Introduction to Robotics

Lecture 2

Shuang Li, Jianwei Zhang
[sli, zhang]@informatik.uni-hamburg.de

University of Hamburg
Faculty of Mathematics, Informatics and Natural Sciences
Department of Informatics
Technical Aspects of Multimodal Systems

May 01, 2020

Outline

Introduction

Spatial Description and Transformations

Forward Kinematics

- More on presentation of a rigid body
- Denavit-Hartenberg convention
- Definition of joint coordinate systems
- Example DH-Parameter of a single joint
- Example DH-Parameter for a manipulator
- Example featuring Mitsubishi PA10-7C

Robot Description

Inverse Kinematics for Manipulators

Differential motion with homogeneous transformations

Jacobian

Trajectory planning

Trajectory generation

Outline (cont.)

Dynamics

Robot Control

Task-Level Programming and Trajectory Generation

Task-level Programming and Path Planning

Task-level Programming and Path Planning

Architectures of Sensor-based Intelligent Systems

Summary

Conclusion and Outlook

Review of last lecture

- Degree of freedom
  - The number of variables to determine position of a control system in space.
- Robot classification
  - mechanical structure
- Rotation matrix
  - ${}^{A}R_{B}^{-1} = {}^{B}R_{A} = {}^{B}R_{A}^{T} \quad \text{and} \quad {}^{A}R_{B}{}^{B}R_{A} = I$
- Homogeneous transformation matrix
  - $T = \begin{bmatrix} R & \vec{p} \\ 0 & 1 \end{bmatrix}$
  - Transformation equation

Transformation equation

In order to find the desired end effector pose:

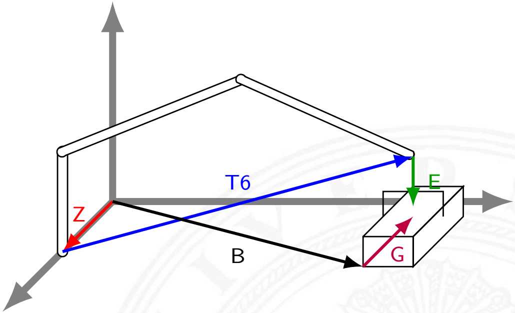
$$Z T_6 E = B G$$

In order to find the manipulator transformation T_6:

$$T_6 = Z^{-1} B G E^{-1}$$

In order to determine the pose of the object B:

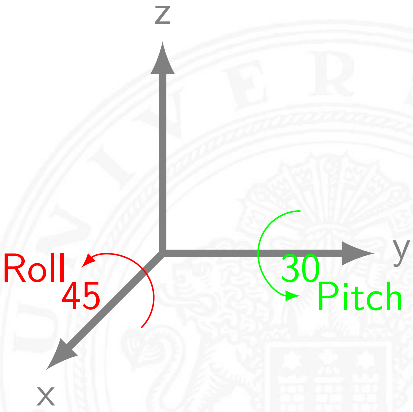
$$B = Z T_6 E G^{-1}$$



Review of last lecture

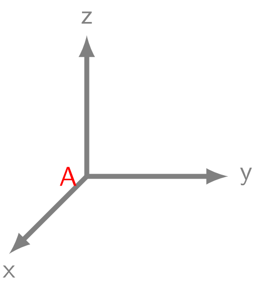
A vector ${}^{A}\vec{P}$ is rotated about $\hat{Y}$ by 30 degrees and is subsequently rotated about $\hat{X}$ by 45 degrees. Give the rotation matrix that accomplishes these rotations in the given order.

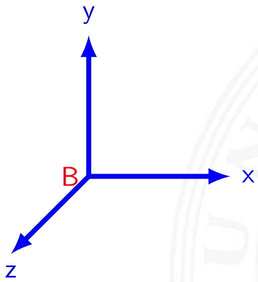
$$
\begin{aligned}
R &= R_{x,45}R_{y,30} \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos45 & -sin45 \\ 0 & sin45 & cos45 \end{bmatrix} \begin{bmatrix} cos30 & 0 & sin30 \\ 0 & 1 & 0 \\ -sin30 & 0 & cos30 \end{bmatrix} \\
&= \begin{bmatrix} 0.866 & 0 & 0.5 \\ 0.353 & 0.707 & -0.612 \\ -0.353 & 0.707 & 0.612 \end{bmatrix}
\end{aligned}
$$



More on presentation of orientation: Euler angles

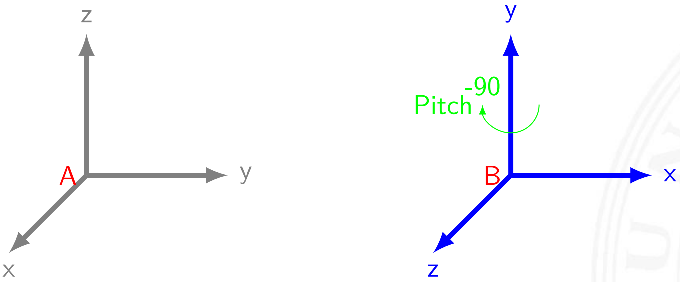
- Euler angles φ, θ, ψ





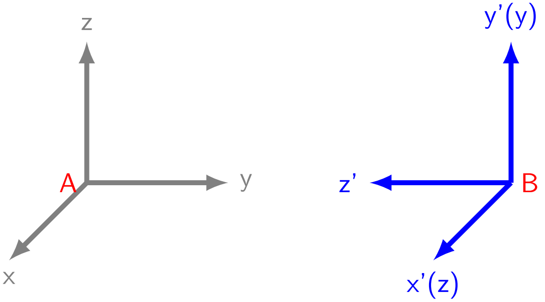
More on presentation of orientation: Euler angles (cont.)

