Introduction to Robotics

Lecture 3

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 08, 2020

Outline

Outline (cont.)

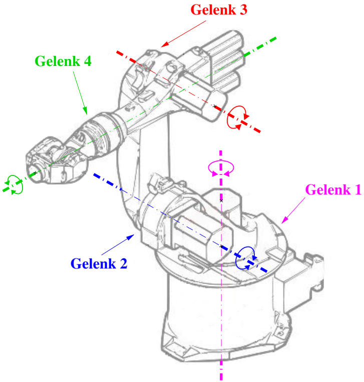
Recapitulation of DH-Parameter

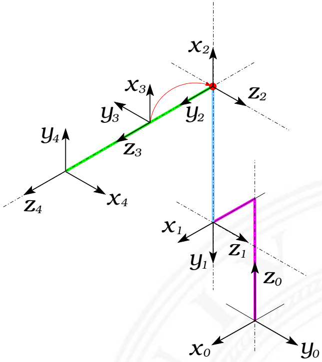
- universal minimal robot description
- based on frame transformations
- four parameters per frame transformation
- serial chain of transformations
- unique description of T_6

Drawbacks

- ambiguous convention
- only kinematic chain described
- missing information on geometry, physical constraints, dynamics, collisions, inertia, sensors, ...

Definition of joint coordinate systems





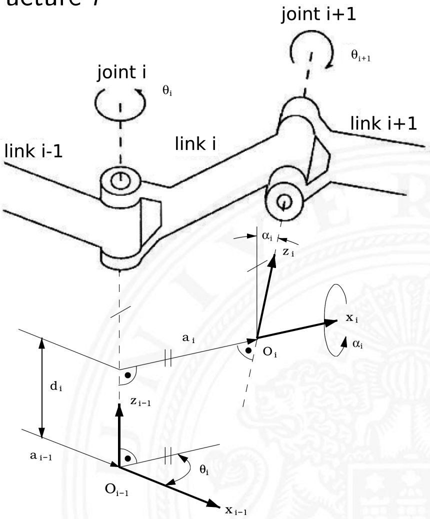
- CS_0 is the stationary origin at the base of the manipulator
- axis z_{i-1} is set along the axis of motion of the i^th joint
- axis x_i is the common normal of $z_{i-1} \times z_i$
- axis y_i concludes a right-handed coordinate system

Parameters for description of two arbitrary links

Two parameters for the description of the link structure i

- a_i : shortest distance between the z_{i-1}-axis and the z_i-axis
- α_i : rotation angle around the x_i-axis, which aligns the z_{i-1}-axis to the z_i-axis

a_i and α_i are constant values due to construction



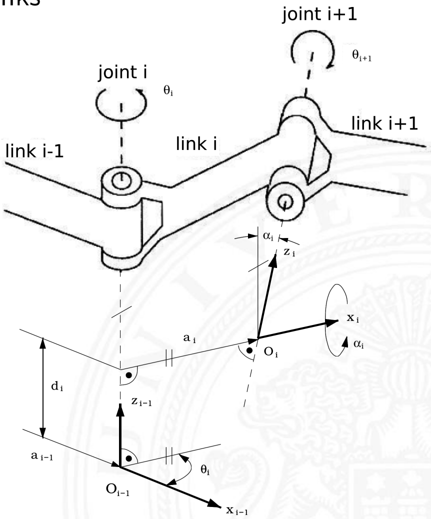
Parameters for description of two arbitrary links (cont.)

Two for relative distance and angle of adjacent links

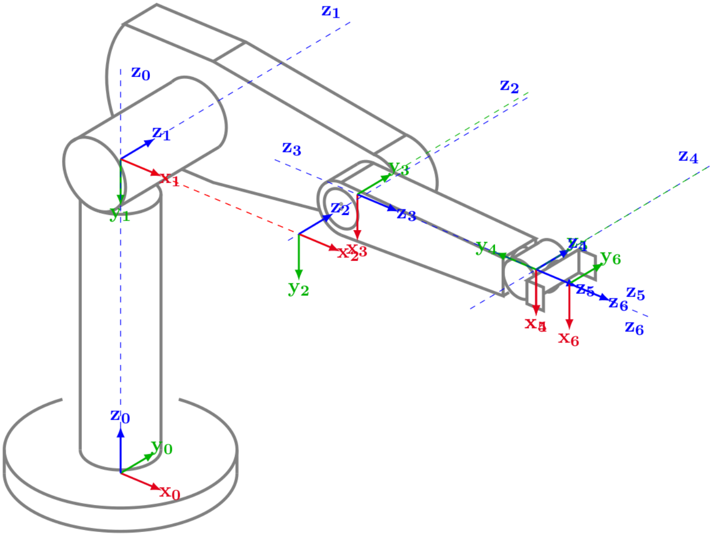
- d_i : distance origin O_{i-1} of the $(i\text{-}1)^{st}$ CS to intersection of z_{i-1}-axis with x_i-axis
- θ_i : joint angle around z_{i-1}-axis to align x_{i-1}- parallel to x_i-axis into x_{i-1}, y_{i-1}-plane

