



Introduction to Robotics

Lecture 2

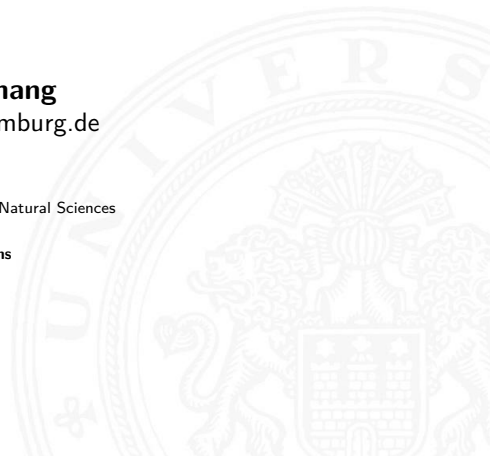
Shuang Li, Jianwei Zhang
[sli, zhang]@informatik.uni-hamburg.de



University of Hamburg
Faculty of Mathematics, Informatics and Natural Sciences
Department of Informatics

Technical Aspects of Multimodal Systems

April 16, 2021





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

Instantaneous Kinematics

Trajectory Generation 1

Trajectory Generation 2

Dynamics





Outline (cont.)

Forward Kinematics

Introduction to Robotics

Robot Control

Path Planning

Task/Manipulation Planning

Telerobotics

Architectures of Sensor-based Intelligent Systems

Summary

Conclusion and Outlook





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





In order to find the desired end effector pose:

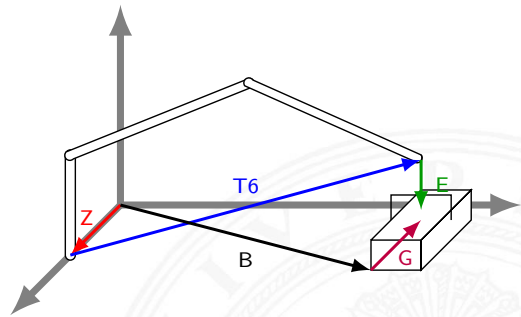
$$ZT_6E = BG$$

In order to find the manipulator transformation T_6 :

$$T_6 = Z^{-1}BGE^{-1}$$

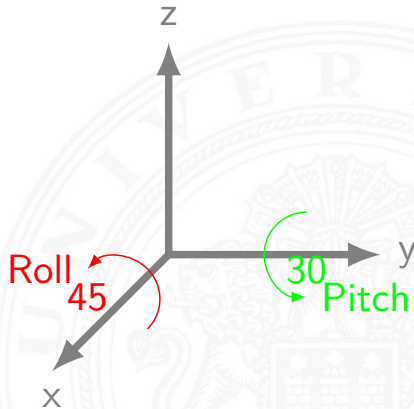
In order to determine the pose of the object B :

$$B = ZT_6EG^{-1}$$



A vector $\vec{A}P$ 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.

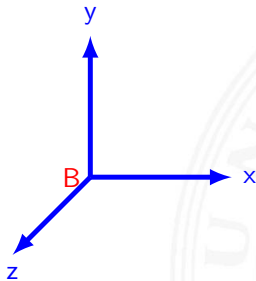
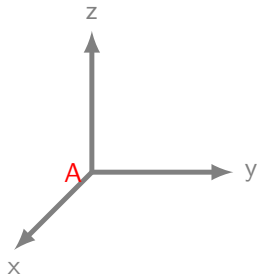
$$\begin{aligned}
 R &= R_{x,45}R_{y,30} \\
 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos 45 & -\sin 45 \\ 0 & \sin 45 & \cos 45 \end{bmatrix} \begin{bmatrix} \cos 30 & 0 & \sin 30 \\ 0 & 1 & 0 \\ -\sin 30 & 0 & \cos 30 \end{bmatrix} \\
 &= \begin{bmatrix} 0.866 & 0 & 0.5 \\ 0.353 & 0.707 & -0.612 \\ -0.353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





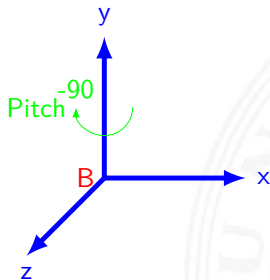
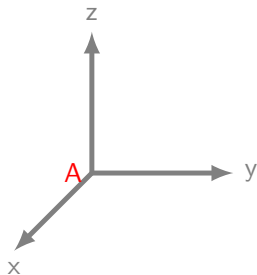
More on presentation of orientation: Euler angles