- Euler-angles φ, θ, ψ



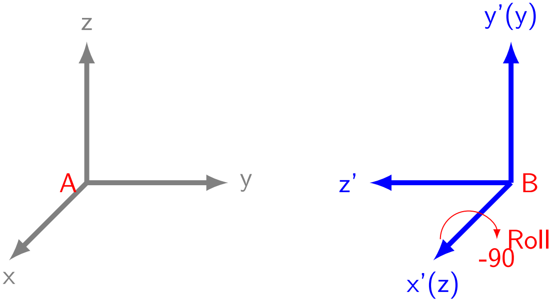
More on presentation of orientation: Euler angles (cont.)

- Euler-angles φ, θ, ψ



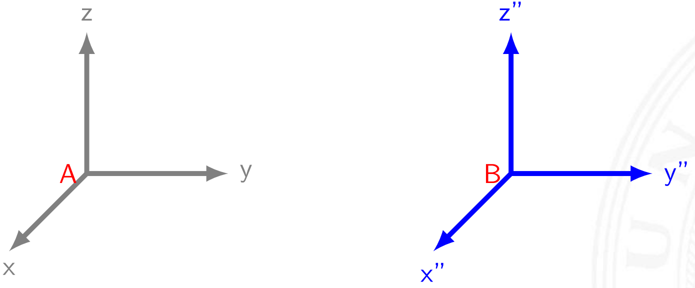
More on presentation of orientation: Euler angles (cont.)

- Euler-angles φ, θ, ψ



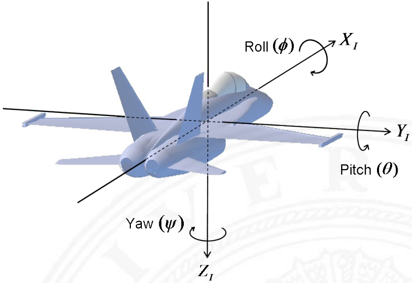
More on presentation of orientation: Euler angles (cont.)

- Euler-angles φ, θ, ψ



More on presentation of orientation: Euler angles (cont.)

- Euler-angles φ, θ, ψ
  - rotations are performed **successively** around the axes, e. g. *ZYX* or *ZXZ* (12 possibilities!)
  - order depends on reference coordinates
  - Intrinsic rotations
  - Extrinsic (fix angle) rotations
- Roll-Pitch-Yaw
  - X-Y-Z fixed angles
  - used in aviation and maritime



Converting Euler Angles to a Rotation Matrix

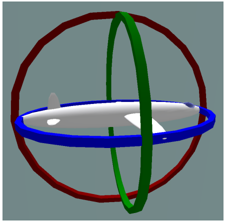
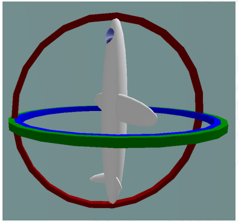
$$R_{x,\varphi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\varphi & -S\varphi \\ 0 & S\varphi & C\varphi \end{bmatrix}$$

$$R_{y,\theta} = \begin{bmatrix} C\theta & 0 & S\theta \\ 0 & 1 & 0 \\ -S\theta & 0 & C\theta \end{bmatrix}$$

$$R_{z,\psi} = \begin{bmatrix} C\psi & -S\psi & 0 \\ S\psi & C\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

More on presentation of orientation

- Rotation matrix
  - implicit, easy to use linear algebra to perform computation
- Euler angles
  - Gimbal lock!
  - When two gimbals rotate around the same axis, the system loses one degree of freedom.

17

More on presentation of orientation (cont.)

- Rotation matrix
  - implicit, easy to use linear algebra to perform computation, singularity-free
- Euler angles φ, θ, ψ
  - explicit, but gimbal lock/singularity happens
- Equivalent angle-axis representation $R_{k,\theta}$
  - the angle for a rotation about an axis vector
- Quaternion $[x, y, z, w]$
  - 4D vectors that represent 3D rigid body orientations
  - Unit quaternion: $x^2 + y^2 + z^2 + w^2 = 1$

Tools

python: Numpy, pyquaternion
c++: Eigen

[17] https://en.wikipedia.org/wiki/Gimbal_lock

Manipulator

- A manipulator is considered as set of **links** connected by **joints**
  - serial robots (vs.parallel robots)
- Types of joints
  - revolute joints
  - prismatic joints

Forward kinematics

- Movement depiction of the mechanical systems as fixed body chains
- Translate a series of joint parameters $\Longrightarrow$ cartesian pose of the end effector

Purpose

Absolute determination of the position of the end effector (TCP) in the cartesian coordinate system

Tool Center Point (TCP) description

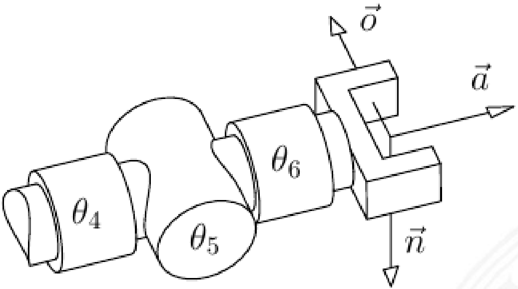
Using a vector $\vec{p}$, the TCP position is depicted.

Three unit vectors:

- $\vec{a}$: (approach vector),
- $\vec{o}$: (orientation vector),
- $\vec{n}$: (normal vector)

specify the orientation of the TCP.

Tool Center Point (TCP) description (cont.)