θ_i and d_i are variable

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



Example featuring PUMA 560



DH parameters of PUMA 560

Joint	a_i	α_i	d_i	θ_i
1	0	$-\frac{\pi}{2}$	d_1	θ_1^*
2	a_2	0	0	θ_2^*
3	a_3	$\frac{\pi}{2}$	d_3	θ_3^*
4	0	$-\frac{\pi}{2}$	d_4	θ_4^*
5	0	$\frac{\pi}{2}$	0	θ_5^*
6	0	0	d_6	θ_6^*

In order to transfer the manipulator-endpoint into the base coordinate system, T_6 is calculated as follows:

Universal Robot Description Format

Documentation

http://wiki.ros.org/urdf
http://wiki.ros.org/urdf/XML
http://wiki.ros.org/urdf/Tutorials

- robot description format used in ROS[19]
- hierarchical description of components
- XML format representing robot model
  - kinematics and dynamics
  - visual
  - collision
  - configuration

[19] http://ros.org

URDF: XML Tree Structure

- 1^{st}-level structure

```
<robot name="samplerobot">
</robot>
```

- 2^{nd}-level structure

  `link`, `joints`, `sensors`, `transmissions`, `gazebo`, `model_state`
- 3^{rd}-level structure

  `visual`, `inertia`, `collision`, `origin`, `parent`, . . .
- 4^{th}-level structure

  ⋮

URDF: XML Tree Structure (cont.)

- Filename: robotname.urdf
- XML prolog:

```
<?xml version="1.0" encoding="utf-8"?>
```

- XML element types

```
<tag attribute="value"/>
```

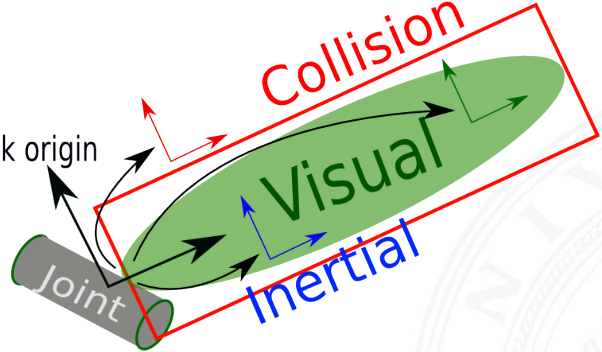
```
<tag attribute="value">
text or element(s)
</tag>
```

- XML comments

```
<!-- Comments are placed within these tags -->
```

URDF: Link

Link describes geometrical properties of a rigid body.



20

URDF: Link (cont.)

```
<link name="sample_link">
<!-- describes the mass and inertial properties of
the link -->
<inertial/>

<!-- describes the visual appearance of the link.
can be described using geometric primitives or
meshes -->
<visual/>

<!-- describes the collision space of the link.
is described like the visual appearance -->
<collision/>
</link>
```

[20] http://wiki.ros.org/urdf/XML/link

URDF: Link – visual – primitives

Geometric primitives for describing visual appearance of the link

```
<link name="base_link">
  <visual>
    <origin xyz="0 0 0.01" rpy="0 0 0"/>
    <geometry>
      <box size="0.2 0.2 0.02"/>
    </geometry>
    <material name="cyan">
      <color rgba="0 1.0 1.0 1.0"/>
    </material>
  </visual>
</link>
```

- Geometric primitives: <box>, <cylinder>, <sphere>
- Materials: <color>, <texture>

