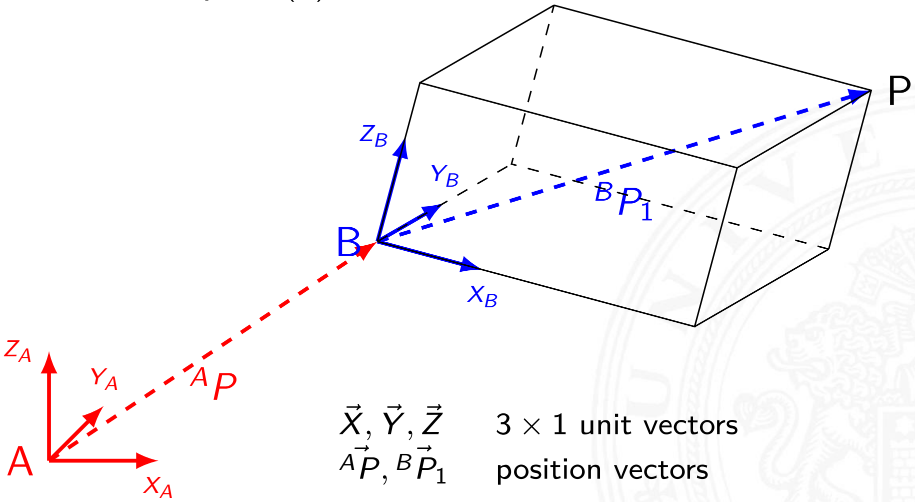
Outline

Outline (cont.)

Coordinate Systems

The **pose** of objects, in other words their **position** and **orientation** in Euclidian space can be described through specification of a cartesian coordinate system (B) in relation to a base coordinate system (A).



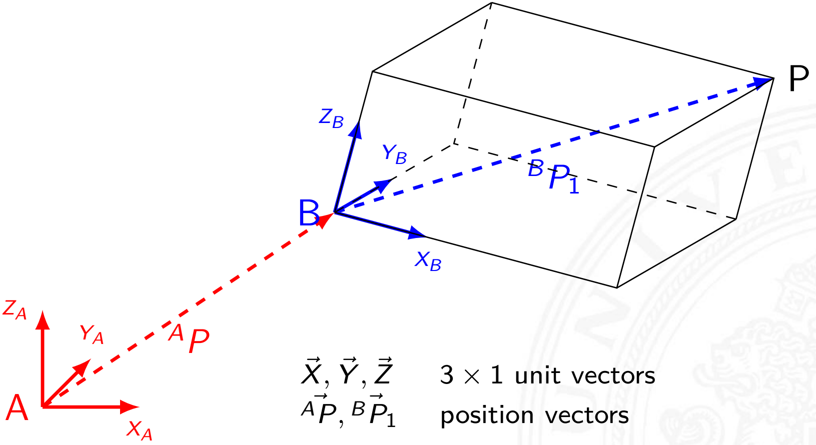
$\vec{X}, \vec{Y}, \vec{Z}$ 3 × 1 unit vectors

$^A\vec{P}, {}^B\vec{P}_1$ position vectors

Specification of position and orientation

Position:

- translation along the axes of the base coordinate system (A)



$\vec{X}, \vec{Y}, \vec{Z}$	3×1 unit vectors
$^A\vec{P}, {}^B\vec{P}_1$	position vectors

- given by position vector $\vec{^A\mathbf{P}} = [^A p_x, ^A p_y, ^A p_z]^T \in \mathcal{R}^3$

`Orientation` (in space):

- given by Rotation matrix $R_B = [\vec{X}_B \ \vec{Y}_B \ \vec{Z}_B] \in \mathcal{R}^{3\times3}$
- given by Rotation matrix ${}^AR_B = [{}^A\vec{X}_B \ {}^A\vec{Y}_B \ {}^A\vec{Z}_B] \in \mathcal{R}^{3\times3}$

- ${}^AR_B$: the orientation of B with respect to A.
  (Latex: `$^{A}R_{B}$`)
- ${}^A\vec{X}_B, {}^A\vec{Y}_B, {}^A\vec{Z}_B$ are projection of $\vec{X}_B$, $\vec{Y}_B$, $\vec{Z}_B$ in A.