Thus, the transformation T consists of the following elements:

$$T = \begin{bmatrix} \vec{n} & \vec{o} & \vec{a} & \vec{p} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Transformation regulation, which describes the relation between joint coordinates of a robot $\mathbf{q}$ and the environment coordinates of the end effector $\mathbf{x}$
- Solely determined by the geometry of the robot
  - Base frame
  - Relation of frames to one another

    $\Longrightarrow$ Formation of a recursive chain
  - Joint coordinates:

$$q_i = \begin{cases} \theta_i & : \text{rotational joint} \\ d_i & : \text{translation joint} \end{cases}$$

Kinematic equations

- In each link, a coordinate frame is attached
- A homogeneous matrix ${}^{i-1}T_i$ depicts the relative translation and rotation between two consecutive joints
  - joint transition
- For a manipulator consisting of six joints:
  - ${}^{0}T_1:$ depicts position and orientation of the first link with respect to the base

$$\vdots$$

  - ${}^{5}T_6:$ depicts position and orientation of the 6th link in regard to link 5

The resulting product is defined as:

$$T_6 = {}^{0}T_1 \, {}^{1}T_2 \, {}^{2}T_3 \, {}^{3}T_4 \, {}^{4}T_5 \, {}^{5}T_6$$

Kinematic description

- Calculation of $T_6 = \prod_{i=1}^{n} T_i$, T_i short for ${}^{i-1}T_i$
  - T_6 defines, how n joint transitions describe 6 cartesian DOF
- Definition of one coordinate system (CS) per segment i
  - generally arbitrary definition
- Determination of one transformation T_i per segment $i = 1..n$
  - generally 6 parameters (3 rotational + 3 translational) required
  - different sets of parameters and transformation orders possible

Solution

Denavit-Hartenberg (DH) convention

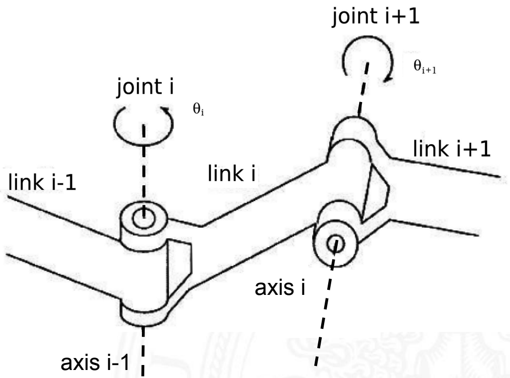
Denavit-Hartenberg convention

- first published by Denavit and Hartenberg in 1955
- established principle
- determination of a transformation matrix T_i using **four** parameters
  - link length, link twist, link offset and joint angle (a_i, α_i, d_i, θ_i)

Parameters for description of two arbitrary links

Two parameters for the description of the link structure i

- link length a_i
- link twist α_i

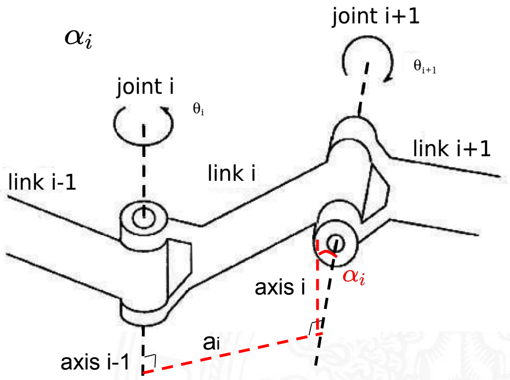


Parameters for description of two arbitrary links

Two parameters for the description of the link structure i

- link length a_i: shortest distance between the axis $i-1$ and the axis i
- link twist α_i: rotation angle from axis $i-1$ to axis i in the right-hand sense about a_i

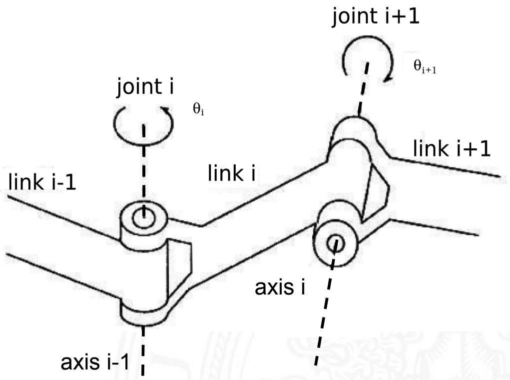
a_i and α_i are constant values due to construction



Parameters for describing two arbitrary links (cont.)

Two for relative distance and angle of adjacent links

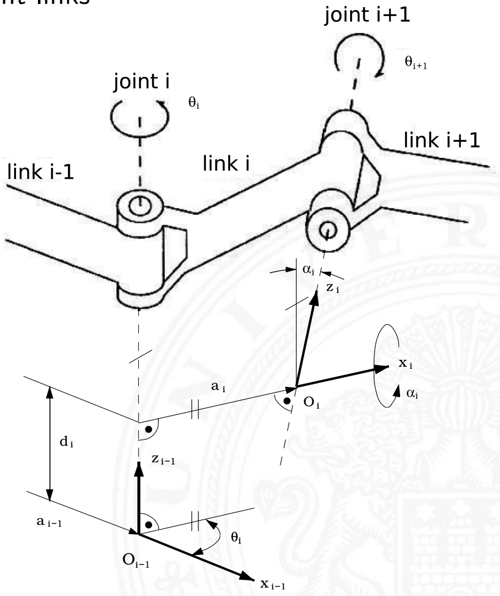
- link offset d_i
- joint angle θ_i



Two for relative distance and angle of adjacent links

- link offset d_i: the distance along the common axis $i-1$ from link $i-1$ to the link i
- joint angle θ_i: the amount of rotation about the common axis $i-1$ between the link $i-1$ and the link i

