Motion Planning for Robotic Manipulation

Johannes Liebrecht

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

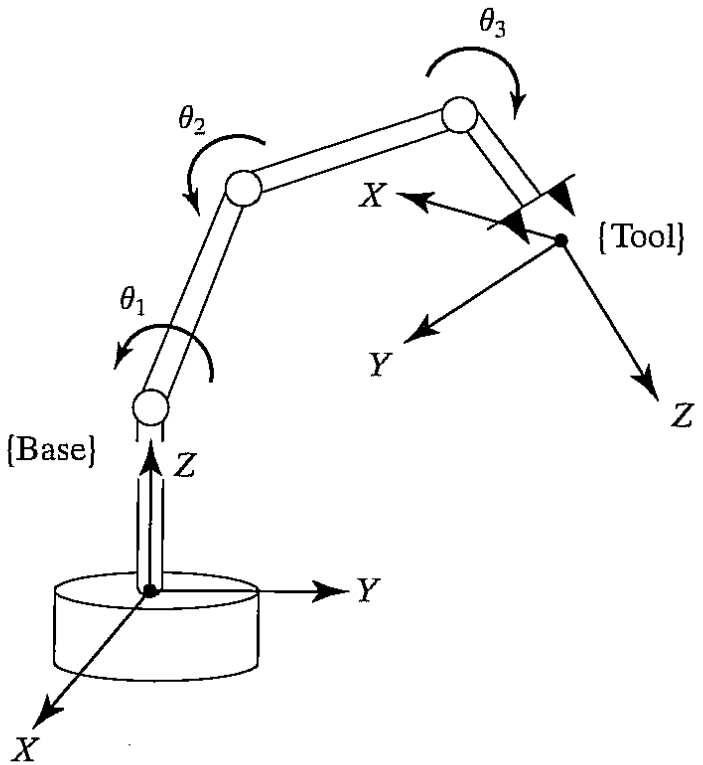
6. January 2014

Outline

Outline

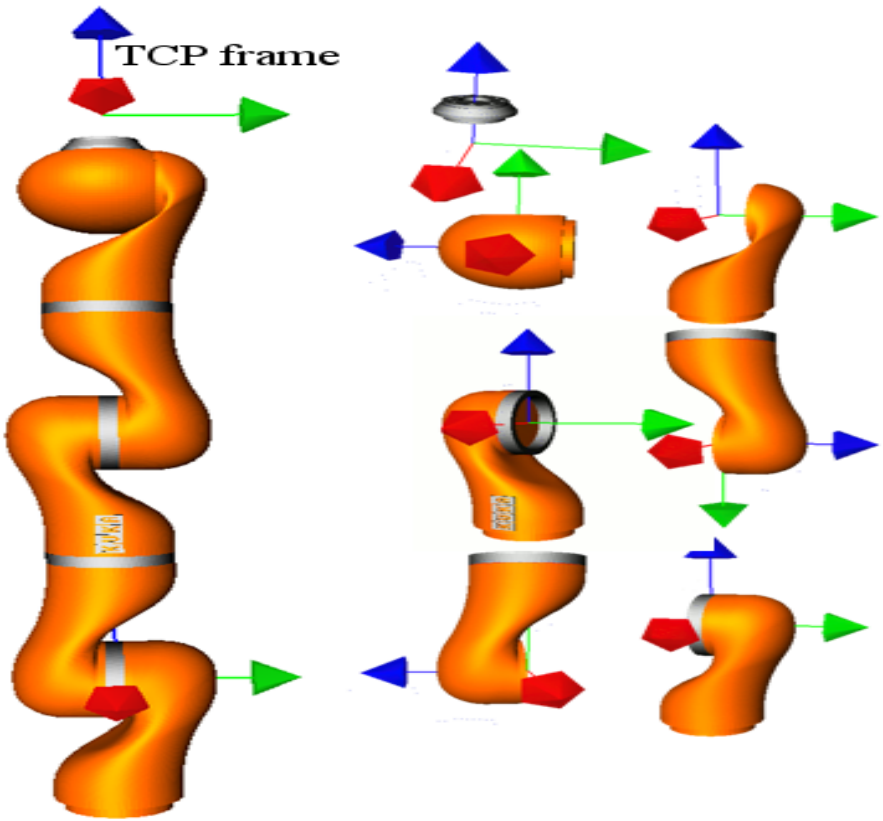
Robotic Manipulator

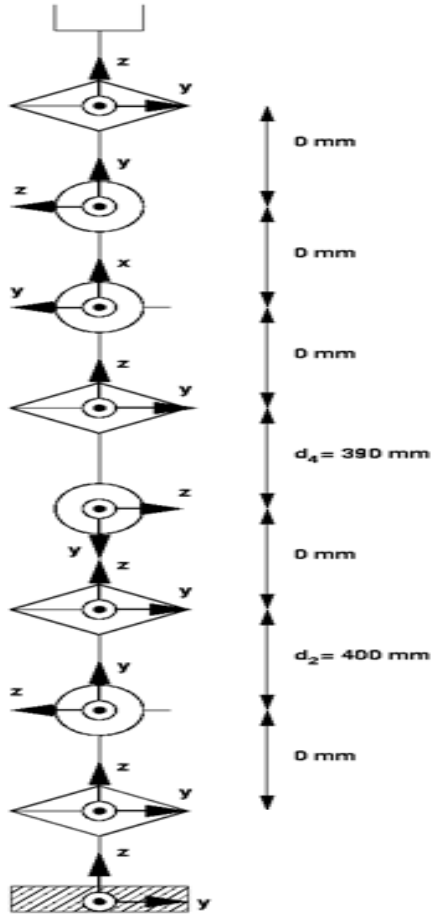
- Degrees of Freedom
- Frames
- Joint/Cartesian space