- ▶ Euler angles φ, θ, ψ



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

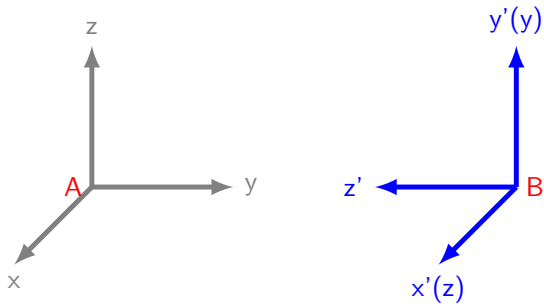
- ▶ Euler-angles φ, θ, ψ





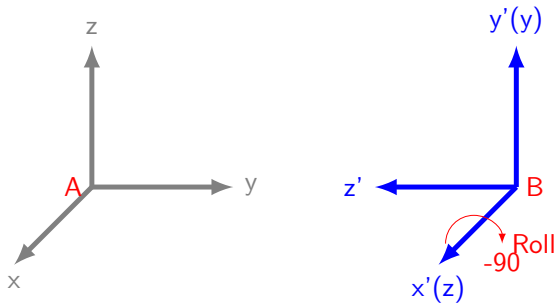
More on presentation of orientation: Euler angles (cont.)

- ▶ Euler-angles φ, θ, ψ



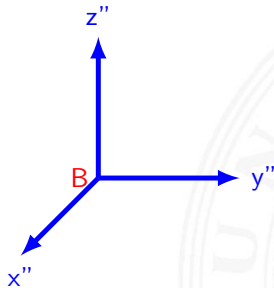
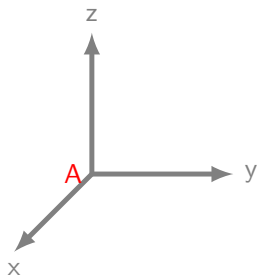
More on presentation of orientation: Euler angles (cont.)

- ▶ Euler-angles φ, θ, ψ



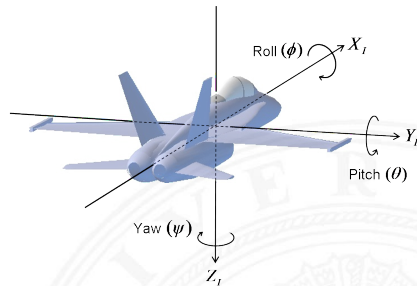
More on presentation of orientation: Euler angles (cont.)

- ▶ 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





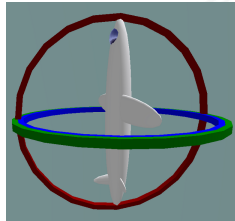
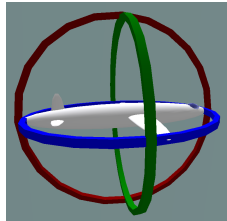
Converting Euler Angles to a Rotation Matrix

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

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

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

- ▶ 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



- ▶ 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

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



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



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

Purpose

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



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

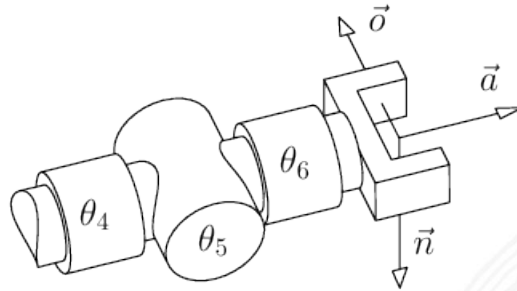
Three unit vectors:

- ▶ \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
 - ⇒ Formation of a recursive chain
 - ▶ Joint coordinates:

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





- ▶ 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:
 - ▶ 0T_1 : depicts position and orientation of the first link with respect to the base
 - ▶ \vdots
 - ▶ 5T_6 : depicts position and orientation of the 6th link in regard to link 5

The resulting product is defined as:

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



- ▶ 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



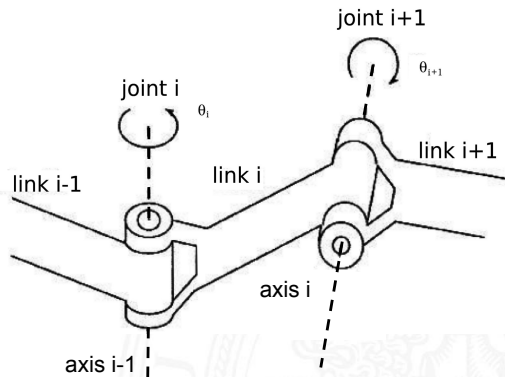
- ▶ first published by Denavit and Hartenberg in 1955
- ▶ established principle
- ▶ determination of a transformation matrix T_i using **four** parameters
 - ▶ link length, link twist, link offset and joint angle
($a_i, \alpha_i, d_i, \theta_i$)