Rotation matrix

Dot product

In terms of the geometric definition, the dot product of two unit vectors $\vec{a}$ and $\vec{b}$ means the projection of the $\vec{a}$ in $\vec{b}$.
$\vec{a} \cdot \vec{b} = \|a\|\|b\| \cos(\theta)$

$$^A\vec{X}_B = \begin{bmatrix} \vec{X}_B \cdot \vec{X}_A \\ \vec{X}_B \cdot \vec{Y}_A \\ \vec{X}_B \cdot \vec{Z}_A \end{bmatrix} \quad \text{and} \quad {}^AR_B = \begin{bmatrix} {}^A\vec{X}_B & {}^A\vec{Y}_B & {}^A\vec{Z}_B \end{bmatrix}$$

$$^AR_B = \begin{bmatrix} \vec{X}_B \cdot \vec{X}_A & \vec{Y}_B \cdot \vec{X}_A & \vec{Z}_B \cdot \vec{X}_A \\ \vec{X}_B \cdot \vec{Y}_A & \vec{Y}_B \cdot \vec{Y}_A & \vec{Z}_B \cdot \vec{Y}_A \\ \vec{X}_B \cdot \vec{Z}_A & \vec{Y}_B \cdot \vec{Z}_A & \vec{Z}_B \cdot \vec{Z}_A \end{bmatrix}$$

Inverse of rotation matrix

$$
\begin{array}{ll}
{}^{A}R_{B} = \begin{bmatrix} \vec{X}_B \cdot \vec{X}_A & \vec{Y}_B \cdot \vec{X}_A & \vec{Z}_B \cdot \vec{X}_A \\ \vec{X}_B \cdot \vec{Y}_A & \vec{Y}_B \cdot \vec{Y}_A & \vec{Z}_B \cdot \vec{Y}_A \\ \vec{X}_B \cdot \vec{Z}_A & \vec{Y}_B \cdot \vec{Z}_A & \vec{Z}_B \cdot \vec{Z}_A \end{bmatrix} & \begin{array}{l} {}^{B}X_A^T \\ \text{the projection of } \vec{X}_A \text{ in B} \\ \\ \end{array}
\end{array}
$$

$$
{}^{A}R_{B} = \begin{bmatrix} {}^{A}\vec{X}_B & {}^{A}\vec{Y}_B & {}^{A}\vec{Z}_B \end{bmatrix} = \begin{bmatrix} {}^{B}\vec{X}_A^T \\ {}^{B}\vec{Y}_A^T \\ {}^{B}\vec{Z}_A^T \end{bmatrix} = \begin{bmatrix} {}^{B}\vec{X}_A & {}^{B}\vec{Y}_A & {}^{B}\vec{Z}_A \end{bmatrix}^T = {}^{B}R_A^T
$$

Inverse of rotation matrix (cont.)

$$^{A}R_{B} = \begin{bmatrix} ^{A}\vec{X}_{B} & ^{A}\vec{Y}_{B} & ^{A}\vec{Z}_{B} \end{bmatrix} = \begin{bmatrix} ^{B}\vec{X}_{A}^{T} \\ ^{B}\vec{Y}_{A}^{T} \\ ^{B}\vec{Z}_{A}^{T} \end{bmatrix} = \begin{bmatrix} ^{B}\vec{X}_{A} & ^{B}\vec{Y}_{A} & ^{B}\vec{Z}_{A} \end{bmatrix}^{T} = {^{B}R_{A}^{T}}$$

The inverse of a rotation matrix is simply its transpose:

$$^{A}R_{B}^{-1} = {^{B}R_{A}} = {^{B}R_{A}^{T}} \quad \text{and} \quad {^{A}R_{B}}{^{B}R_{A}} = I$$

whereas I is the identity matrix.