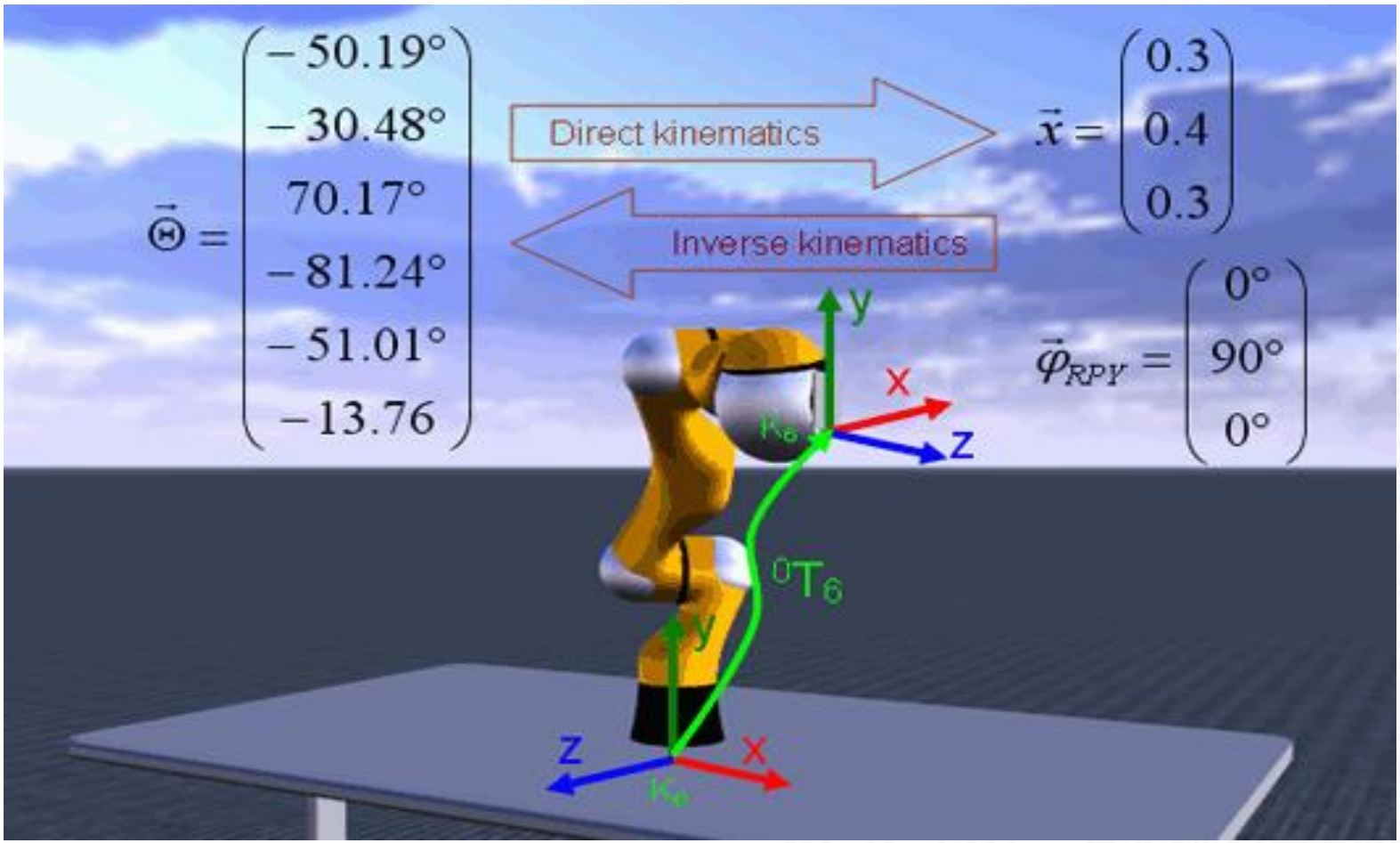
Outline

Kinematic Chain





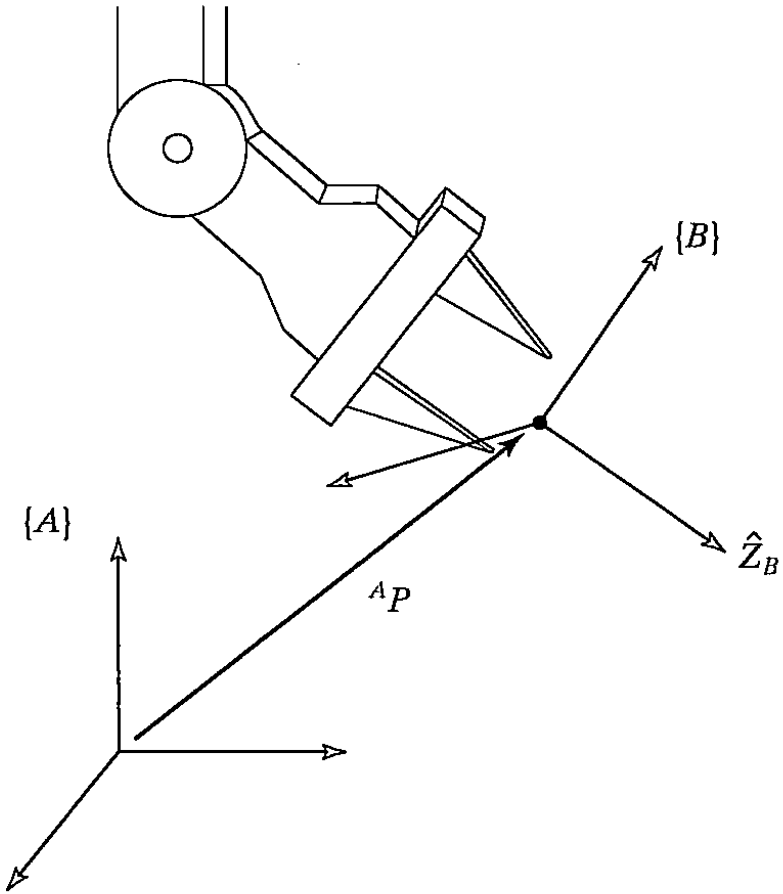
Kinematic Problems



Homogen Transformation

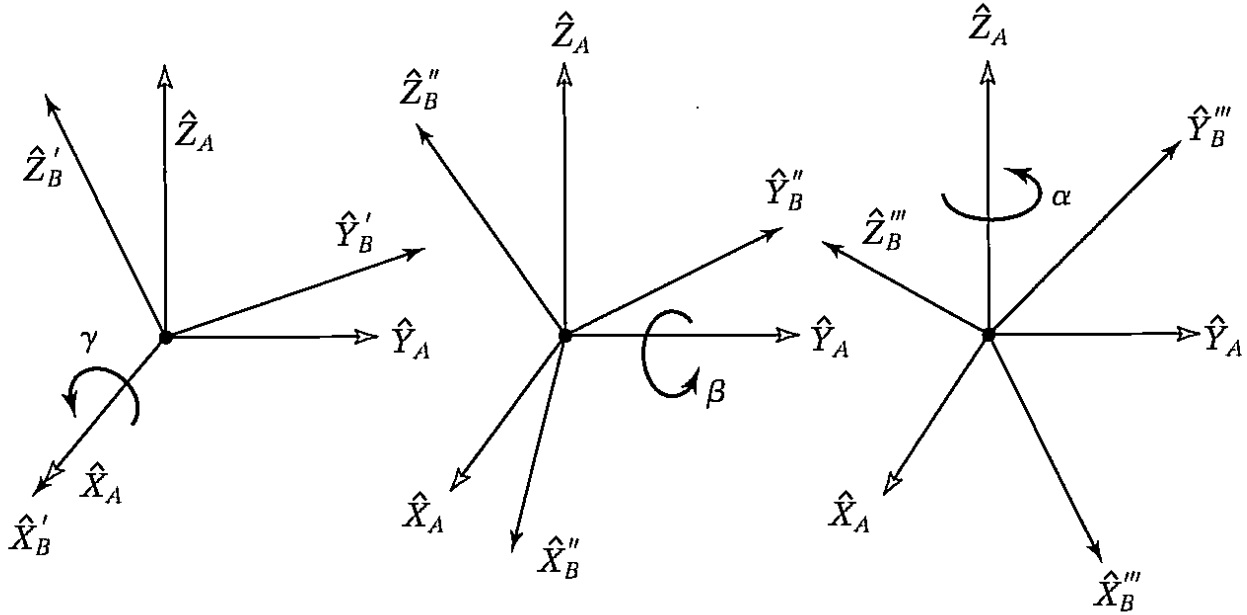
- homogene matrix
  - translation
  - rotation
  - projection