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



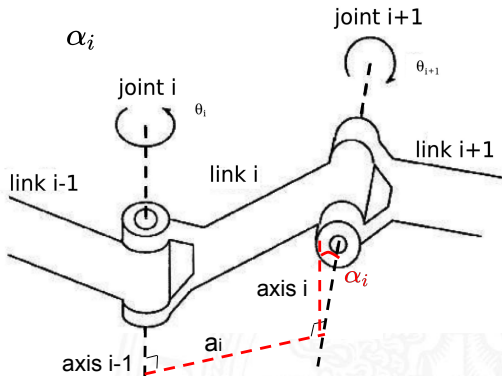


Parameters for description of two arbitrary links

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

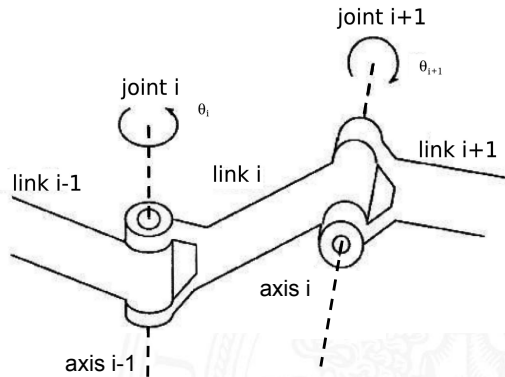
a_i and α_i are constant values due to construction





Two for relative distance and angle of adjacent links

- ▶ link offset d_i
- ▶ joint angle θ_i



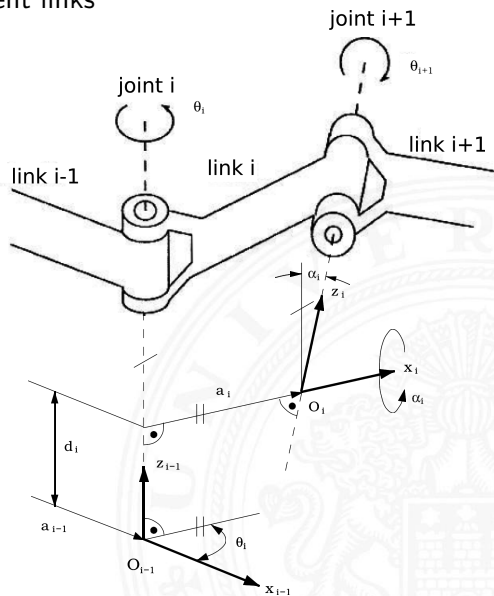
Parameters for describing two arbitrary links (cont.)

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





Four DH parameters:

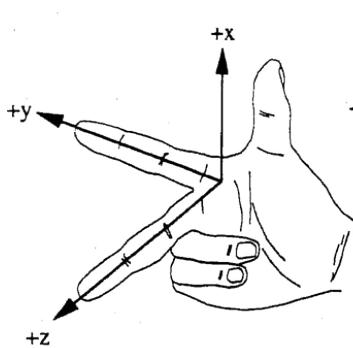
link length, link twist, link offset and joint angle

$(a_i, \alpha_i, d_i, \theta_i)$

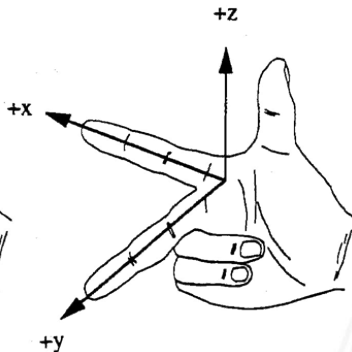
- ▶ 3 fixed link parameters
- ▶ one joint variable
 - ▶ revoluted: θ_i variable
 - ▶ prismatic: d_i variable
- ▶ a_i, α_i : describe the link i
- ▶ d_i, θ_i : describe the link's connection



