Introduction to Robotics for cognitive science

Dr. Andrej Lúčny

KAI FMFI UK

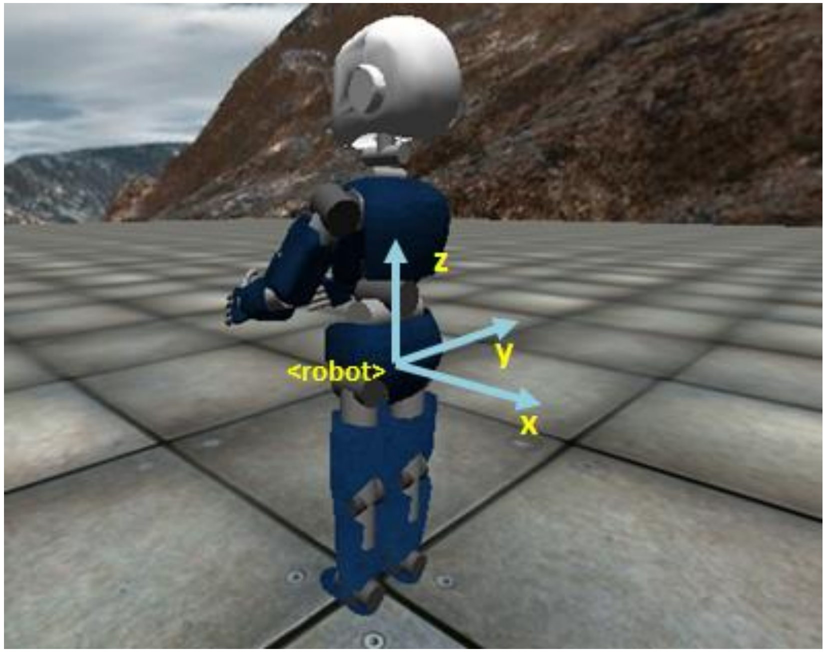
lucny@fmph.uniba.sk

www.agentspace.org/kv

2

Direct kinematics

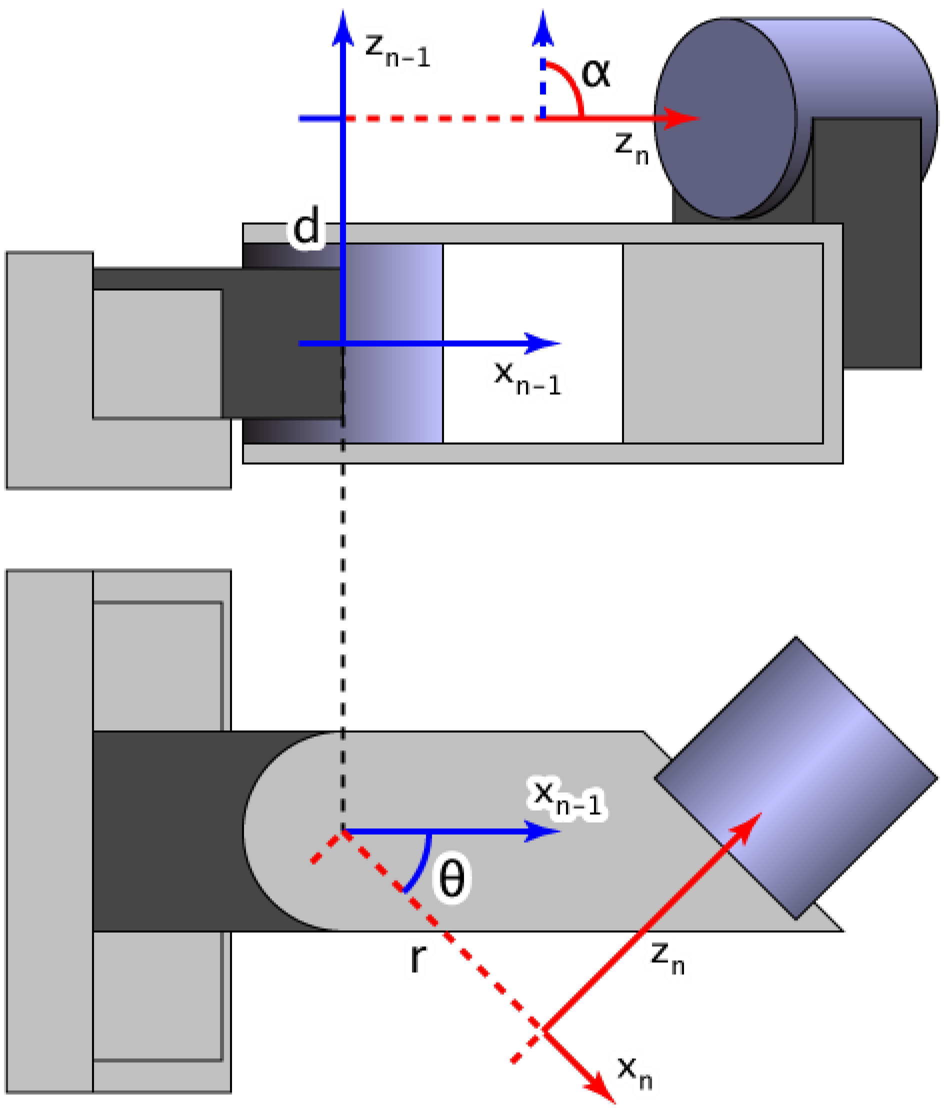
- Calculation of the robot position from the motors positions (coordinates from angles)