URDF: Link – visual – meshes

3D meshes for describing visual appearance of the link

```
<link name="base_link">
  <visual>
    <origin xyz="0 0 0.01" rpy="0 0 0"/>
    <geometry>
      <mesh filename="meshes/base_link.dae"
    </geometry>
  </visual>
  <collision>
     <origin xyz="0 0 0.01" rpy="0 0 0"/>
     <geometry>
          <cylinder radius="1" length="0.5"/>
     </geometry>
  </collision>
</link>
```

- the <collision> element can be simpler from the <visual> in order to reduce computation time

URDF: Link – inertial

Parameters describing the physical properties of the link

```
<link name="base_link">
  <inertial>
    <origin xyz="0 0 0" rpy="0 0 0"/>
    <mass value="1">
    <inertia ixx="100" ixy="0" ixz="0"
             iyy="100" iyz="0" izz="100" />
  </inertial>
</link>
```

- center of gravity <origin xyz>
- object mass <mass value>
- inertia tensor <intertia>

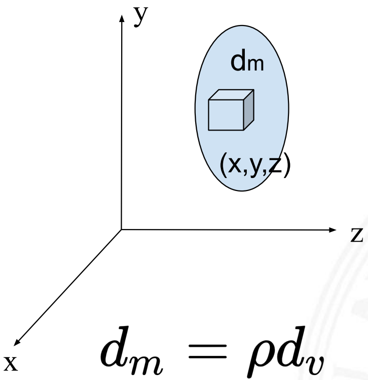
URDF: Inertia

Inertia tensor describes the distribution of mass of the link

- orientation and position of the inertia CS described by <origin> tag
- $$ {}^{A}I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & Izz \end{bmatrix} $$
  - diagonal values describe main inertial axes `ixx`, `iyy`, `izz`
  - `ixy`, `ixz`, `iyz` are 0 for symmetric primitives
  - rotations around largest and smallest inertial axis are most stable

URDF: Inertia (cont.)

- moments of inertia:
  $I_{xx} = \int(y^2 + z^2)dm$ $I_{yy} = \int(x^2 + z^2)dm$ $I_{zz} = \int(x^2 + y^2)dm$

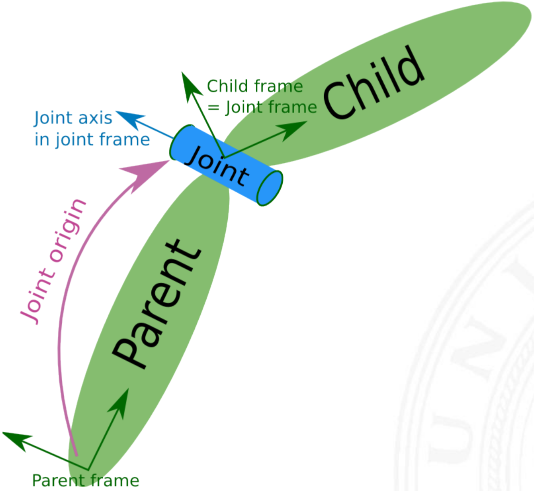


$$d_m = \rho d_v$$

- Products of Inertia:
  $I_{xy} = I_{yx} = \int xydm$ $I_{zy} = I_{yz} = \int yzdm$ $I_{xz} = I_{zx} = \int zydm$

URDF: Joint

Joint describes geometrical connections of two links.



21

URDF: Joint (cont.)

```
<joint name="base_link_to_cyl" type="revolute">
  <!-- describes joint position and orientation -->
  <origin xyz="0 0 0.07" rpy="0 0 0"/>

  <!-- describes the related links -->
  <parent link="base_link"/>
  <child link="base_cyl"/>

  <!-- describes the axis of rotation-->
  <axis xyz="0 0 1"/>

  <!-- describes the joint limits-->
  <limit velocity="1.5707963267"
         lower="-3.1415926535" upper="3.1415926535"/>
</joint>
```

[21] http://wiki.ros.org/urdf/XML/joint

URDF: Joint (cont.)