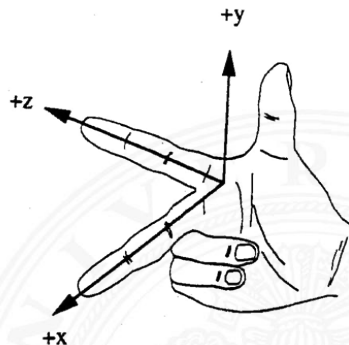
Right-Handed Coordinate System



Configuration 1

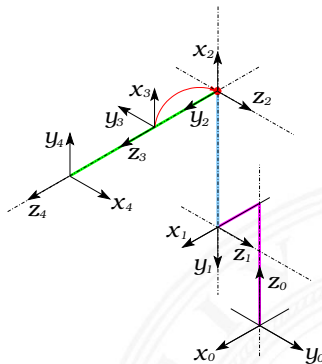
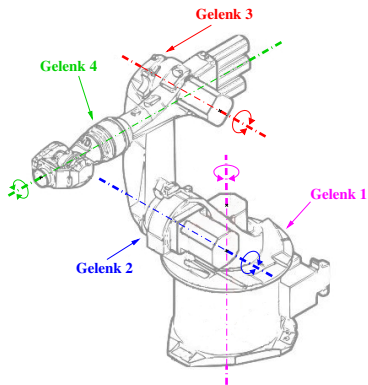


Configuration 2



Configuration 3

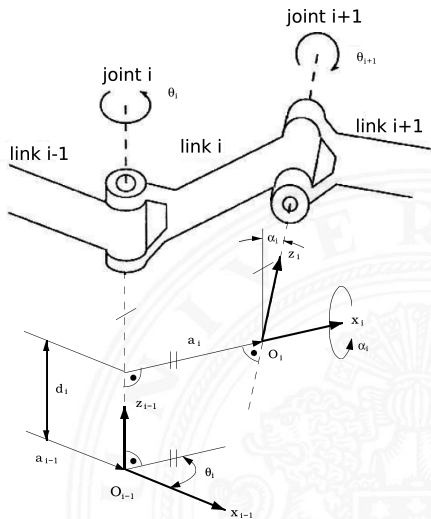
Definition of joint coordinate systems (classic)



- ▶ 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

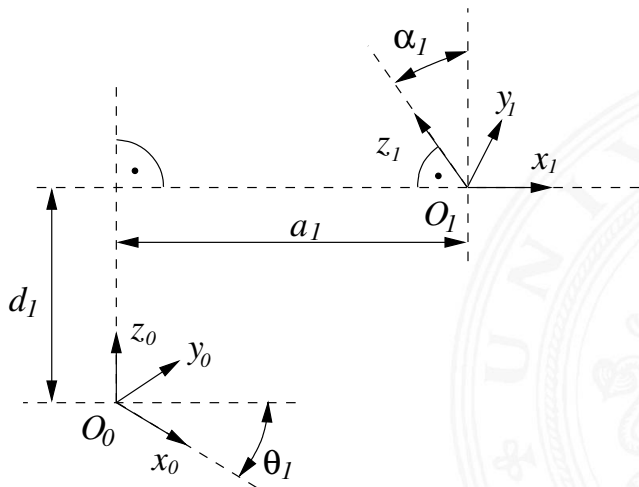


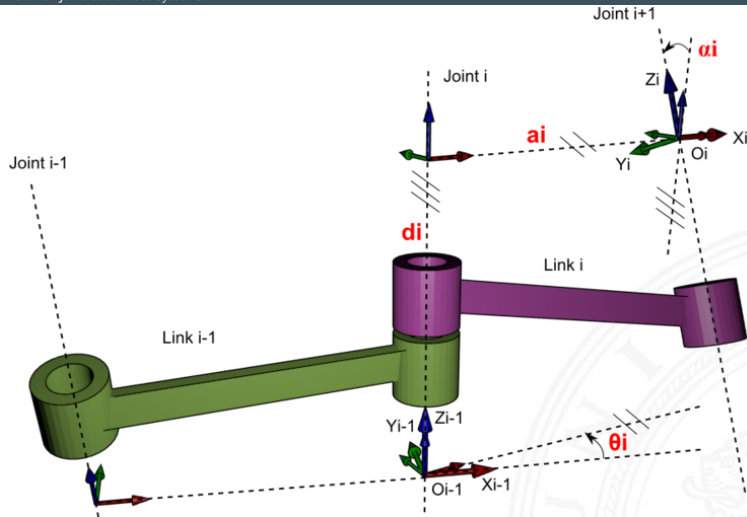
DH-Parameters and -Preconditions (classic)