θ_i and d_i are variable
- rotational: θ_i variable, d_i fixed
- translational: d_i variable, θ_i fixed

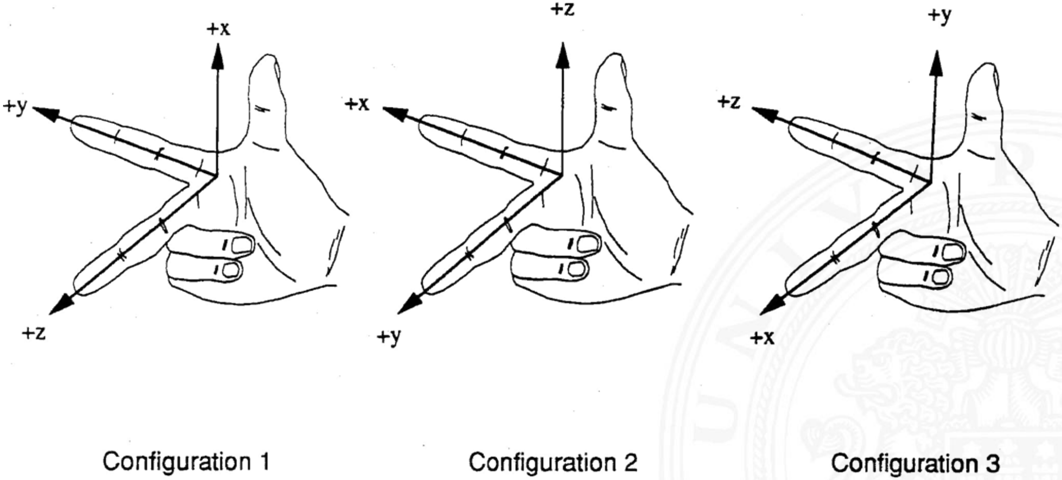


DH Parameters summary

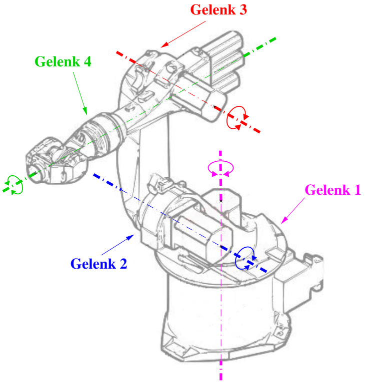
Four DH parameters:
link length, link twist, link offset and joint angle
$(a_i, \alpha_i, d_i, \theta_i)$

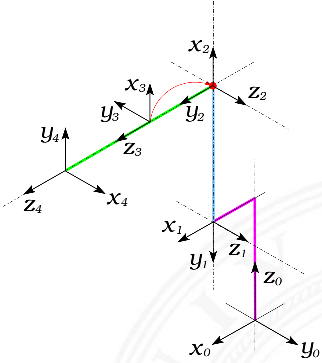
- 3 fixed link parameters
- one joint variable
  - revolute: θ_i variable
  - prismatic: d_i variable
- a_i, α_i: describe the link i
- d_i, θ_i: describe the link's connection

Right-Handed Coordinate System



Definition of joint coordinate systems (classic)

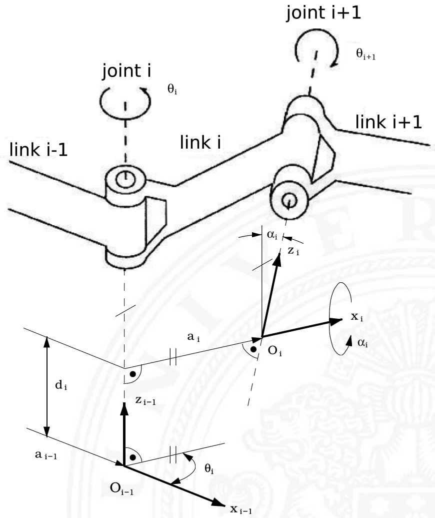




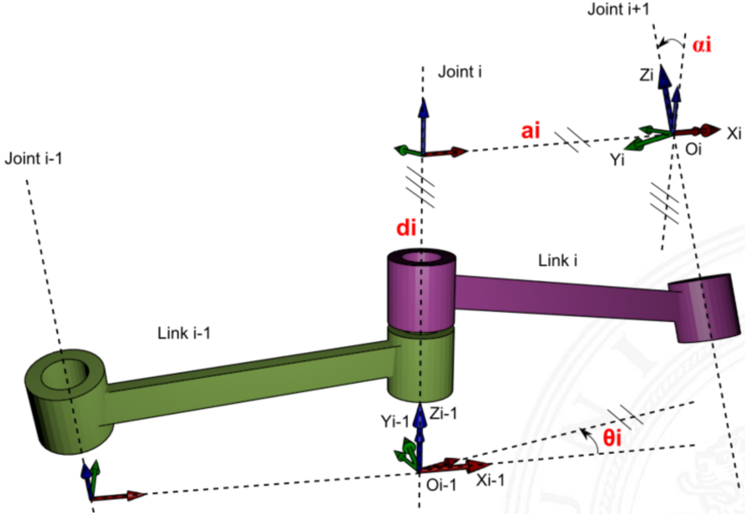
- axis z_{i-1} is set along the axis of motion of the i^{th} joint
- axis x_i is parallel to the common normal of z_{i-1} and z_i ($x_i \parallel (z_{i-1} \times z_i)$).
- axis y_i concludes a right-handed coordinate system
- CS_0 is the stationary origin at the base of the manipulator

