Robotics: Denavit–Hartenberg notation

Vladimír Petrík

vladimir.petrik@cvut.cz

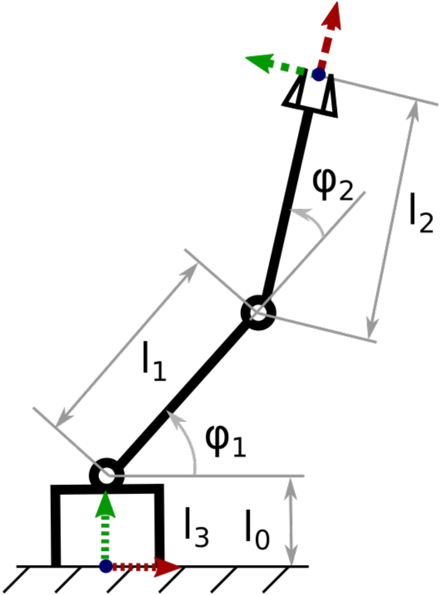
20.10.2023

Denavit–Hartenberg notation

- Method for assigning frames to links in kinematic chains
- Introduced by Jacques Denavit and Richard Hartenberg in 1955
- Minimal representation
- Sometimes used in robotics

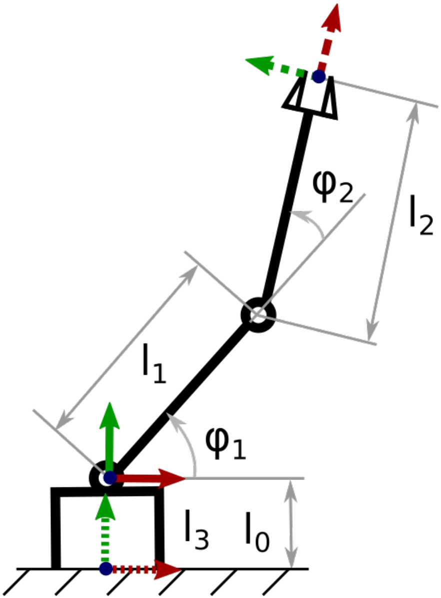
Motivation

- Consider FK for a planar 2-DoF manipulator
  $\varphi_1, \varphi_2 \rightarrow T \in SE(2)$



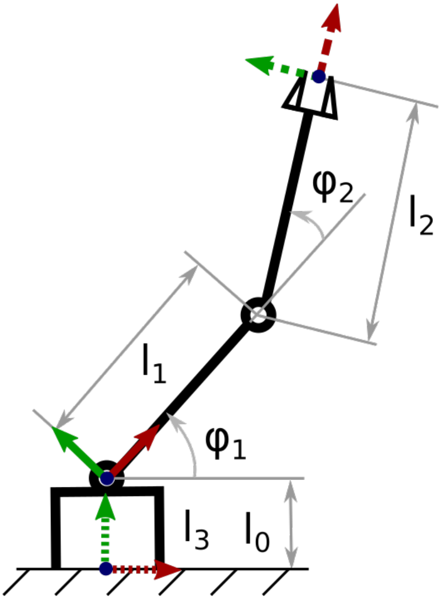
Motivation

- Consider FK for a planar 2-DoF manipulator $\varphi_1, \varphi_2 \to T \in SE(2)$
- $T_1 = T_y(l_0)$



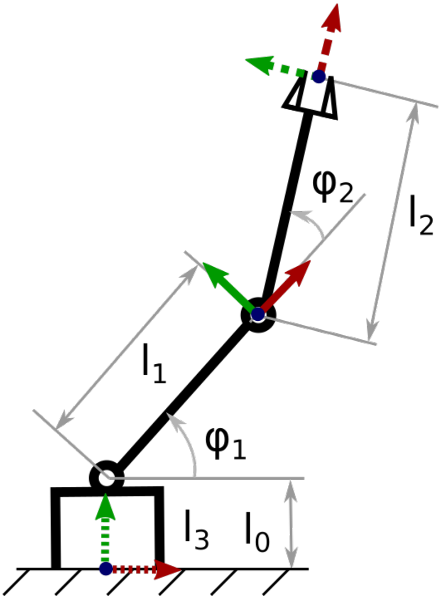
Motivation

- Consider FK for a planar 2-DoF manipulator
  $\varphi_1, \varphi_2 \to T \in SE(2)$
- $T_1 = T_y(l_0)$
- $T_2 = R(\varphi_1)$