Idea: Determination of the transformation matrix ${}^{i-1}T_i$ using four joint parameters $(a_i, \alpha_i, d_i, \theta_i)$ and two preconditions

Precondition 1 x_i is perpendicular to z_{i-1}

Precondition 2 x_i intersects z_{i-1}





Transformation order

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



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

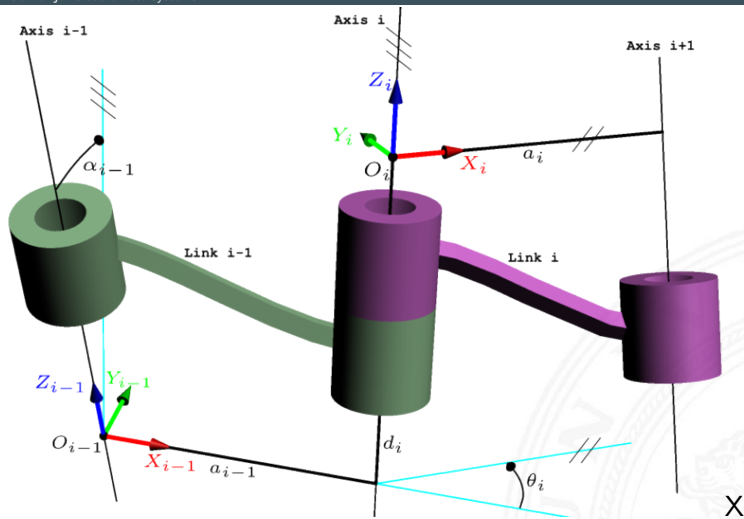
Using the product of four homogeneous transformations, which transform the coordinate frame $i - 1$ into the coordinate frame i , the matrix T_i can be calculated as follows:

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

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

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

Modified Parameters



Transformation order

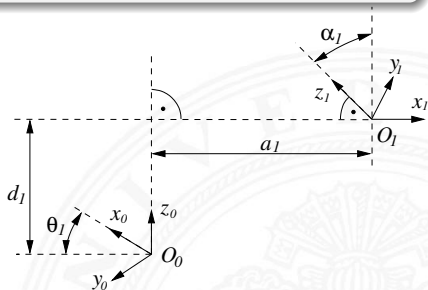
$${}^{i-1}T_i = R_{X_{i-1}}(\alpha_{i-1}) \cdot T_{X_{i-1}}(a_{i-1}) \cdot R_{Z_i}(\theta_i) \cdot T_{Z_i}(d_i) \rightarrow CS_i$$

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 ambiguous, 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

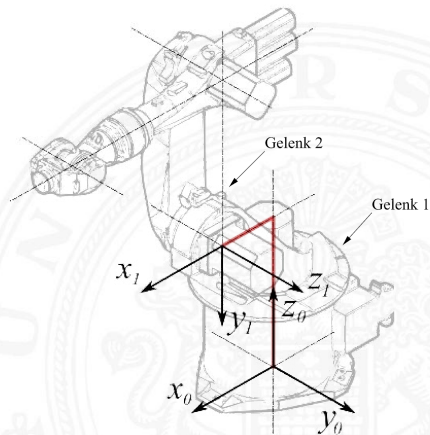
Determination of DH-Parameter (θ, d, a, α) for calculation of joint transformation:

$$T_1 = R_z(\theta_1) T_z(d_1) T_x(a_1) R_x(\alpha_1)$$

joint angle rotate by θ_1 around z_0 , such that x_0 is parallel to x_1

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

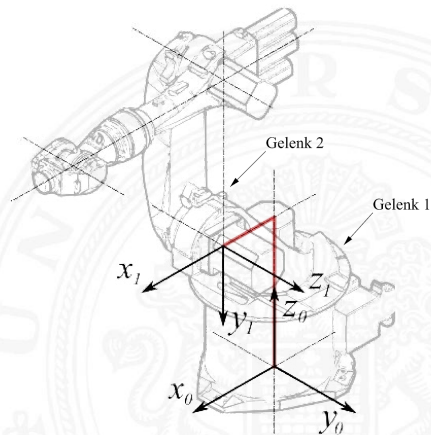
for the shown joint configuration $\theta_1 = 0^\circ$