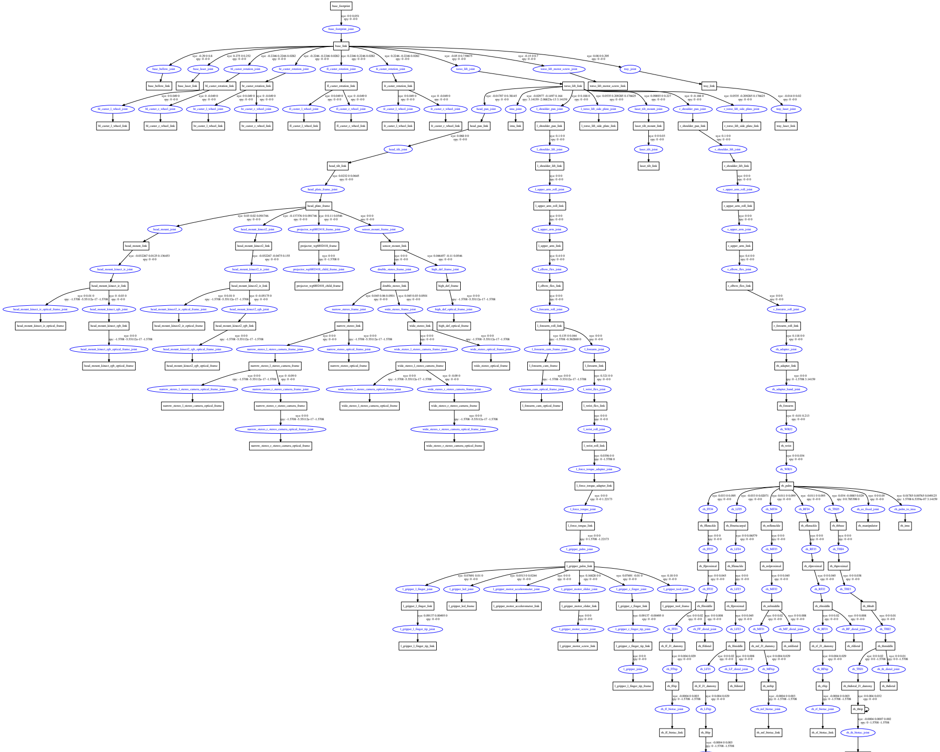
`type`	`revolute`, `continuous`, `prismatic`, `fixed`, `floating`, `planar`
`parent_link`	link which the joint is connected to
`child_link`	link which is connected to the joint
`axis`	joint axis relative to the joint CS. Represented using a normalized vector
`limit`	joint limits for motion (`lower`, `upper`), `velocity` and `effort`
`dynamics`	damping, friction
`calibration`	rising, falling
`mimic`	joint, multiplier, offset
`safety_controller`	soft_lower_limit, soft_upper_limit, k_position, k_velocity

URDF: Other elements

- sensor
  - position and orientation relative to link
  - sensor properties
    - update rate
    - resolution
    - minimum / maximum angle
- transmissions
  - relation of motor to joint motion
- gazebo
  - simulation properties
- model state
  - description of different robot configurations

Complex Hierachy

Full URDF hierarchy of the TAMS PR2 with the Shadow Hand.



Outline

Outline (cont.)

Inverse kinematics for manipulators

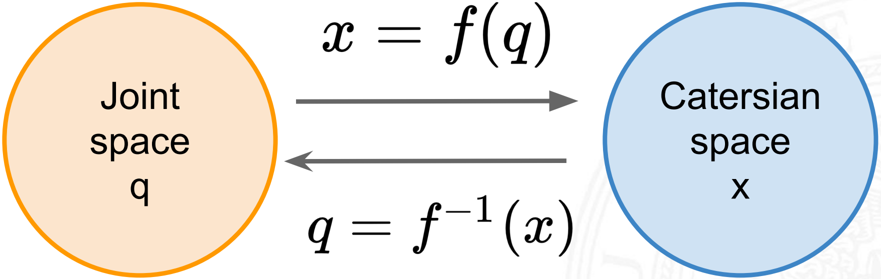
- **Forward Kinematics**: given robot configurations(joint angles), find position & orientations of the end-effector

Set of problems

- In the majority of cases the control of robot manipulators takes place in the *joint space*,
- The informations about objects are mostly given in the *cartesian space*.