Specification of position and orientation

- Position:
  - given through ${}^{A}\vec{P} \in \mathcal{R}^3$
- Orientation:
  - given through the projection of $\vec{X}_B, \vec{Y}_B, \vec{Z}_B \in \mathcal{R}^3$ of B to the origin system A
  - summarized to rotation matrix ${}^{A}R_B = [{}^{A}\vec{X}_B \; {}^{A}\vec{Y}_B \; {}^{A}\vec{Z}_B] \in \mathcal{R}^{3\times3}$

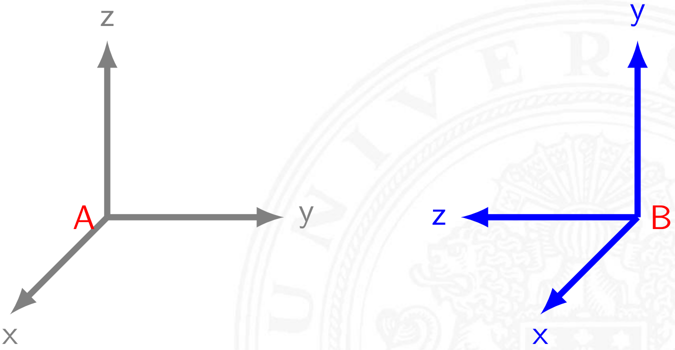
$$
{}^{A}R_B = \begin{bmatrix} r_{11} & r_{21} & r_{31} \\ r_{12} & r_{22} & r_{32} \\ r_{13} & r_{23} & r_{33} \end{bmatrix}
$$

  - redundant, since there are 9 parameters for 3 degrees of freedom

Example of rotation matrix

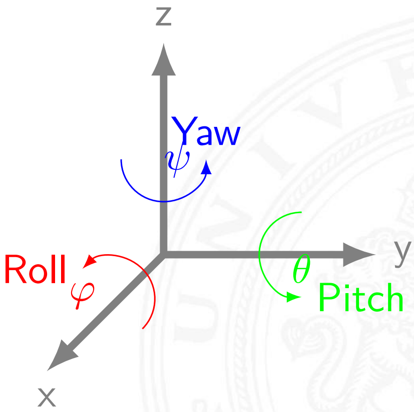
Write the Rotation matrix of ${}^{A}R_{B}$.
${}^{A}R_{B} = [{}^{A}\vec{X}_{B}\ {}^{A}\vec{Y}_{B}\ {}^{A}\vec{Z}_{B}]$

$$
{}^{A}R_{B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}
$$



Sequential multiplication of the rotation matrices by order of rotation.

1. rotation φ (*phi*) around the x-axis
   $R_{x,\varphi}$ – Roll
2. rotation θ (*theta*) around the y-axis
   $R_{y,\theta}$ – Pitch
3. rotation ψ (*psi*) around the z-axis
   $R_{z,\psi}$ – Yaw



Rotatory transformation

(shortened representation: S : sin, C : cos)
The rotation matrix corresponding to a rotation around the x-axis with angle φ (*phi*):

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

Rotatory transformation (cont.)

The rotation matrix corresponding to a rotation around the y-axis with angle θ (*theta*):

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

Rotatory transformation (cont.)

The rotation matrix corresponding to a rotation around the *z*-axis with angle ψ (*psi*):

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

Concatenation of rotation matrices

$$R_{\psi,\theta,\varphi} = R_{z,\psi} R_{y,\theta} R_{x,\varphi}$$

$$= \begin{bmatrix} C\psi & -S\psi & 0 \\ S\psi & C\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C\theta & 0 & S\theta \\ 0 & 1 & 0 \\ -S\theta & 0 & C\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\varphi & -S\varphi \\ 0 & S\varphi & C\varphi \end{bmatrix}$$

$$= \begin{bmatrix} C\psi C\theta & C\psi S\theta S\varphi - S\psi C\varphi & C\psi S\theta C\varphi + S\psi S\varphi \\ S\psi C\theta & S\psi S\theta S\varphi + C\psi C\varphi & S\psi S\theta C\varphi - C\psi S\varphi \\ -S\theta & C\theta S\varphi & C\theta C\varphi \end{bmatrix}$$

Remark: Matrix multiplication is not commutative:

$$AB \neq BA$$

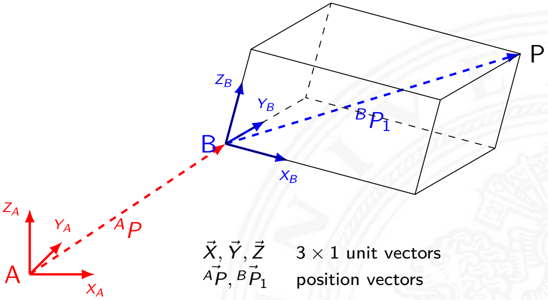
Concatenation of rotation matrices

- `Several rotations` can be multiplied. The following applies:
  - If the rotations are performed in relation to the current, newly defined (or changed) coordinate system, the newly added transformation matrices need to be multiplicatively appended on the right-hand side.
  - If all of them are performed in relation to the fixed reference coordinate system, the transformation matrices need to be multiplicatively appended on the left-hand side.