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

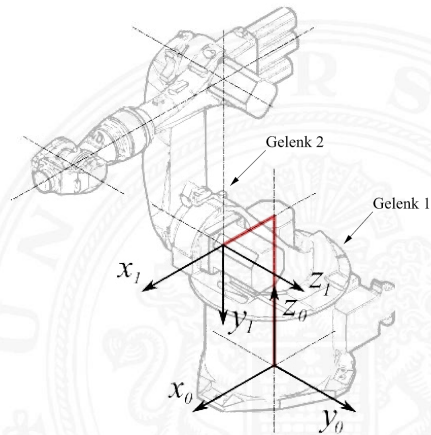
$$T_z(d_1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



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

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

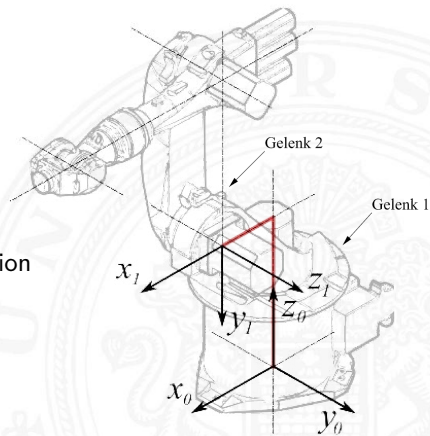


Example DH-Parameter of a single joint (cont.)

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, $\alpha_1 = -90^\circ$ due to construction



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

- ▶ total transformation of CS_0 to CS_1 (general case)

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

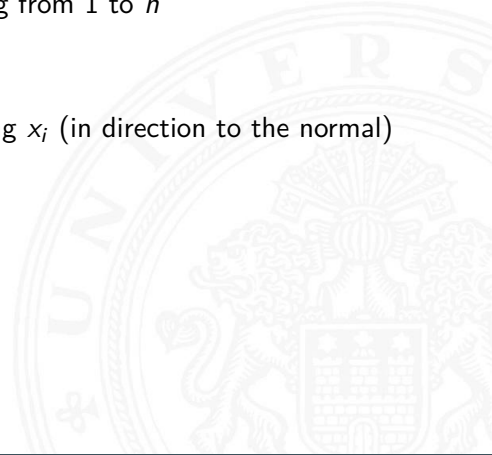
- ▶ rotary case: variable θ_1 and fixed d_1, a_1 und ($\alpha_1 = -90^\circ$)

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



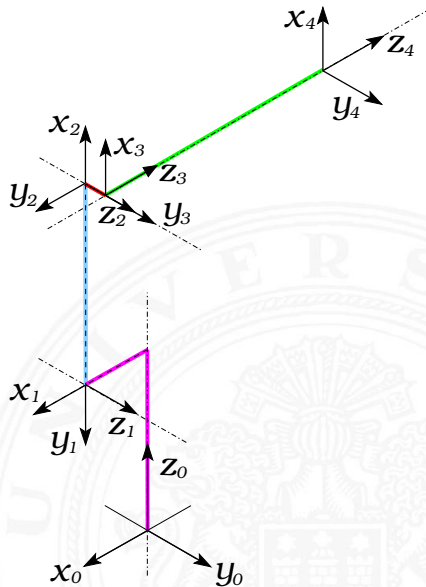
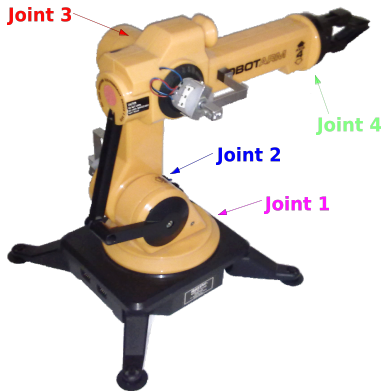
Procedure for predefined structure

- ▶ 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 away 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