Inverse kinematics for manipulators

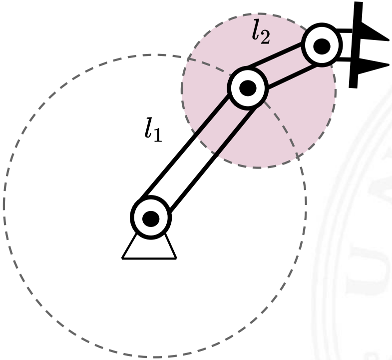
- **Inverse Kinematics**: give position & orientations of the end-effector, find robot configurations(joint angles)



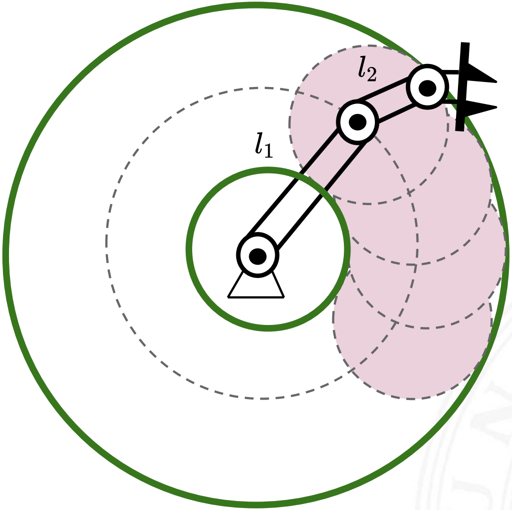
Existence of solutions: Workspace

Workspace: the volume of space that is reachable for the tool of the manipulator.

- reachable workspace
- dexterous workspace

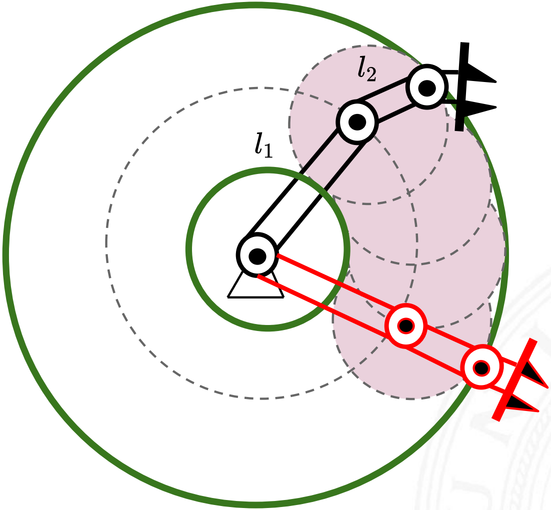


Existence of solutions: Workspace (cont.)



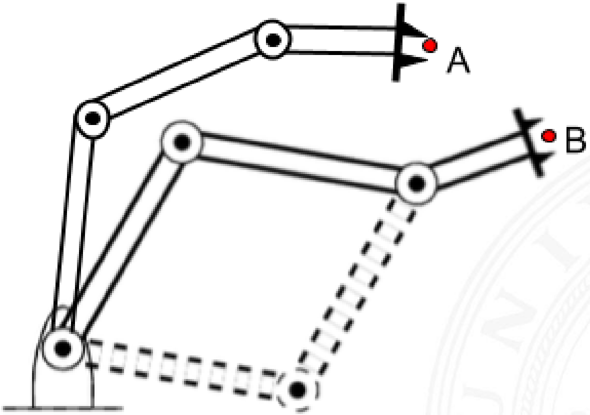
if $l_1 \neq l_2$, the reachable workspace becomes a ring of outer radius $|l_1 + l_2|$,and inner radius $|l_1 - l_2|$.

Existence of solutions: Workspace (cont.)



Does the workspace change if joint limits are considered?
For example, $q_1 \in [0, \pi], q_2 \in [0, \pi]$.

Multiple solutions



The solution using the example of PUMA 560

$$T_6 = T'T'' = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where

$$n_x = C_1[C_{23}(C_4C_5C_6 - S_4S_6) - S_{23}S_5C_6] - S_1(S_4C_5C_6 + C_4S_6) \quad (2)$$

$$n_y = S_1[C_{23}(C_4C_5C_6 - S_4S_6 - S_{23}S_5S_6] + C_1(S_4C_5C_6 + C_4S_6) \quad (3)$$

$$n_z = -S_{23}[C_4C_5C_6 - S_4S_6] - C_{23}S_5C_6 \quad (4)$$

The solution using the example of PUMA 560 (cont.)

$$o_x = \dots \tag{5}$$

$$o_y = \dots \tag{6}$$

$$o_z = \dots \tag{7}$$

$$a_x = \dots \tag{8}$$

$$a_y = \dots \tag{9}$$

$$a_z = \dots \tag{10}$$

$$p_x = C_1[d_6(C_{23}C_4S_5 + S_{23}C_5) + S_{23}d_4 + a_3C_{23} + a_2C_2] - S_1(d_6S_4S_5 + d_2) \tag{11}$$

$$p_y = S_1[d_6(C_{23}C_4S_5 + S_{23}C_5) + S_{23}d_4 + s_3C_{23} + a_2C_2] + C_1(d_6S_4S_5 + d_2) \tag{12}$$

$$p_z = d_6(C_{23}C_5 - S_{23}C_4S_5) + C_{23}d_4 - a_3S_{23} - a_2S_2 \tag{13}$$