DH Parameters

- link length a_i: distance from z_{i-1}-axis to z_i-axis measured along x_i-axis
- link twist α_i: angle from z_{i-1}-axis to z_i-axis measured around x_i-axis
- link offset d_i: distance from x_{i-1} to x_i measured along z_{i-1}-axis
- joint angle θ_i: joint angle from x_{i-1} to x_i measured around z_{i-1}-axis



Classic Parameters



Transformation order

$$T_i = R_{z_{i-1}}(\theta_i) \cdot T_{z_{i-1}}(d_i) \cdot T_{x_i}(a_i) \cdot R_{x_i}(\alpha_i) \rightarrow CS_i$$

Frame transformation for two links (classic)

Creation of the relation between frame i and frame $(i-1)$ through the following rotations and translations:

- Rotate around z_{i-1} by angle θ_i
- Translate along z_{i-1} by d_i
- Translate along x_i by a_i
- Rotate around x_i by angle α_i

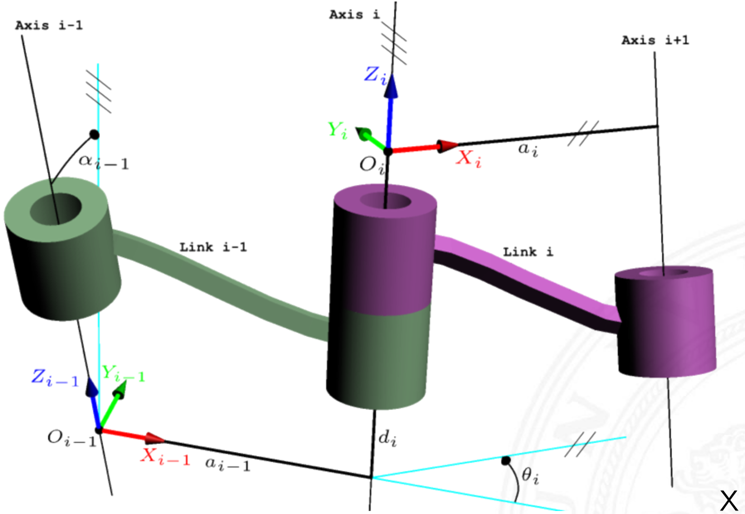
Using the product of four homogeneous transformations, which transform the coordinate frame $i-1$ into the coordinate frame i, the matrix A_i can be calculated as follows:

$$T_i = R_{z_{i-1}}(\theta_i) \cdot T_{z_{i-1}}(d_i) \cdot T_{x_i}(a_i) \cdot R_{x_i}(\alpha_i) \rightarrow CS_i$$

Frame transformation for two links (classic) (cont.)

$$T_i = \begin{bmatrix} C\theta_i & -S\theta_i & 0 & 0 \\ S\theta_i & C\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dots & 0 \\ \dots & 0 \\ \dots & d_i \\ \dots & 1 \end{bmatrix} \begin{bmatrix} \dots & a_i \\ \dots & 0 \\ \dots & 0 \\ \dots & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha_i & -S\alpha_i & 0 \\ 0 & S\alpha_i & C\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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



Transformation order

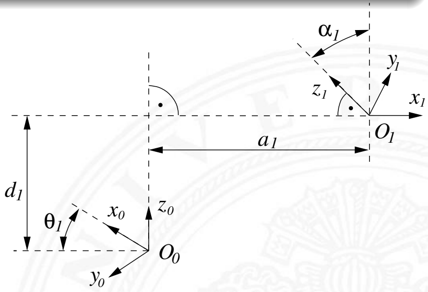
$$T_i = R_{x_{i-1}}(\alpha_{i-1}) \cdot T_{x_{i-1}}(a_{i-1}) \cdot R_{z_i}(\theta_i) \cdot T_{z_i}(d_i) \rightarrow CS_i$$

Definition of joint coordinate systems: Exceptions

Beware

The Denavit-Hartenberg convention is ambiguous!

- z_{i-1} is parallel to z_i
  - arbitrary shortest normal
  - usually $d_i = 0$ is chosen
- z_{i-1} intersects z_i
  - usually $a_i = 0$ such that CS lies in the intersection point
- orientation of CS_n ambigous, as no joint $n + 1$ exists
  - x_n must be a normal to z_{n-1}
  - usually z_n is chosen to point in the direction of the approach vector $\vec{a}$ of the tcp

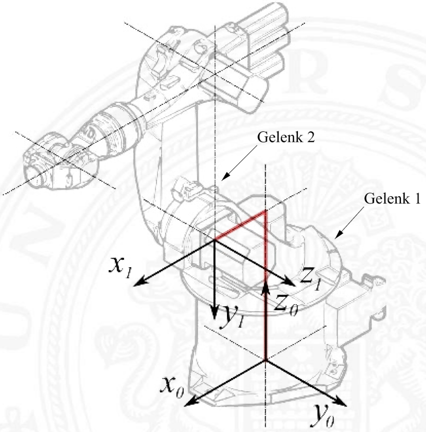


Example DH-Parameter of a single joint

Determination of DH-Parameter (θ, d, a, α) for calculation of joint transformation:
$T_1 = R_z(\theta_1) T_z(d_1) T_x(a_1) R_x(\alpha_1)$
joint angle rotate by θ_1 around z_0, such that x_0 is parallel to x_1

$$R_z(\theta_1) = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

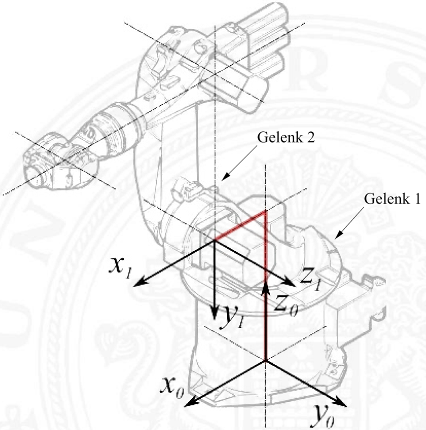
for the shown joint configuration $\theta_1 = 0^\circ$