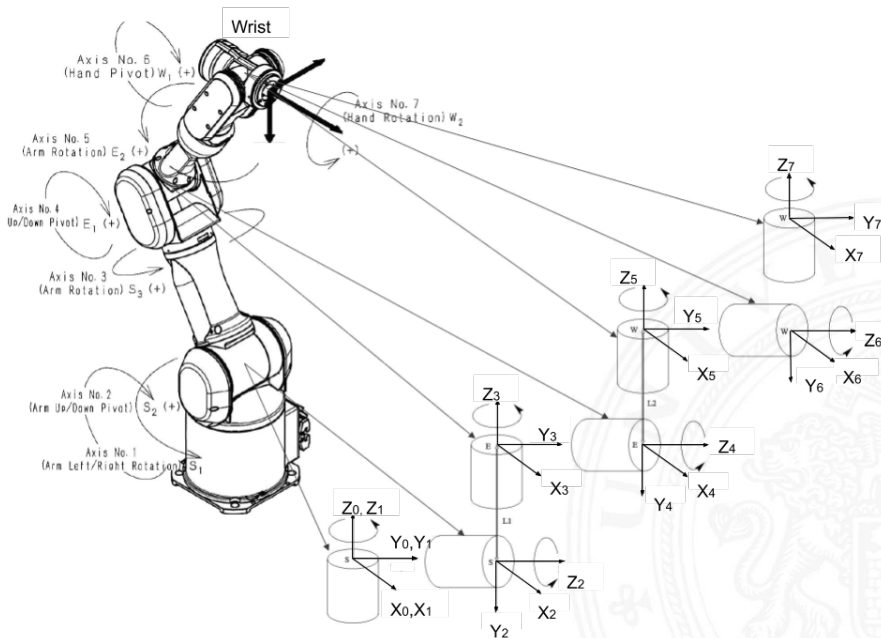
$$\begin{aligned} T_6 &= T_1 \cdot T_2 \cdot T_3 \cdot T_4 \\ &= \begin{bmatrix} \cos \theta_1 & 0 & -\sin \theta_1 & 20 \cos \theta_1 \\ \sin \theta_1 & 0 & \cos \theta_1 & 20 \sin \theta_1 \\ 0 & -1 & 0 & 100 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & 160 \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & 160 \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &\quad \begin{bmatrix} \cos \theta_3 & 0 & \sin \theta_3 & 0 \\ \sin \theta_3 & 0 & -\cos \theta_3 & 0 \\ 0 & 1 & 0 & 28 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta_4 & -\sin \theta_4 & 0 & 0 \\ \sin \theta_4 & \cos \theta_4 & 0 & 0 \\ 0 & 0 & 1 & 250 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos \theta_1 \cos \theta_4 (\cos \theta_2 \cos \theta_3 - \sin \theta_2 \sin \theta_3) - \sin \theta_1 \sin \theta_4 & \dots & \dots & \dots \\ \sin \theta_1 \cos \theta_4 (\sin \theta_2 \cos \theta_3 + \cos \theta_2 \sin \theta_3) + \cos \theta_1 \sin \theta_4 & \dots & \dots & \dots \\ -\cos \theta_4 (\sin \theta_2 \cos \theta_3 + \cos \theta_2 \sin \theta_3) & \dots & \dots & \dots \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

Sum-of-Angle formula

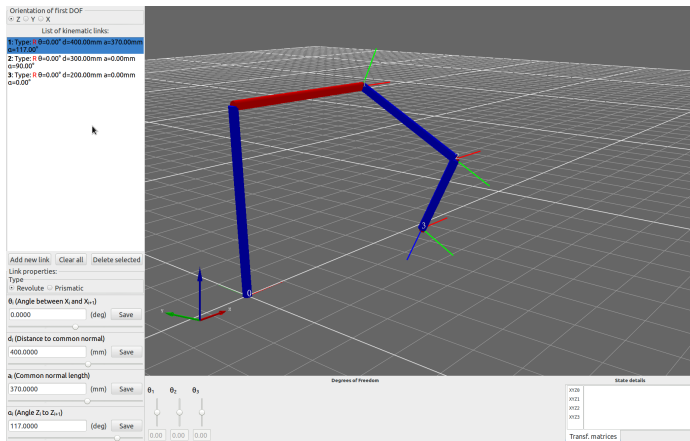
$$C_{23} = C_2 C_3 - S_2 S_3,$$

$$S_{23} = C_2 S_3 + S_2 C_3$$

Mitsubishi PA10-7C



Download link



18



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



- [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.



- [7] T. Flash and N. Hogan, "The coordination of arm movements: an experimentally confirmed mathematical model," *Journal of neuroscience*, vol. 5, no. 7, pp. 1688–1703, 1985.
- [8] T. Kröger and F. M. Wahl, "Online trajectory generation: Basic concepts for instantaneous reactions to unforeseen events," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 94–111, 2009.
- [9] 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.
- [10] 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.
- [11] 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.