Remark

- Non-linear equations
- Existence of solutions
- Multiple solutions
- `Different solution strategy:` closed solutions vs. numerical solutions

Different methods for solution finding

Closed form (analytical):
An expression is said to be a closed-form experession if it can be expressed analytically in terms of a bounded number of certain 'well-known' functions.

- $+ - \times \div$
- nth roots
- exponent and logarithm
- trigonometric and inverse trigonometric functions
- Do not include infinite series, continued fractions, integrals or limits.

Different methods for solution finding (cont.)

Closed form (analytical):

- algebraic solution
  - + accurate solution by means of equations
  - – solution is not geometrically representative
- geometrical solution
  - + case-by-case analysis of possible robot configurations
  - – robot specific

Numerical form:

- iterative methods
  - + the methods are transferable
  - – computationally intensive, for several exceptions the convergence can not be guaranteed

Algebraic solution

Algebraic Approach manipulates the given equations into a form whose solution is known.

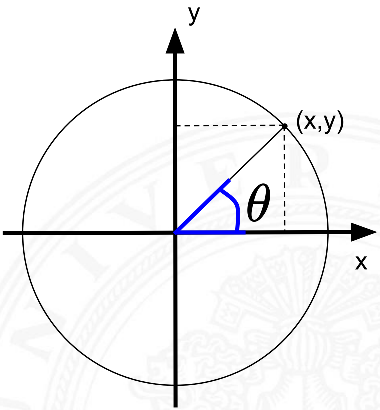
- Method1: **Transcendental equations**

  1. $\sin\theta = a \Rightarrow \theta = A\tan 2(a, \pm\sqrt{1-a^2})$
  2. $\cos\theta = b \Rightarrow \theta = \pm A\tan 2(\sqrt{1-b^2}, b)$
  3. $\left\{\begin{array}{c}\sin\theta = a\\ \cos\theta = b\end{array}\right. \Rightarrow \theta = A\tan 2(a, b)$
  4. $a\cos\theta + b\sin\theta = 0 \Rightarrow \theta = A\tan 2(a, -b)\ or\ \theta = A\tan 2(-a, b)$
  5. $a\cos\theta + b\sin\theta = c \Rightarrow \theta = A\tan 2(b, a) \pm A\tan 2(\sqrt{a^2+b^2-c^2}, c)$
  6. $\left\{\begin{array}{c}a\cos\theta - b\sin\theta = c\\ a\sin\theta + b\cos\theta = d\end{array}\right. \Rightarrow \theta = A\tan 2(ad - ba, ac - bd)$

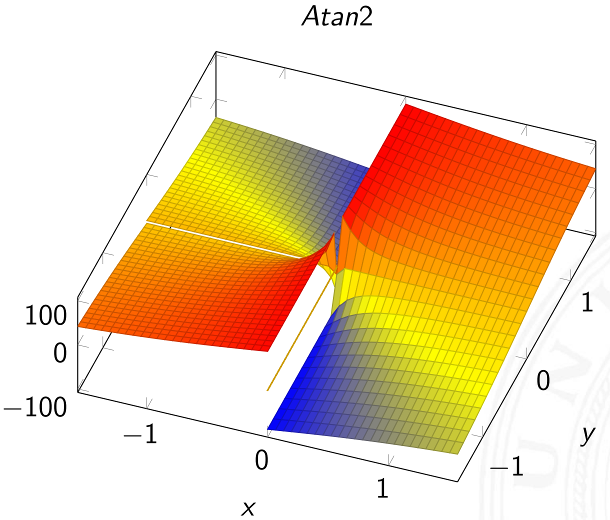
Algebraic solution (cont.)