Robot body

- For the calculation we need data about robot body
- Typically this data are expressed in form of Denavit-Hartenberg (D-H) convention which specify for each joint four parameters:
  - r, d, θ, α

(constants)



Math: projective geometry

- We need to calculate translations and rotations
- We can represent rotations in 2D by matrix 2x2

$$[R] = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}$$

$$\begin{bmatrix} x_{rot} \\ y_{rot} \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

in 3D by 3x3

$$[R] = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\alpha & -sin\alpha \\ 0 & sin\alpha & cos\alpha \end{bmatrix}$$

$$\begin{bmatrix} x_{rot} \\ y_{rot} \\ z_{rot} \end{bmatrix} = [R] \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

- We can compose these matrices by multiplication:

$$[R] = [R_1][R_2]\ldots[R_N]$$

Math: projective geometry

- The representation of translation is much more tricky
- It is not possible to multiply a 3D vector *(x,y,z)* by a 3x3 matrix to get $(x+t_x, y+t_y, z+t_z)$
- Therefore we extend coordinates to (x,y,z,1)
- The translation is represented by 4x4 matrix

$$[T] = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} x+t_x \\ y+t_y \\ z+t_z \\ 1 \end{bmatrix} = [T] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- (we extend R to 4x4 with zeros)

- All together *D-H* parameters of a joints can be represented by a product of two rotational and two transactional matrices

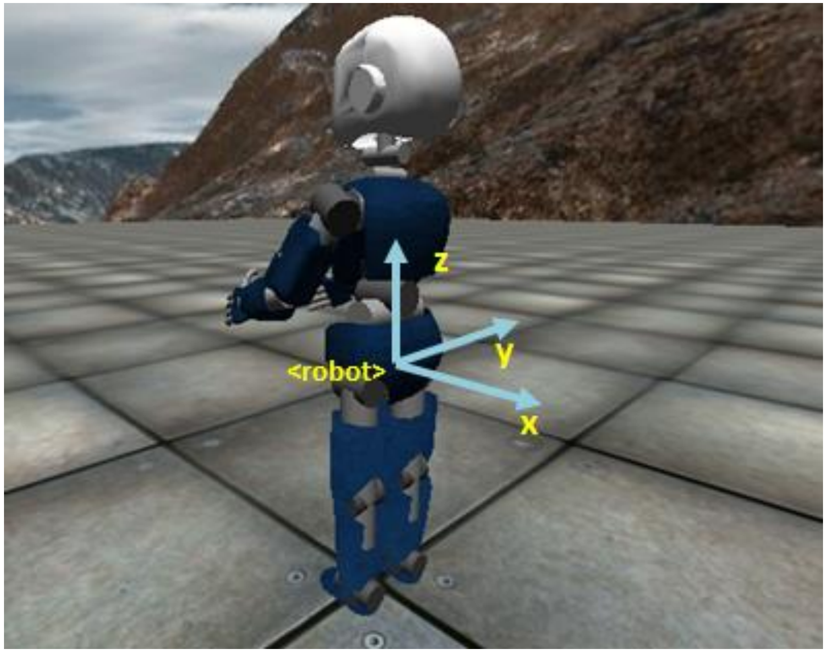
$$[DH] = \begin{bmatrix} \cos\theta & -\sin\theta\cos\alpha & \sin\theta\sin\alpha & \cos\theta r \\ \sin\theta & \cos\theta\cos\alpha & -\cos\theta\sin\alpha & \sin\theta r \\ 0 & \sin\alpha & \cos\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- The we calculate positions in 3D by