- [12] A. Cowley, W. Marshall, B. Cohen, and C. J. Taylor, “Depth space collision detection for motion planning,” 2013.
- [13] Hornung, Armin and Wurm, Kai M. and Bennewitz, Maren and Stachniss, Cyrill and Burgard, Wolfram, “OctoMap: an efficient probabilistic 3D mapping framework based on octrees,” *Autonomous Robots*, vol. 34, pp. 189–206, 2013.
- [14] D. Berenson, S. S. Srinivasa, D. Ferguson, and J. J. Kuffner, “Manipulation planning on constraint manifolds,” in *2009 IEEE International Conference on Robotics and Automation*, pp. 625–632, 2009.
- [15] S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.
- [16] O. Khatib, “The Potential Field Approach and Operational Space Formulation in Robot Control,” in *Adaptive and Learning Systems*, pp. 367–377, Springer, 1986.
- [17] L. E. Kavraki, P. Svestka, J. Latombe, and M. H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.



- [18] J. Kuffner and S. LaValle, "RRT-Connect: An Efficient Approach to Single-Query Path Planning.," vol. 2, pp. 995–1001, 01 2000.
- [19] J. Starek, J. Gómez, E. Schmerling, L. Janson, L. Moreno, and M. Pavone, "An asymptotically-optimal sampling-based algorithm for bi-directional motion planning," *Proceedings of the ... IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2015, 07 2015.
- [20] D. Hsu, J. . Latombe, and R. Motwani, "Path planning in expansive configuration spaces," in *Proceedings of International Conference on Robotics and Automation*, vol. 3, pp. 2719–2726 vol.3, 1997.
- [21] A. H. Qureshi, A. Simeonov, M. J. Bency, and M. C. Yip, "Motion planning networks," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 2118–2124, IEEE, 2019.
- [22] J. Schulman, J. Ho, A. Lee, I. Awwal, H. Bradlow, and P. Abbeel, "Finding locally optimal, collision-free trajectories with sequential convex optimization," in *in Proc. Robotics: Science and Systems*, 2013.



- [23] A. T. Miller and P. K. Allen, “Graspit! a versatile simulator for robotic grasping,” *IEEE Robotics Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004.
- [24] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, “Grasp pose detection in point clouds,” *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017.
- [25] L. P. Kaelbling and T. Lozano-Pérez, “Hierarchical task and motion planning in the now,” in *2011 IEEE International Conference on Robotics and Automation*, pp. 1470–1477, 2011.
- [26] N. T. Dantam, Z. K. Kingston, S. Chaudhuri, and L. E. Kavraki, “Incremental task and motion planning: A constraint-based approach.,” in *Robotics: Science and Systems*, pp. 1–6, 2016.
- [27] J. Ferrer-Mestres, G. Francès, and H. Geffner, “Combined task and motion planning as classical ai planning,” *arXiv preprint arXiv:1706.06927*, 2017.
- [28] M. Görner, R. Haschke, H. Ritter, and J. Zhang, “Movelt! Task Constructor for Task-Level Motion Planning,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.



- [29] K. Hauser and J.-C. Latombe, "Multi-modal motion planning in non-expansive spaces," *The International Journal of Robotics Research*, vol. 29, no. 7, pp. 897–915, 2010.
- [30] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Springer, 2016.
- [31] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, S. Levine, and G. Brain, "Time-contrastive networks: Self-supervised learning from video," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1134–1141, IEEE, 2018.
- [32] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," *arXiv preprint arXiv:1703.03400*, 2017.
- [33] R. Brooks, "A robust layered control system for a mobile robot," *Robotics and Automation, IEEE Journal of*, vol. 2, pp. 14–23, Mar 1986.
- [34] M. J. Mataric, "Interaction and intelligent behavior.," tech. rep., DTIC Document, 1994.



- [35] M. P. Georgeff and A. L. Lansky, "Reactive reasoning and planning.," in *AAAI*, vol. 87, pp. 677–682, 1987.
- [36] 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.
- [37] 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.
- [38] L. Einig, *Hierarchical Plan Generation and Selection for Shortest Plans based on Experienced Execution Duration*.
Master thesis, Universität Hamburg, 2015.
- [39] J. Craig, *Introduction to Robotics: Mechanics & Control. Solutions Manual*.
Addison-Wesley Pub. Co., 1986.



- [40] H. Siegert and S. Bocionek, *Robotik: Programmierung intelligenter Roboter: Programmierung intelligenter Roboter*. Springer-Lehrbuch, Springer Berlin Heidelberg, 2013.
- [41] R. Schilling, *Fundamentals of robotics: analysis and control*. Prentice Hall, 1990.
- [42] T. Yoshikawa, *Foundations of Robotics: Analysis and Control*. Cambridge, MA, USA: MIT Press, 1990.
- [43] M. Spong, *Robot Dynamics And Control*. Wiley India Pvt. Limited, 2008.