We define the function *Atan*2 as:

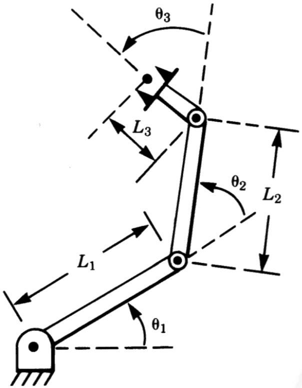
$$\theta = Atan2(y, x) = \begin{cases} Atan(\frac{y}{x}) & \text{for } +x \\ Atan(\frac{y}{x}) + \pi & \text{for } -x, +y_0 \\ Atan(\frac{y}{x}) - \pi & \text{for } -x, -y \\ \frac{\pi}{2} & \text{for } x = 0, +y \\ \frac{-\pi}{2} & \text{for } x = 0, -y \\ NaN & \text{for } x = 0, y = 0 \end{cases}$$



Algebraic solution (cont.)



Example: a planar 3 DOF manipulator



Joint	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	0	l_1	0	θ_2
3	0	l_2	0	θ_3

The algebraical solution for the 3 DOF planar

$$T_6 = {}^0T_3 = \begin{bmatrix} C_{123} & -S_{123} & 0 & l_1 C_1 + l_2 C_{12} \\ S_{123} & C_{123} & 0 & l_1 S_1 + l_2 S_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

with $C_{ij[k]} = \cos(\theta_i + \theta_j[+\theta_k])$

Specification for the TCP: (x, y, ϕ). For such kind of vectors applies:

$${}^0T_3 = \begin{bmatrix} C_\phi & -S_\phi & 0 & x \\ S_\phi & C_\phi & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The algebraical solution for the 3 DOF planar (cont.)

Resultant, four equations can be derived:

$$C_\phi = C_{123} \tag{14}$$
$$S_\phi = S_{123} \tag{15}$$
$$x = l_1 C_1 + l_2 C_{12} \tag{16}$$
$$y = l_1 S_1 + l_2 S_{12} \tag{17}$$

The algebraical solution for the 3 DOF planar (cont.)

Square and add (20) ($x = l_1 C_1 + l_2 C_{12}$) and (21) ($y = l_1 S_1 + l_2 S_{12}$)

$$x^2 + y^2 = l_1^1 + l_2^2 + 2 l_1 l_2 C_2$$

using

$$C_{12} = C_1 C_2 - S_1 S_2, S_{12} = C_1 S_2 + S_1 C_2$$

giving

$$C_2 = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2 l_1 l_2}$$

for goal in workspace

$$S_2 = \pm\sqrt{1 - C_2^2}$$

solution

$$\theta_2 = atan2(S_2, C_2)$$

The algebraical solution for the 3 DOF planar (cont.)

solve (20) $(x = l_1 C_1 + l_2 C_{12})$ and (21) $(y = l_1 S_1 + l_2 S_{12})$ for θ_1

$$\theta_1 = atan2(y, x) - atan2(k_2, k_1)$$

where $k_1 = l_1 + l_2 C_2$ and $k_2 = l_2 S_2$.

solve θ_3 from (19) $(C_\phi = C_{123})$ and (18) $(S_\phi = S_{123})$

$$\theta_1 + \theta_2 + \theta_3 = atan2(S_\phi, C_\phi) = \phi$$

Analytical solvability of manipulator

The closed solution exists if specific constraints (sufficient constraints) for the arm geometry are satisfied:

`If` 3 sequent axes intersect in a given point

`or if` 3 sequent axes are parallel to each other

- manipulators should be designed regarding these constraints
- most of them are
  - PUMA 560: axes 4, 5 & 6 intersect in a single point
  - Mitsubishi PA10, KUKA LWR, PR2
  - 3-DOF planar (RPC)

Algebraical solution (polynomial conversion)

Method2: **Reduction to polynomial**

The following substitutions are used for the polynomial conversion of transcendental equations:

$$u = tan\frac{\theta}{2}$$

$$\cos\theta = \frac{1-u^2}{1+u^2}$$

$$\sin\theta = \frac{2u}{1+u^2}$$

Algebraical solution (polynomial conversion) (cont.)

