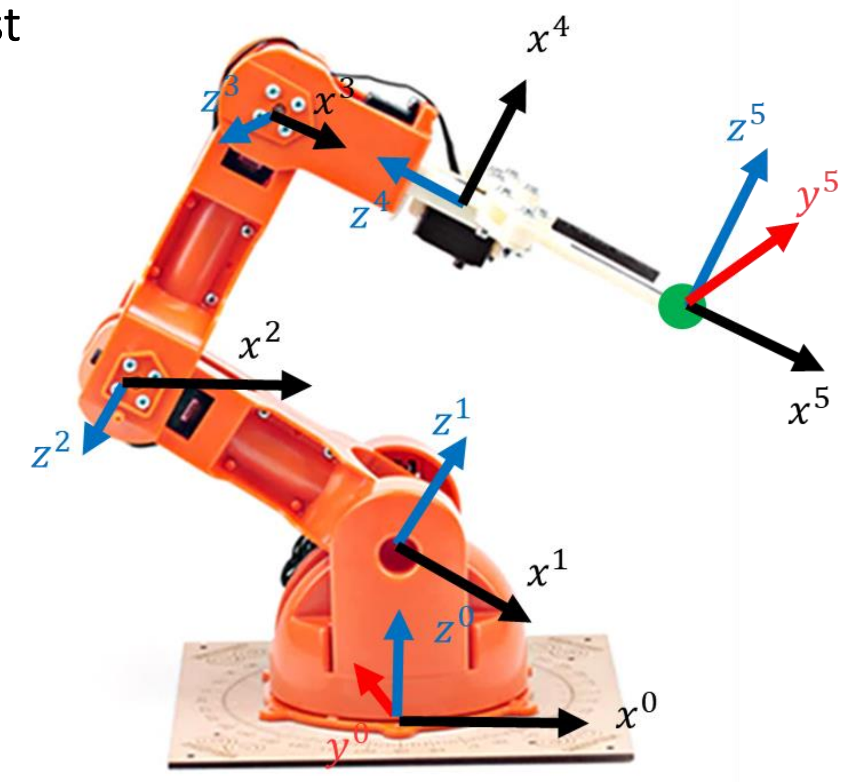
TECHNISCHE HOCHSCHULE LÜBECK

Inverse Kinematics / Backward Transformation

- General approach backward transformation
- If there is a spherical wrist, then the wrist orientation R and the target position p can be calculated back to wrist position:
-

$$p^{wrist} = p - d * r^3$$

  - d is the length of the tool
  - r^3 is the approach vector
  -



Bildquelle: https://store.arduino.cc/products/tinkerkit-braccio-robot

Inverse Kinematics / Backward Transformation

- Numerical approach backward transformation
  - Determination of the zeros of the expression:

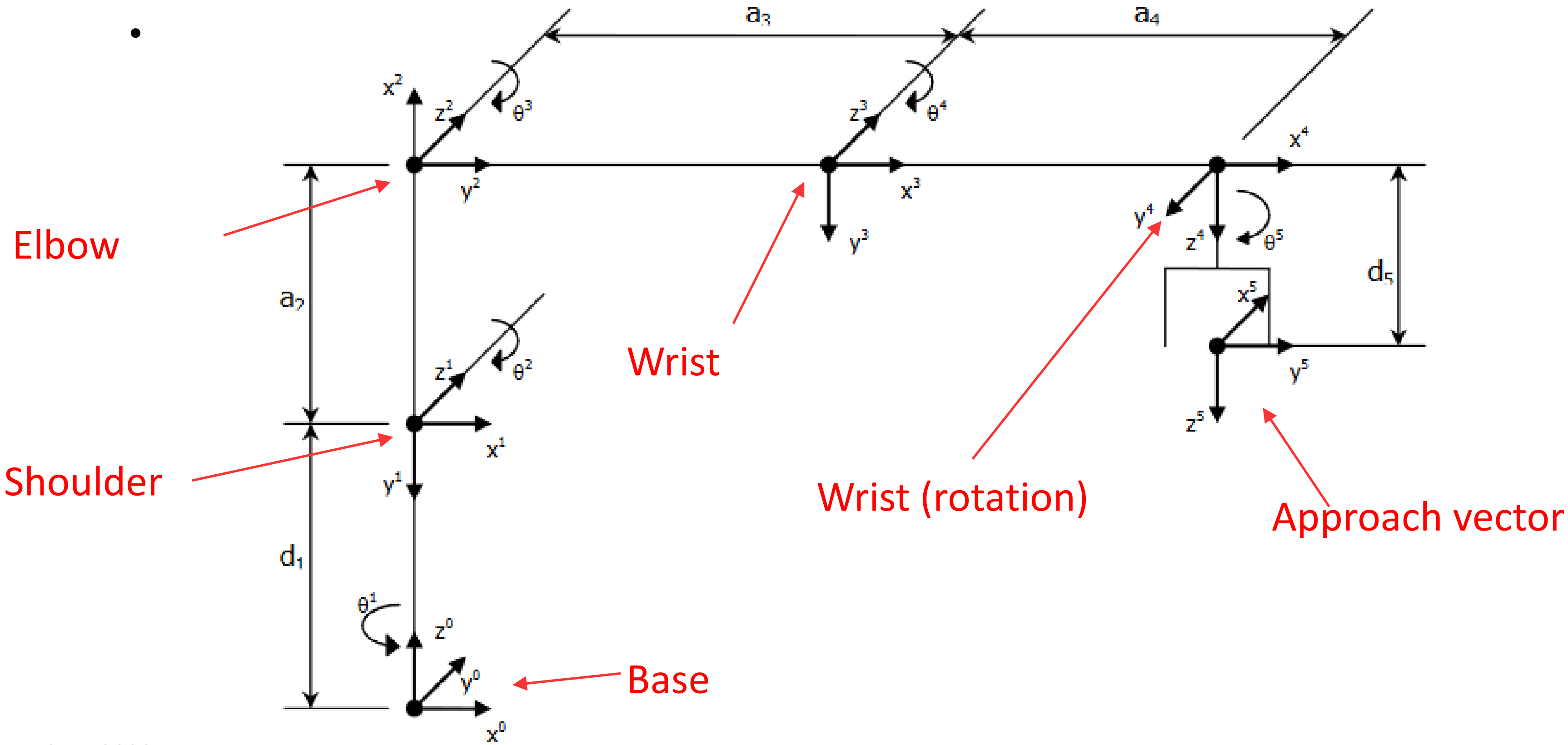
$$f(\theta_1, \theta_2, \dots, \theta_n) - \binom{p}{R} = 0$$

  - Approach, e.g. by Newton method
  -
  - Contains the partial derivatives of the TCP coordinates according to the individual joint angles (Jakobi matrix)
  -

TECHNISCHE HOCHSCHULE LÜBECK

Inverse Kinematics / Backward Transformation

- Reverse transformation 5-axis articulated arm (Schilling – Fundamentals of Robotics)
  - Kinematic model
  -



Bildquelle: Schilling, R.J. „Fundamentals of Robotics“, 2003

TECHNISCHE HOCHSCHULE LÜBECK

Inverse Kinematics / Backward Transformation

- Reverse transformation 5-axis articulated arm
- "Tool configuration vector" w:

$$w(q) = \begin{pmatrix} \boxed{\begin{matrix} x \\ y \\ z \end{matrix}} \\ \boxed{\begin{matrix} Rot.\ around\ x_0 \\ Rot.\ around\ y_0 \\ Rot.\ around\ z_0 \end{matrix}} \end{pmatrix}$$

Position of the gripper (here origin of KS 5) related to KS 0

Approach vector

- Vector of joint variable q:

$$q = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \end{pmatrix} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \end{pmatrix}$$