  - scaling

$$\mathrm{H} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



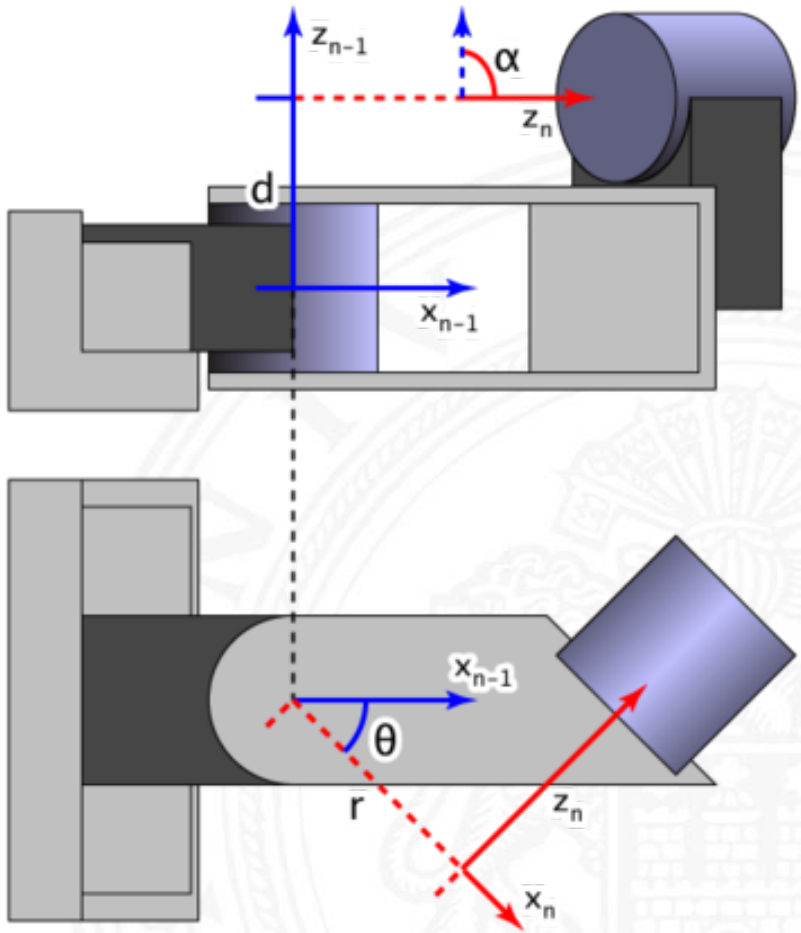
X-Y-Z fixed angles (roll, pitch, yaw angles)

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\gamma) & -sin(\gamma) \\ 0 & sin(\gamma) & cos(\gamma) \end{bmatrix} \begin{bmatrix} cos(\beta) & 0 & sin(\beta) \\ 0 & 1 & 0 \\ -sin(\beta) & 0 & cos(\beta) \end{bmatrix} \begin{bmatrix} cos(\alpha) & -sin(\alpha) & 0 \\ sin(\alpha) & cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