Example:
The following transcendental equation is given:

$$a\cos\theta + b\sin\theta = c$$

$\Rightarrow \theta = A\tan 2(b, a) \pm A\tan 2(\sqrt{a^2 + b^2 - c^2}, c)$
After the polynomial conversion:

$$a(1 - u^2) + 2bu = c(1 + u^2)$$

The solution for u:

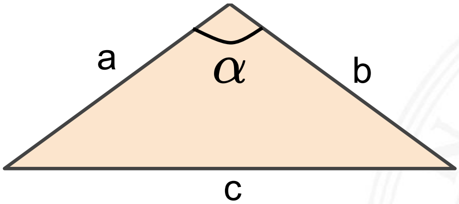
$$u = \frac{b \pm \sqrt{b^2 - a^2 - c^2}}{a + c}$$

Then:

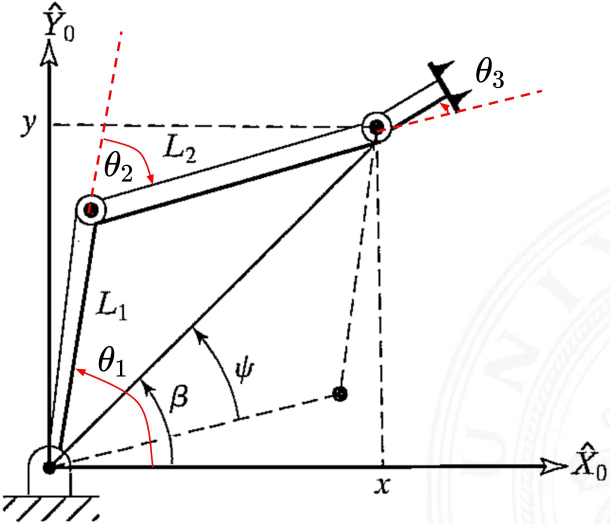
$$\theta = 2tan^{-1}(\frac{b \pm \sqrt{b^2 - a^2 - c^2}}{a + c})$$

Geometrical solution

- Decompose the spatial geometry of the arm into several plane geometry problems
- Law of cosines: $c^2 = a^2 + b^2 - 2ab\cos\alpha$



The geometrical solution for the example 1



The geometrical solution for the example 1 (cont.)

Calculate θ_2 via the `law of cosines`:

$$x^2 + y^2 = l_1^2 + l_2^2 - 2 l_1 l_2 \cos(180 + \theta_2)$$

The solution:

$$\theta_2 = \pm cos^{-1} \frac{x^2 + y^2 - l_1^2 - l_2^2}{2 l_1 l_2}$$

$$\theta_1 = \beta \pm \psi$$

where:

$$\beta = atan2_m(y, x), \quad \cos\psi = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2 l_1 \sqrt{x^2 + y^2}}$$

For $\theta_1, \theta_2, \theta_3$ applies:

$$\theta_1 + \theta_2 + \theta_3 = \phi$$

Exercise

Assume we have derived the forward kinematics as:

$$ {}^{0}T_3 = \begin{bmatrix} C_1 C_{23} & -C_1 S_{23} & S_1 & C_1(C_2 l_2 + l_1) \\ S_1 C_{23} & -S_1 S_{23} & -C_1 & S_1(C_2 l_2 + l_1) \\ S_{23} & C_{23} & 0 & S_2 l_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

And we know:

$$ {}^{0}T_3 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Question: How to solve the inverse kinematics?

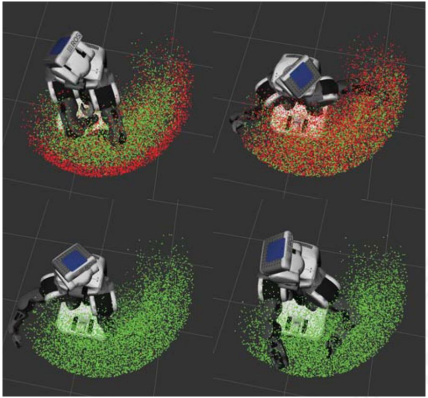
IKFast

- Closed-form solutions
- OpenRAVE
- faster (4 μs) but only work with any number of joints arranged in a chain
- Tutorial: ikfast MoveIt! kinematics_base plugin

TracIK

- TRACLabs' IK solver
- Tutorial: trac_ik MoveIt! kinematics_base plugin
- two IK implementations:
  KDL's Newton-based convergence algorithm
  SQP (Sequential Quadratic Programming) nonlinear optimization approach
- trac_ik_python (RPC)

BioIK 1



KDL

TrackIK

BioIK 2

Download link: bio_ik MoveIt! kinematics_base plugin
22

[22]Ruppel, P., Hendrich, N., Starke, S. and Zhang, J., 2018, May. Cost functions to specify full-body motion and multi-goal manipulation tasks. In 2018 ICRA (pp. 3152-3159). IEEE.

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.