Inverse Kinematics / Backward Transformation

TECHNISCHE HOCHSCHULE LÜBECK

- Reverse transformation 5-axis articulated arm (Schilling – Fundamentals of Robotics)
  - Determine the bases‘ rotation angle

$$q_1 = atan2(w_2, w_1)$$

  - atan2(y,x) function is the 4-quadrant variant for the arcus-tangent function (available in most programming languages, based on the sequence of x,y)

x-value	Quadrant	atan2(y, x)
x > 0	1, 4	atan(y/x)
x = 0	1, 4	[sgn(y)] * $\pi/2$
x < 0	2, 3	atan(y/x) + [sgn(y)] * π

TECHNISCHE HOCHSCHULE LÜBECK

Inverse Kinematics / Backward Transformation

- Reverse transformation 5-axis articulated arm (Schilling – Fundamentals of Robotics)
  - Calculation of elbow joint angle q_3
    - The total tool pitch angle corresponds to the given angle w_4

$$q_{234} = q_2 + q_3 + q_4 = w_4$$

$$q_{234} = atan2(-w_4 \cos q_1 - w_5 \sin q_1, -w_6)$$

    - 2 intermediate variables are introduced:

$$b_1 = w_1 \cos q_1 + w_2 \sin q_1 - a_4 \cos q_{234} + d_5 \sin q_{234}$$

$$b_2 = d_1 - a_4 \sin q_{234} - d_5 \cos q_{234} - w_3$$

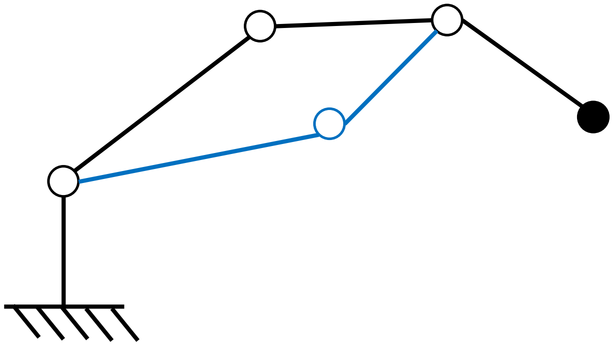
    - From this, b^2 is calculated: $b^2 = b_1^2 + b_2^2$

Inverse Kinematics / Backward Transformation

- Reverse transformation 5-axis articulated arm (nach Schilling – Fundamentals of Robotics)
  - Calculation of elbow joint angle q_3
    - and from this the joint angle q_3:

$$q_3 = \pm \text{acos} \frac{b^2 - a_2^2 - a_3^2}{2a_2 \, a_3}$$

    - +/- takes into account the „ellbow up“- and the „ellbow down“-variant, respectively



TECHNISCHE HOCHSCHULE LÜBECK

Inverse Kinematics / Backward Transformation

- Reverse transformation 5-axis articulated arm (Schilling – Fundamentals of Robotics)
  - Calculation of the shoulder-joint angle q_2

$$q_2 = \operatorname{atan}\big((a_2 + a_3 \cos q_3)b_2 - a_3 b_1 \sin q_3\,, (a_2 + a_3 \cos q_3)b_1 + a_3 b_2 \sin q_3\big)$$

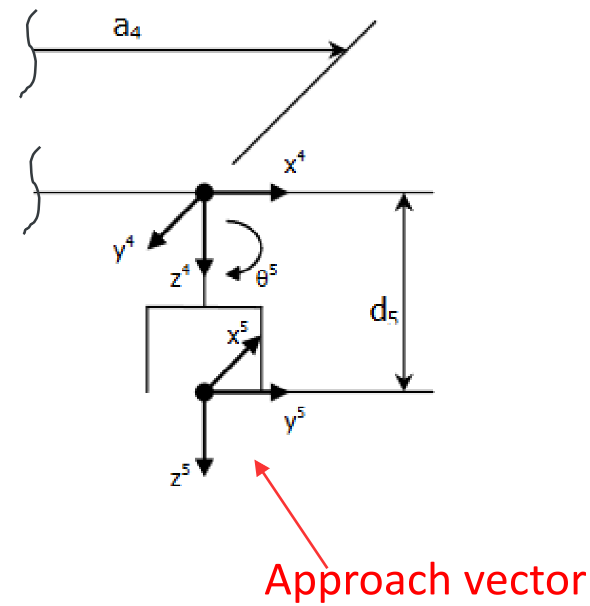
  - Tool pitch angle:
  -
  - Tool roll angle q_5:

$$q_4 = q_{234} - q_2 - q_3$$

$$q_5 = \pi \ln(\sqrt{w_4^2 + w_5^2 + w_6^2})$$

Inverse Kinematics / Backward Transformation

TECHNISCHE HOCHSCHULE LÜBECK



- for the 5-axis articulated robot, the approach vector has to be givenin the form:

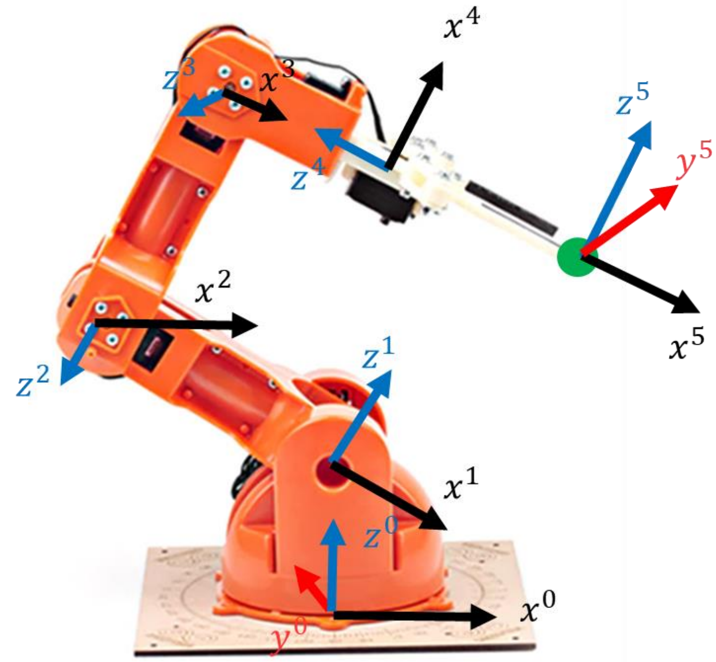
$$w = \binom{w_p}{w_o} \begin{pmatrix} p \\ \exp\left(\frac{q_n}{\pi}\right) r_3 \end{pmatrix}$$

- where w_p is the position vector and w_o describes the orientation.
- the function $\mathrm{f}(q_n) = \exp\left(\frac{q_n}{\pi}\right) r_3$:
  - describes the roll angle based on the length of w_o
  - r_3 is the unit vector
  - is always positive
  - its inverse function is well defined: $f^{-1}(w) = \pi \ln(\sqrt{w_4^2 + w_5^2 + w_6^2})$