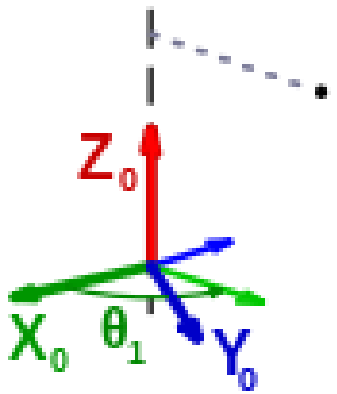
Denavit-Hartenberg convention

- z-axis is in the direction of the joint axis
- x-axis is parallel to the common normal:
  $x_n = z_n \times z_{n-1}$
- y-axis follows by right-handed coordinate system

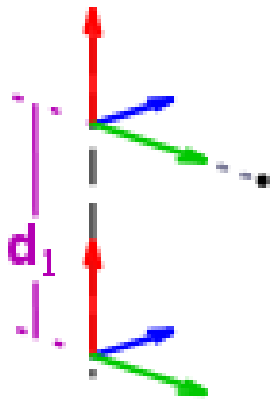


Denavit-Hartenberg parameter $(\theta_n, d_n, a_n, \alpha_n)$

$$R(z_{n-1}, \theta_n) = \begin{bmatrix} cos(\theta_n) & -sin(\theta_n) & 0 & 0 \\ sin(\theta_n) & cos(\theta_n) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

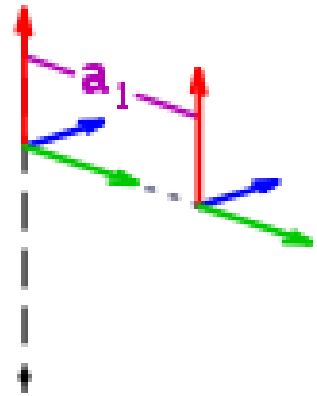


$$T(z_{n-1}, d_n) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

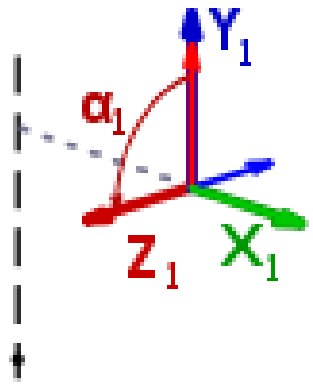


Denavit-Hartenberg parameter$(\theta_n, d_n, a_n, \alpha_n)$

$$T(x_n, a_n) = \begin{bmatrix} 1 & 0 & 0 & a_n \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



$$R(x_n, \alpha_n) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\alpha_n) & -sin(\alpha_n) & 0 \\ 0 & sin(\alpha_n) & cos(\alpha_n) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Denavit-Hartenberg parameter$(\theta_n, d_n, a_n, \alpha_n)$

$$
{}^{n-1}T_n = \begin{bmatrix} cos(\theta_n) & -sin(\theta_n)cos(\alpha_n) & sin(\theta_n)sin(\alpha_n) & a_n cos(\theta_n) \\ sin(\theta_n) & cos(\theta_n) & -cos(\theta_n)sin(\alpha_n) & a_n sin(\theta_n) \\ 0 & sin(\alpha_n) & cos(\alpha_n) & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{n-1}T_n^{-1} = \begin{bmatrix} cos(\theta_n) & sin(\theta_n) & 0 & -a_n \\ -sin(\theta_n)cos(\alpha_n) & cos(\theta_n)cos(\alpha_n) & sin(\alpha_n) & -d_n sin(\alpha_n) \\ sin(\alpha_n)sin(\theta_n) & -cos(\theta_n)sin(\alpha_n) & cos(\alpha_n) & -d_n cos(\alpha_n) \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Example Puma 560

$$ {}_0T^6 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$
\begin{aligned}
r_{11} &= c_1[c_{23}(c_4c_5c_6 - s_4s_5) - s_{23}s_5c_5] + s_1(s_4c_5c_6 + c_4s_6),\\
r_{21} &= s_1[c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6 - c_1(s_4c_5c_6 + c_4s_6),\\
r_{31} &= -s_{23}(c_4c_5c_6 - s_4s_6) - c_{23}s_5c_6,
\end{aligned}
$$