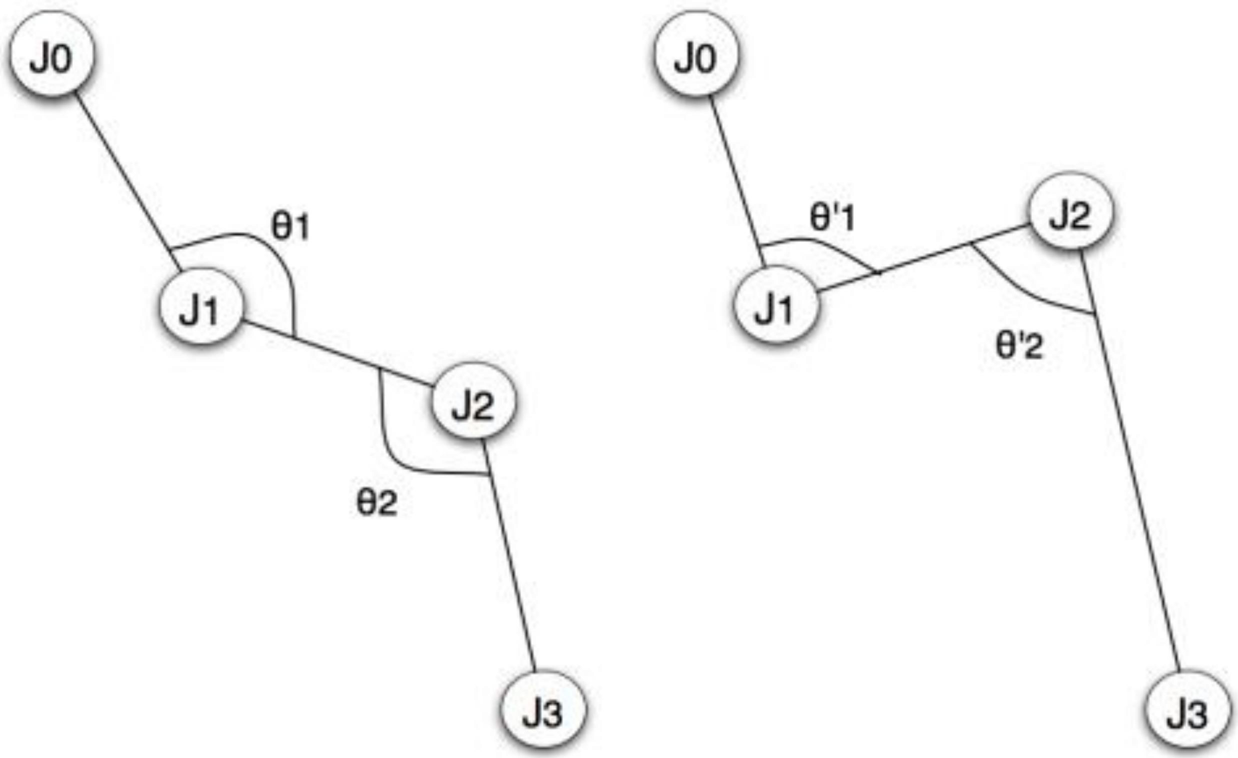
$$[DH_1][DH_2]\ldots[DH_N]\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Inverse kinematics

- Calculation of the motors positions from the robot positions (angles from coordinates)