Inverse Kinematics / Backward Transformation

TECHNISCHE HOCHSCHULE LÜBECK

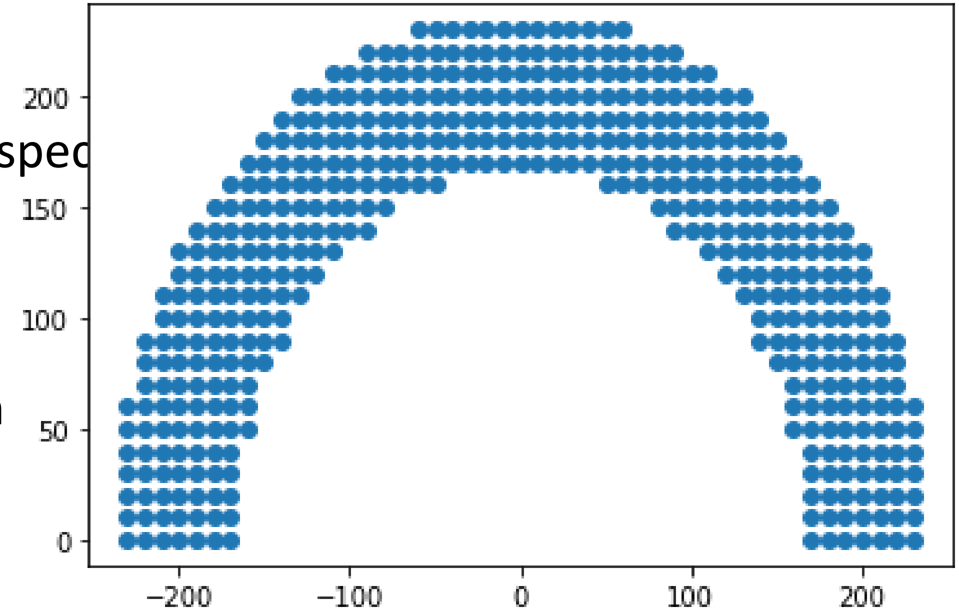
- Reverse transformation 5-axis articulated arm (Schilling – Fundamentals of Robotics)
  - Model considers general configuration as shown
  - To adapt the model to the Tinkerbot Braccio
    - Comparison of DH parameters (sign)
    - Adaptation of coordinate systems:
    - Consider different direction of rotation
    - Consider offset, especially with the tool pitch angle!



Bildquelle: https://store.arduino.cc/products/tinkerkit-braccio-robot

Inverse Kinematics / Backward Transformation

- Working area of the Tinkerbot Braccio (5-axis articulated arm)
- Boundary conditions:
  - Orientation of the gripper:
  - tilted by 11,25° (=pi/16) with respect to the z-axis of the base KS
  - Rotation of the hand
    Joint axis of rotation always 90°
  - Position of the gripper tip 3 mm above X-Y plane

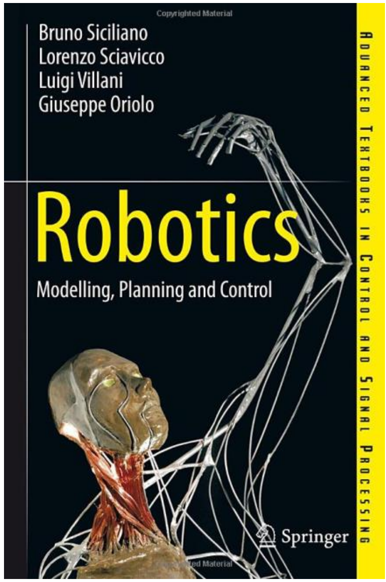


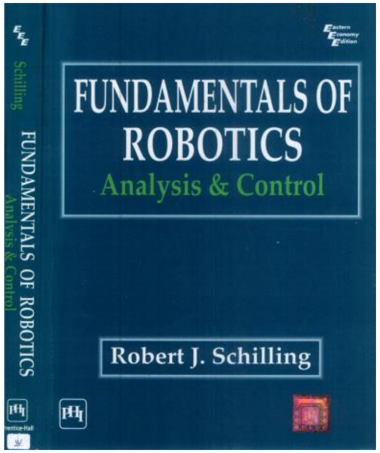
- Result:
  - 5-axis articulated arm has a very limited working area
  - Gripper can hardly be aligned parallel to the z-axis of the base KS

Inverse Kinematics / Backward Transformation

TECHNISCHE HOCHSCHULE LÜBECK

- Analytical solutions exist e.g. for the following kinematics:
    - 5-axis articulated arm
    - Scara
    - 6-axis articulated arm
- The following references:
    - B.Siciliano, L. Sciavicco, L. Villani, G. Oriolo "Robotics: Modelling, Planning and Control , Springer 2009
      Attention: different approach to DH transformation!
    - Schilling, R.J. "Fundamentals of Robotics", Prentice Hall 2003





Summary

- DH method forms the basis for calculating TCP position based on the joint variables
- Inverse kinematics is necessary in practical application to determine the joint variables for a TCP pose.
-