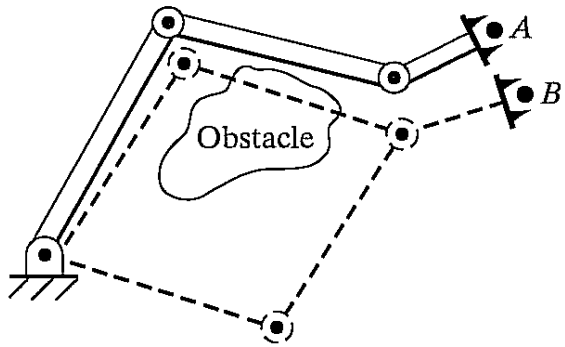
$$
\begin{aligned}
r_{12} &= c_1[c_{23}(-c_4c_5s_6 - s_4c_6) + s_{23}s_5s_6] + s_1(c_4c_6 - s_4c_5s_6),\\
r_{22} &= s_1[c_{23}(-c_4c_5s_6 - s_4c_6) + s_{23}s_5s_6] - c_1(c_4c_6 - s_4c_5s_6),\\
r_{32} &= -s_{23}(-c_4c_5s_6 - s_4c_6) + c_{23}s_5s_6,
\end{aligned}
$$

$$
\begin{aligned}
r_{13} &= -c_1(c_{23}c_4s_5 + s_{23}c_5) - s_1s_4s_5,\\
r_{23} &= -s_1(c_{23}c_4s_5 + s_{23}c_5) + c_1s_4s_5,\\
r_{33} &= s_{23}c_4s_5 - c_{23}c_5,
\end{aligned}
$$

$$
\begin{aligned}
p_x &= c_1[a_2c_2 + a_3c_{23} - d_4s_{23}] - d_3s_1,\\
p_y &= s_1[a_2c_2 + a_3c_{23} - d_4s_{23}] + d_3c_1,\\
p_z &= -a_3s_{23} - a_2s_2 - d_4c_{23}.
\end{aligned}
$$

Inverse Kinematics

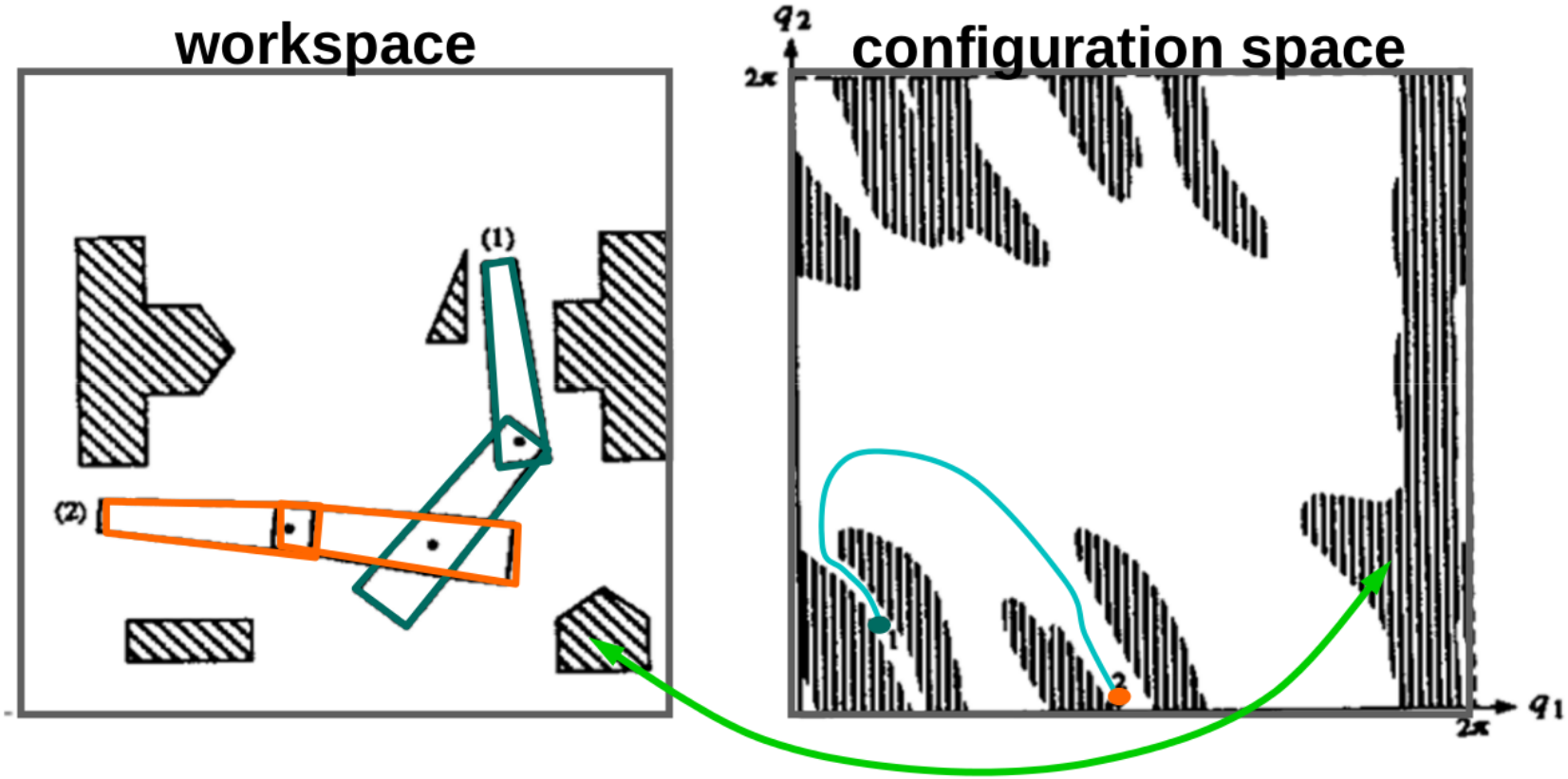
- multiple solutions
- closed-form solution
  - 3 joint axes intersect at a point
  - 3 joint axes are parallel to one another