Mapping by rotation matrix

Mapping: changing descriptions from frame to frame.
For example, change the reference frame of ${}^{B}\vec{P}_1$?

$$
\begin{aligned}
{}^{A}\vec{P}_1 &= \begin{bmatrix} {}^{B}\vec{X}_A \cdot {}^{B}\vec{P}_1 \\ {}^{B}\vec{Y}_A \cdot {}^{B}\vec{P}_1 \\ {}^{B}\vec{Z}_A \cdot {}^{B}\vec{P}_1 \end{bmatrix} \\
&= \begin{bmatrix} {}^{B}\vec{X}_A^{T} \\ {}^{B}\vec{Y}_A^{T} \\ {}^{B}\vec{Z}_A^{T} \end{bmatrix} \cdot {}^{B}\vec{P}_1 \\
&= {}^{A}R_B {}^{B}\vec{P}_1
\end{aligned}
$$



$\vec{X}, \vec{Y}, \vec{Z}$ 3 × 1 unit vectors
${}^{A}\vec{P}, {}^{B}\vec{P}_1$ position vectors

Summary: three common uses of a rotation matrix

Three common uses of a rotation matrix:

- represent an orientation
- rotate a vector or frame
- change the frame of reference of a vector or frame

Homogenous transformation

- `Homogeneous transformation matrix:`

$$T = \begin{bmatrix} R & \vec{p} \\ P & S \end{bmatrix}$$

where P depicts the perspective transformation and S the scaling.

- In robotics, $P = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$ and $S = 1$. Other values are used for computer graphics.

Homogenous transformation (cont.)

- Combination of $\vec{p}$ and R to $T = \begin{bmatrix} R & \vec{p} \\ \vec{0} & 1 \end{bmatrix} \in \mathcal{R}^{4\times4}$
- Concatenation of several T through matrix multiplication
  - ${}^{A}T_{B}\,{}^{B}T_{C} = {}^{A}T_{C}$
- not commutative, in other words ${}^{B}T_{C}\,{}^{A}T_{B} \neq {}^{A}T_{B}\,{}^{B}T_{C}$

They are represented as four vectors using the elements of homogeneous transformation.

$$T = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{21} & r_{31} & p_x \\ r_{12} & r_{22} & r_{32} & p_y \\ r_{13} & r_{23} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

Inverse transformation

The inverse of a rotation matrix is simply its transpose:

$$R^{-1} = R^T \text{ and } RR^T = I$$

whereas I is the identity matrix.
The inverse of (1) is:

$$T^{-1} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & -\mathbf{p}^\mathbf{T} \cdot \mathbf{r}_1 \\ r_{21} & r_{22} & r_{23} & -\mathbf{p}^\mathbf{T} \cdot \mathbf{r}_2 \\ r_{31} & r_{32} & r_{33} & -\mathbf{p}^\mathbf{T} \cdot \mathbf{r}_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

whereas $\mathbf{r}_1$, $\mathbf{r}_2$, $\mathbf{r}_3$ and $\mathbf{p}$ are the four column vectors of (1) and $\cdot$ represents the dot product of vectors.