Example DH-Parameter of a single joint (cont.)

link offset translate by d_1 along z_0 until the intersection of z_0 and x_1

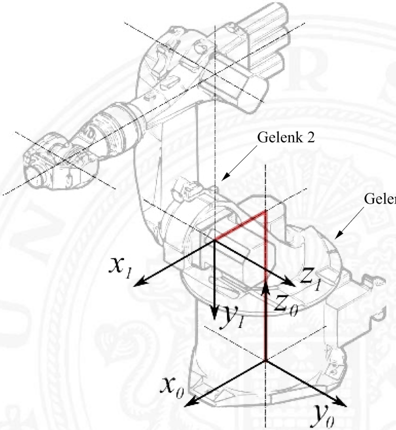
$$T_z(d_1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Example DH-Parameter of a single joint (cont.)

link length translate by a_1 along x_1 such that the origins of both CS are congruent

$$T_x(a_1) = \begin{bmatrix} 1 & 0 & 0 & a_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

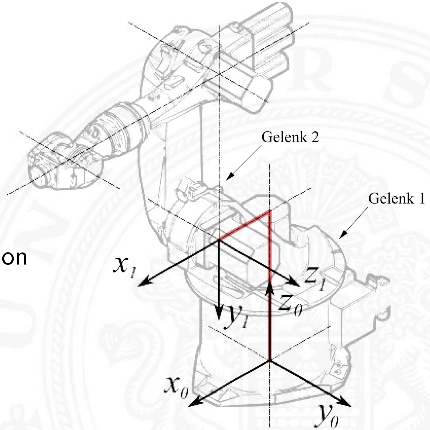


Example DH-Parameter of a single joint (cont.)

link twist rotate z_0 by α_1 around x_1, such that z_0 lines up with z_1

$$R_{x(\alpha_1)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_1) & -\sin(\alpha_1) & 0 \\ 0 & \sin(\alpha_1) & \cos(\alpha_1) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

for the shown joint configuration, $\alpha_1 = -90°$ due to construction



Example DH-Parameter of a single joint (cont.)

- total transformation of CS_0 to CS_1 (general case)

$$
\begin{aligned}
{}^0T_1 &= R_z(\theta_1) \cdot T_z(d_1) \cdot T_x(a_1) \cdot R_x(\alpha_1) \\
&= \begin{bmatrix} cos\theta_1 & -sin\theta_1 cos\alpha_1 & sin\theta_1 sin\alpha_1 & a_1 cos\theta_1 \\ sin\theta_1 & cos\theta_1 cos\alpha_1 & -cos\theta_1 sin\alpha_1 & a_1 sin\theta_1 \\ 0 & sin\alpha_1 & cos\alpha_1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

- rotary case: variable θ_1 and fixed d_1, a_1 *und* $(\alpha_1 = -90^\circ)$

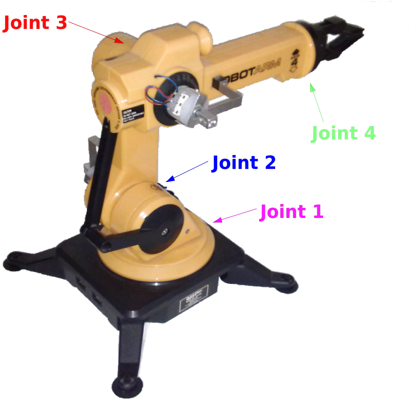
$$
\begin{aligned}
{}^0T_1 &= R_z(\theta_1) \cdot T_z(d_1) \cdot T_x(a_1) \cdot R_x(-90^\circ) \\
&= \begin{bmatrix} cos\theta_1 & 0 & -sin\theta_1 & a_1 cos\theta_1 \\ sin\theta_1 & 0 & cos\theta_1 & a_1 sin\theta_1 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

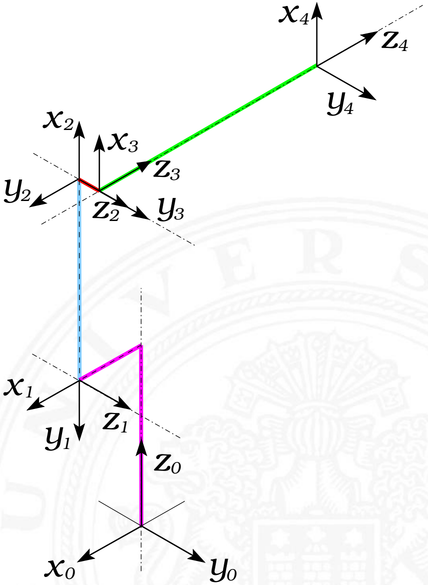
Procedure for predefined structure

- Fixed origin: CS_0 is the fixed frame at the base of the manipulator
- Determination of axes and consecutive numbering from 1 to n
- Positioning O_i on rotation- or shear-axis i,
  z_i points aways from z_{i-1}
- Determination of normal between the axes; setting x_i (in direction to the normal)
- Determination of y_i (right-hand system)
- Read off Denavit-Hartenberg parameters
- Calculation of overall transformation

Example DH-Parameter for Quickshot

- Definition of CS corresponding to DH convention
- Determination of DH-Parameter