- Inverse kinematics can have more solutions or no solution



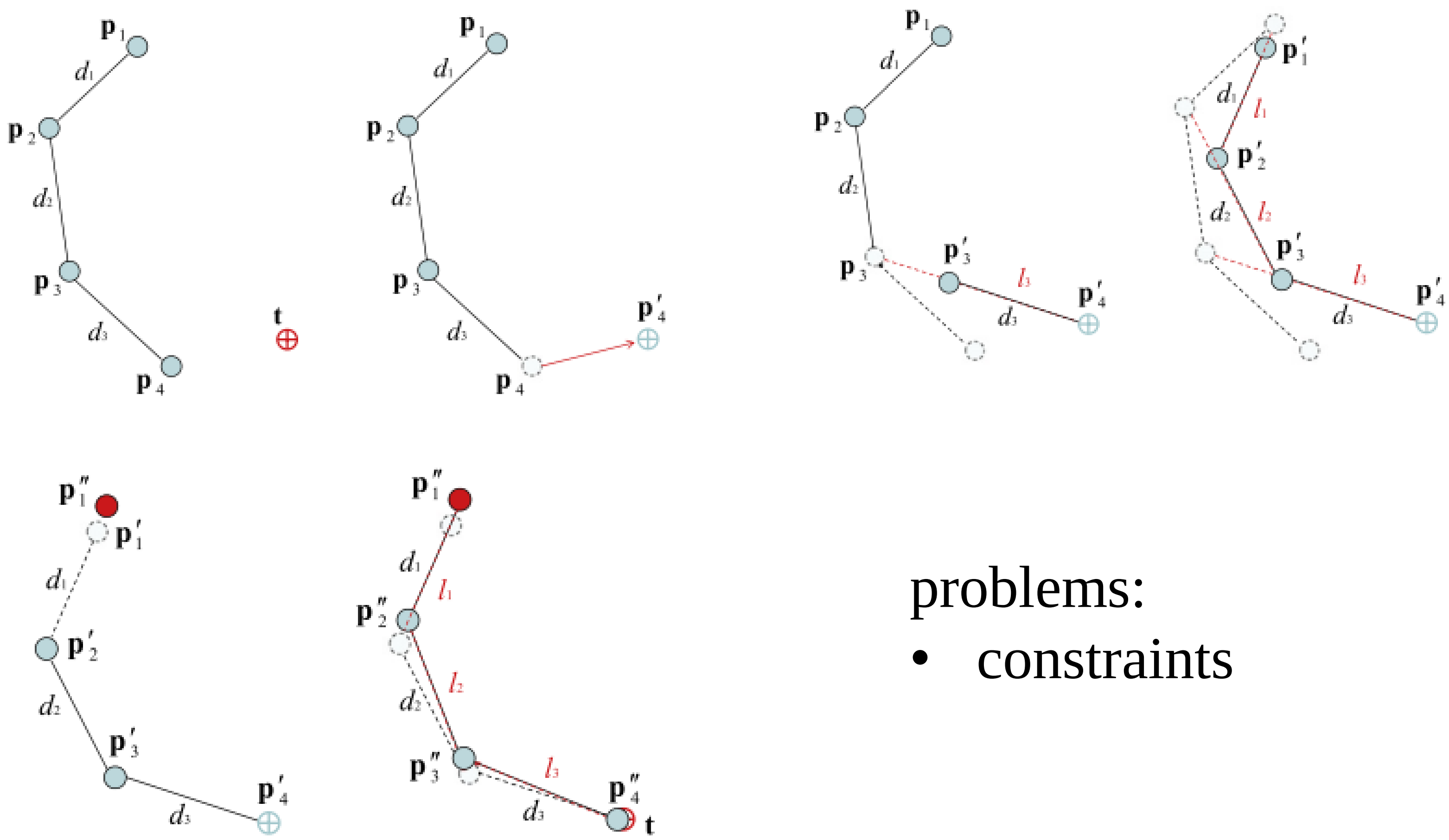
- Solution of inverse kinematics is more complicated. In analytic form it can be calculated only for specific cases.
- However we have heuristic iterative methods which can find solution, e.g. FARBRIK algorithm

FABRIK

forward and backward reaching inverse kinematics

forward

backward



problems:

- constraints