Translatory transformation

A translation with a vector $[p_x, p_y, p_z]^T$ is expressed through a transformation:

$$T_{(p_x,p_y,p_z)} = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotatory transformation

The transformation corresponding to a rotation around the x-axis with angle φ (*phi*):

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

The transformation corresponding to a rotation around the y-axis with angle θ (*theta*):

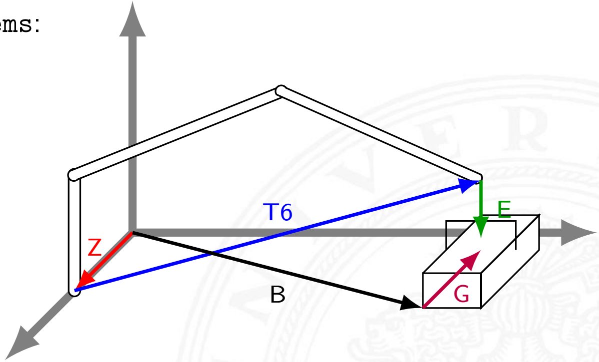
$$T_{y,\theta} = \begin{bmatrix} C\theta & 0 & S\theta & 0 \\ 0 & 1 & 0 & 0 \\ -S\theta & 0 & C\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotatory transformation (cont.)

The transformation corresponding to a rotation around the *z*-axis with angle ψ (*psi*):

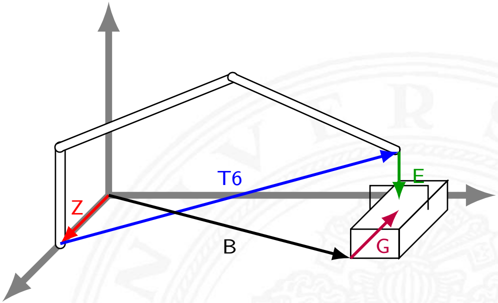
$$T_{z,\psi} = \begin{bmatrix} C\psi & -S\psi & 0 & 0 \\ S\psi & C\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- `Transform of Coordinate systems`:
  frame: a reference S
  typical frames:
  - robot base
  - end effector
  - table (world)
  -
  - object
  - camera
  - ...



One has the following transformations:

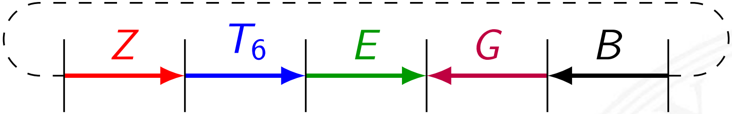
- Z :
  World $\rightarrow$ Manipulator base
- T_6 :
  Manipulator base $\rightarrow$ Manipulator end
- E :
  Manipulator end $\rightarrow$ End effector
- B :
  World $\rightarrow$ Object
- G :
  Object $\rightarrow$ End effector



Transformation equation

There are two descriptions for the desired end effector pose, one in relation to the object and the other in relation to the manipulator. Both descriptions should equal to each other for grasping:

$$Z T_6 E = B G$$



In order to find the manipulator transformation:

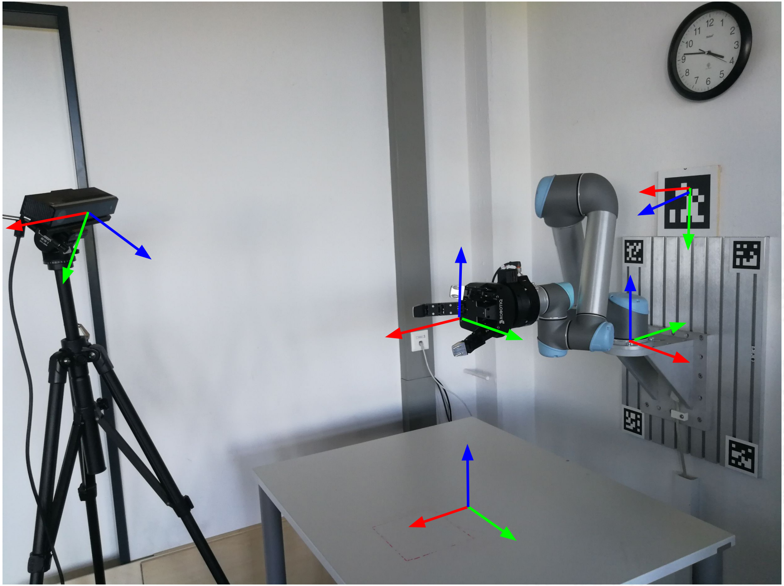
$$T_6 = Z^{-1} B G E^{-1}$$

In order to determine the pose of the object:

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

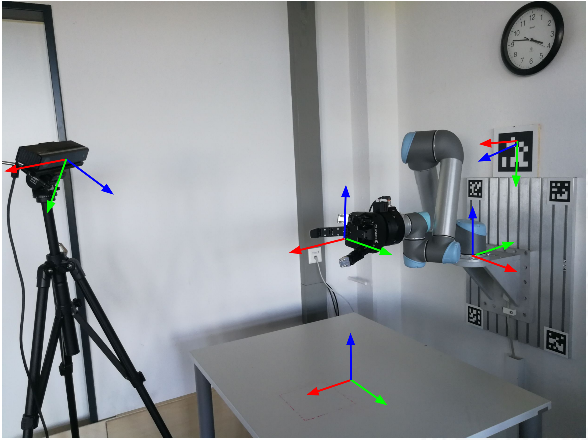
This is also called kinematic chain.

Example: coordinate transformation



Example: coordinate transformation

Given $T_{Base-Apriltag}$, $T_{Camera-Apritag}$, $T_{Camera-Object}$, calculate $T_{Base-Object}$.



$$T_{Base-Object} = T_{Base-Apriltag} T^{-1}_{Camera-Apritag} T_{Camera-Object}$$

Summary of homogeneous transformations

- `A homogeneous transformation` depicts the position and orientation of a coordinate frame in space.
- `If the coordinate frame` is defined in relation to a solid object, the position and orientation of the solid object is unambiguously specified.
- Three common uses of a transformation matrix: to represent a rigid-body configuration; to change the frame of reference of a vector or a frame; to displace a vector or a frame.

Summary of homogeneous transformations (cont.)

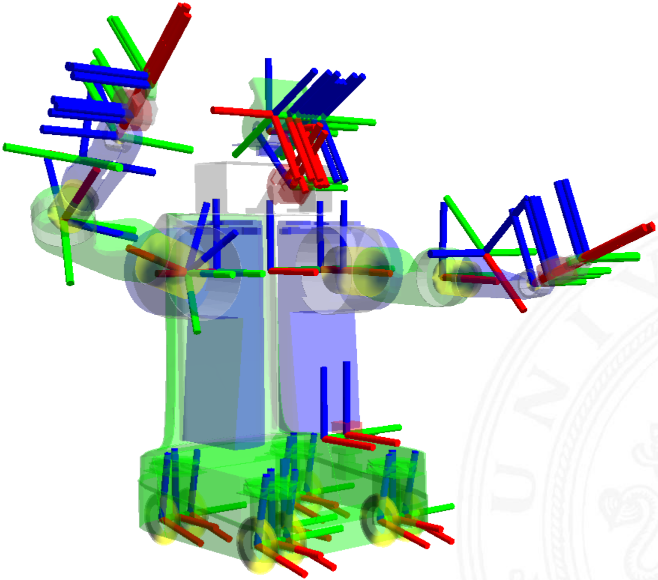
- Several translations and rotations can be multiplied.
  - right-hand multiplication → in relation to the current, newly defined (or changed) coordinate system.
  - left-hand multiplication → in relation to the fixed reference coordinate system.

Coordinates of a manipulator

- Joint coordinates:
  A vector $\mathbf{q}(t) = (q_1(t), q_2(t), ..., q_n(t))^T$
  (a robot configuration)
- End effector coordinates
  (Object coordinates):
  - A vector $\mathbf{p} = [p_x, p_y, p_z]^T$
  - Rotation matrix:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

Outlook



Outlook (cont.)

- Can we use less of 9 parameters to represent the orientation?
- How to construct the transformation matrix of the manipulator's end-effector relative to the base of the manipulator?

Suggestions

- Read (available on google & library):
  - J. F. Engelberger, *Robotics in service*. MIT Press, 1989
  - 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
  - R. Paul, *Robot Manipulators: Mathematics, Programming, and Control: the Computer Control of Robot Manipulators*. Artificial Intelligence Series, MIT Press, 1981
  - J. Craig, *Introduction to Robotics: Pearson New International Edition: Mechanics and Control*. Always learning, Pearson Education, Limited, 2013
- Repeat your linear algebra knowledge, especially regarding elementary algebra of matrices.

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] 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.

Bibliography (cont.)

[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, "MoveIt! Task Constructor for Task-Level Motion Planning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

Bibliography (cont.)

[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.

Bibliography (cont.)

[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.