Example Transformation matrix T_6

$$T_6 = T_1 \cdot T_2 \cdot T_3 \cdot T_4$$

$$= \begin{bmatrix} \cos\theta_1 & 0 & -\sin\theta_1 & 20\cos\theta_1 \\ \sin\theta_1 & 0 & \cos\theta_1 & 20\sin\theta_1 \\ 0 & -1 & 0 & 100 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & 160\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & 160\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot$$

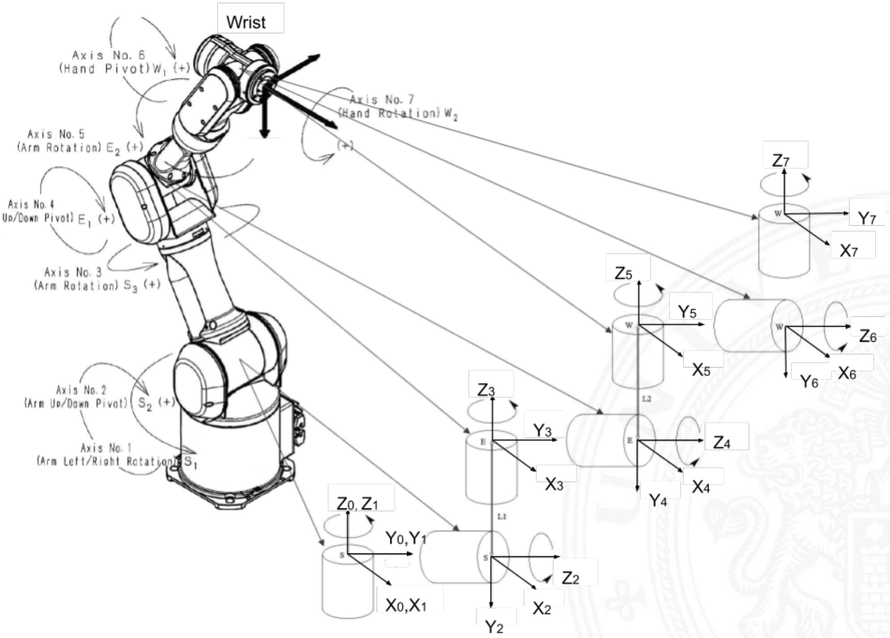
$$\begin{bmatrix} \cos\theta_3 & 0 & \sin\theta_3 & 0 \\ \sin\theta_3 & 0 & -\cos\theta_3 & 0 \\ 0 & 1 & 0 & 28 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_4 & -\sin\theta_4 & 0 & 0 \\ \sin\theta_4 & \cos\theta_4 & 0 & 0 \\ 0 & 0 & 1 & 250 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta_1 \cos\theta_4(\cos\theta_2 \cos\theta_3 - \sin\theta_2 \sin\theta_3) - \sin\theta_1 \sin\theta_4 & \dots & \dots & \dots \\ \sin\theta_1 \cos\theta_4(\sin\theta_2 \cos\theta_3 + \cos\theta_2 \sin\theta_3) + cos\theta_1 \sin\theta_4 & \dots & \dots & \dots \\ -\cos\theta_4(\sin\theta_2 \cos\theta_3 + \cos\theta_2 \sin\theta_3) & \dots & \dots & \dots \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Sum-of-Angle formula

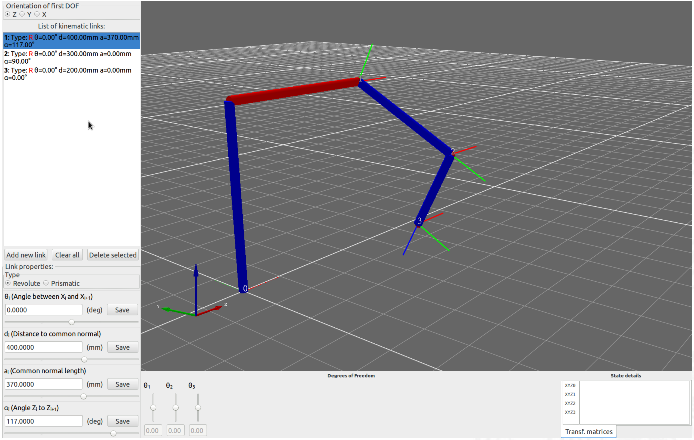
$$C_{23} = C_2 C_3 - S_2 S_3,$$
$$S_{23} = C_2 S_3 + S_2 C_3$$

Mitsubishi PA10-7C



Robotic arm kinematic GUI from MRPT

Download link



[18]

[18] Mobile Robot Programming Toolkit, https://www.mrpt.org/MRPT_in_GNU/Linux_repositories

Programming implementation

Write your own FK function!

- Robotics toolbox in Matlab
  - the implementation of book "Robotics, Vision & Control" by Peter Corke
- PythonRobotics
  - Python code collection of robotics algorithms, especially for autonomous navigation
- Robotics library
  - C++ framework for robot kinematics, dynamics, motion planning, control
- pybotics
  - provides a simple and clear interface to simulate and evaluate common robot concepts

Bibliography

[1] G.-Z. Yang, R. J. Full, N. Jacobstein, P. Fischer, J. Bellingham, H. Choset, H. Christensen, P. Dario, B. J. Nelson, and R. Taylor, "Ten robotics technologies of the year," 2019.

[2] J. K. Yim, E. K. Wang, and R. S. Fearing, "Drift-free roll and pitch estimation for high-acceleration hopping," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8986–8992, IEEE, 2019.

[3] J. F. Engelberger, *Robotics in service*.
MIT Press, 1989.

[4] K. Fu, R. González, and C. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*.
McGraw-Hill series in CAD/CAM robotics and computer vision, McGraw-Hill, 1987.