i	α_{i-1}	a_{i-1}	d_i	θ_i	ll	ul
0	0	0	0	0	-170	170
1	0	90	0	0	-120	120
2	0	-90	0.4	0	-170	170
3	0	-90	0	0	-120	120
4	0	90	0.39	0	-170	170
5	0	90	0	0	-130	130
6	0	-90	0	0	-170	170

Outline

REMINDER: CSPACE



Rapidly Exploring Random Tree(RRT) algorithm

```
BUILD_RRT(q_init)
 1  T.init(q_init);
 2  for k = 1 to K do
 3      q_rand ← RANDOM_CONFIG();
 4      EXTEND(T, q_rand);
 5  Return T
```

Figure 2: The basic RRT construction algorithm.

```
EXTEND(T, q)
 1  q_near ← NEAREST_NEIGHBOR(q, T);
 2  if NEW_CONFIG(q, q_near, q_new) then
 3      T.add_vertex(q_new);
 4      T.add_edge(q_near, q_new);
 5      if q_new = q then
 6          Return Reached;
 7      else
 8          Return Advanced;
 9  Return Trapped;
```

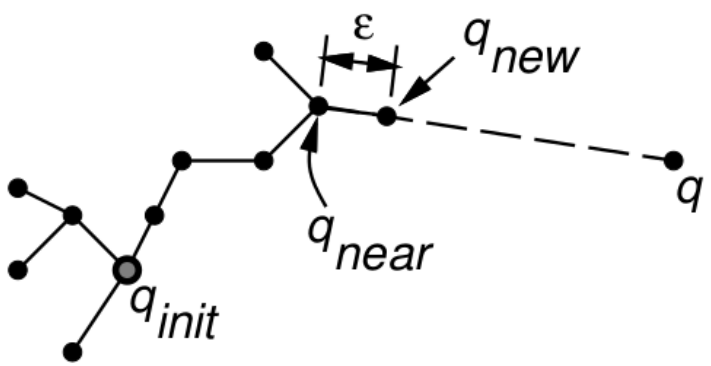


Figure 3: The EXTEND operation.

Why RRTs rapidly explore?

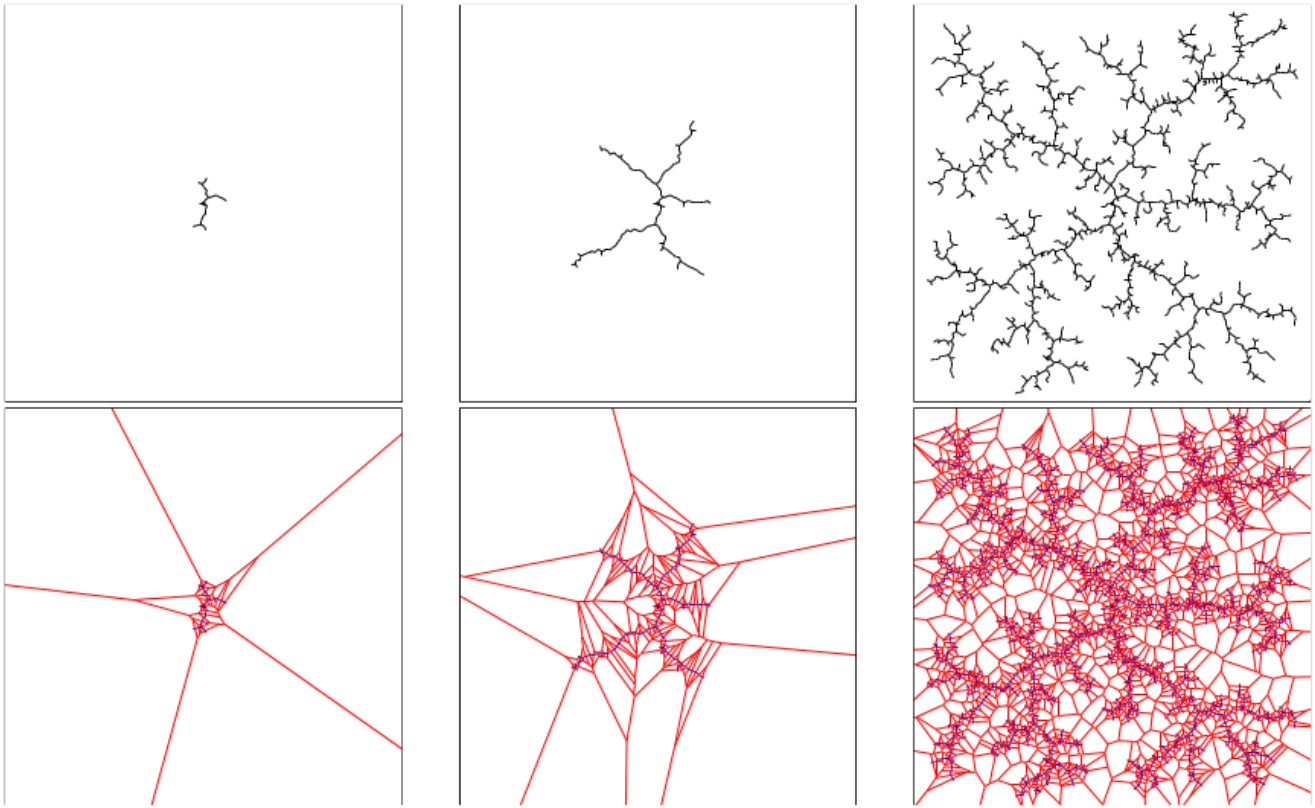


Figure 4: An RRT is biased by large Voronoi regions to rapidly explore, before uniformly covering the space.

RRT-Connect algorithm

```
CONNECT(T, q)
 1  repeat
 2      S ← EXTEND(T, q);
 3  until not (S = Advanced)
 4  Return S;
```

```
RRT_CONNECT_PLANNER(q_init, q_goal)
 1  T_a.init(q_init); T_b.init(q_goal);
 2  for k = 1 to K do
 3      q_rand ← RANDOM_CONFIG();
 4      if not (EXTEND(T_a, q_rand) = Trapped) then
 5          if (CONNECT(T_b, q_new) = Reached) then
 6              Return PATH(T_a, T_b);
 7      SWAP(T_a, T_b);
 8  Return Failure
```

Figure 5: The RRT-Connect algorithm.

Summary

- Kinematic Problem
  - Forward Kinematics
  - Inverse Kinematics
- Motion Planning
  - RRT
  - RRT-Connect

References

- John J. Craig, Introduction to Robotics: Mechanics and Control., 2005 Pearson Education, Inc.
- James J. Kuffner & Steven M. LaValle, RRT-Connect: An Efficient Approach to Single-Query Path Planning, 2000 IEEE International Conference on Robotics and Automation
- www.wikipedia.com, Denavit–Hartenberg parameters

Thank you for your attention!