[5] R. Paul, *Robot Manipulators: Mathematics, Programming, and Control: the Computer Control of Robot Manipulators*.
Artificial Intelligence Series, MIT Press, 1981.

[6] J. Craig, *Introduction to Robotics: Pearson New International Edition: Mechanics and Control*.
Always learning, Pearson Education, Limited, 2013.

Bibliography (cont.)

[7] W. Böhm, G. Farin, and J. Kahmann, “A Survey of Curve and Surface Methods in CAGD,” *Comput. Aided Geom. Des.*, vol. 1, pp. 1–60, July 1984.

[8] J. Zhang and A. Knoll, “Constructing Fuzzy Controllers with B-spline Models - Principles and Applications,” *International Journal of Intelligent Systems*, vol. 13, no. 2-3, pp. 257–285, 1998.

[9] M. Eck and H. Hoppe, “Automatic Reconstruction of B-spline Surfaces of Arbitrary Topological Type,” in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’96, (New York, NY, USA), pp. 325–334, ACM, 1996.

[10] M. C. Ferch, *Lernen von Montagestrategien in einer verteilten Multiroboterumgebung*. PhD thesis, Bielefeld University, 2001.

[11] J. H. Reif, “Complexity of the Mover’s Problem and Generalizations - Extended Abstract,” *Proceedings of the 20th Annual IEEE Conference on Foundations of Computer Science*, pp. 421–427, 1979.

Bibliography (cont.)

[12] J. T. Schwartz and M. Sharir, "A Survey of Motion Planning and Related Geometric Algorithms," *Artificial Intelligence*, vol. 37, no. 1, pp. 157–169, 1988.

[13] J. Canny, *The Complexity of Robot Motion Planning*.
MIT press, 1988.

[14] T. Lozano-Pérez, J. L. Jones, P. A. O'Donnell, and E. Mazer, *Handey: A Robot Task Planner*.
Cambridge, MA, USA: MIT Press, 1992.

[15] O. Khatib, "The Potential Field Approach and Operational Space Formulation in Robot Control," in *Adaptive and Learning Systems*, pp. 367–377, Springer, 1986.

[16] J. Barraquand, L. Kavraki, R. Motwani, J.-C. Latombe, T.-Y. Li, and P. Raghavan, "A Random Sampling Scheme for Path Planning," in *Robotics Research* (G. Giralt and G. Hirzinger, eds.), pp. 249–264, Springer London, 1996.

[17] R. Geraerts and M. H. Overmars, "A Comparative Study of Probabilistic Roadmap Planners," in *Algorithmic Foundations of Robotics V*, pp. 43–57, Springer, 2004.

Bibliography (cont.)

[18] K. Nishiwaki, J. Kuffner, S. Kagami, M. Inaba, and H. Inoue, "The Experimental Humanoid Robot H7: A Research Platform for Autonomous Behaviour," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1850, pp. 79–107, 2007.

[19] R. Brooks, "A robust layered control system for a mobile robot," *Robotics and Automation, IEEE Journal of*, vol. 2, pp. 14–23, Mar 1986.

[20] M. J. Mataric, "Interaction and intelligent behavior.," tech. rep., DTIC Document, 1994.

[21] M. P. Georgeff and A. L. Lansky, "Reactive reasoning and planning.," in *AAAI*, vol. 87, pp. 677–682, 1987.

[22] J. Zhang and A. Knoll, *Integrating Deliberative and Reactive Strategies via Fuzzy Modular Control*, pp. 367–385.
Heidelberg: Physica-Verlag HD, 2001.

[23] J. S. Albus, "The nist real-time control system (rcs): an approach to intelligent systems research," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 9, no. 2-3, pp. 157–174, 1997.

Bibliography (cont.)

[24] A. Meystel, "Nested hierarchical control," 1993.

[25] G. Saridis, "Machine-intelligent robots: A hierarchical control approach," in *Machine Intelligence and Knowledge Engineering for Robotic Applications* (A. Wong and A. Pugh, eds.), vol. 33 of *NATO ASI Series*, pp. 221–234, Springer Berlin Heidelberg, 1987.

[26] T. Fukuda and T. Shibata, "Hierarchical intelligent control for robotic motion by using fuzzy, artificial intelligence, and neural network," in *Neural Networks, 1992. IJCNN., International Joint Conference on*, vol. 1, pp. 269–274 vol.1, Jun 1992.

[27] R. C. Arkin and T. Balch, "Aura: principles and practice in review," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 9, no. 2-3, pp. 175–189, 1997.

[28] E. Gat, "Integrating reaction and planning in a heterogeneous asynchronous architecture for mobile robot navigation," *ACM SIGART Bulletin*, vol. 2, no. 4, pp. 70–74, 1991.

[29] L. Einig, *Hierarchical Plan Generation and Selection for Shortest Plans based on Experienced Execution Duration.*
Master thesis, Universität Hamburg, 2015.

Bibliography (cont.)

[30] J. Craig, *Introduction to Robotics: Mechanics & Control. Solutions Manual.* Addison-Wesley Pub. Co., 1986.

[31] H. Siegert and S. Bocionek, *Robotik: Programmierung intelligenter Roboter: Programmierung intelligenter Roboter.* Springer-Lehrbuch, Springer Berlin Heidelberg, 2013.

[32] R. Schilling, *Fundamentals of robotics: analysis and control.* Prentice Hall, 1990.

[33] T. Yoshikawa, *Foundations of Robotics: Analysis and Control.* Cambridge, MA, USA: MIT Press, 1990.

[34] M. Spong, *Robot Dynamics And Control.* Wiley India Pvt. Limited, 2008.