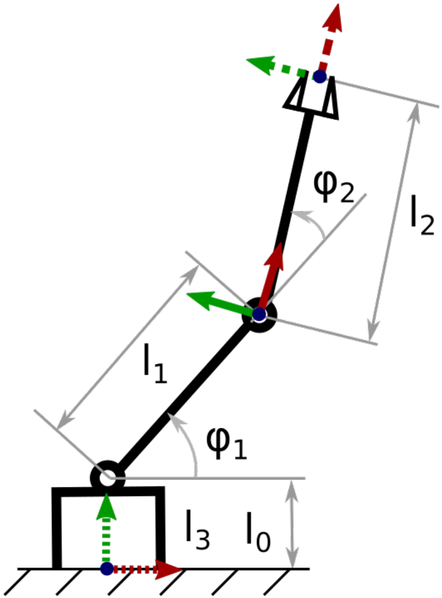
Motivation

- Consider FK for a planar 2-DoF manipulator
  $\varphi_1, \varphi_2 \rightarrow T \in SE(2)$
- $T_1 = T_y(l_0)$
- $T_2 = R(\varphi_1)T_x(l_1)$



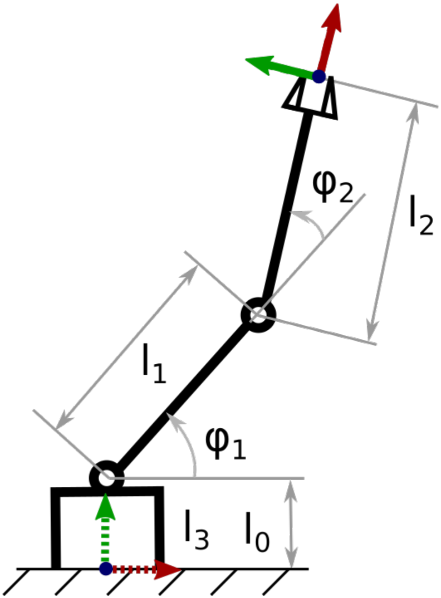
Motivation

- Consider FK for a planar 2-DoF manipulator $\varphi_1, \varphi_2 \rightarrow T \in SE(2)$
- $T_1 = T_y(l_0)$
- $T_2 = R(\varphi_1)T_x(l_1)$
- $T_3 = R(\varphi_2)$



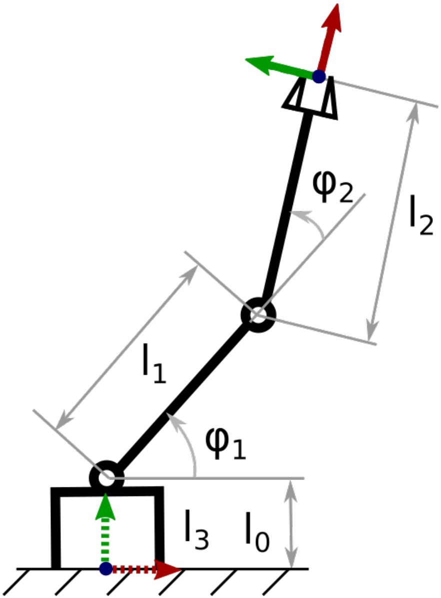
Motivation

- Consider FK for a planar 2-DoF manipulator $\varphi_1, \varphi_2 \rightarrow T \in SE(2)$
- $T_1 = T_y(l_0)$
- $T_2 = R(\varphi_1)T_x(l_1)$
- $T_3 = R(\varphi_2)T_x(l_2)$



Motivation

- Consider FK for a planar 2-DoF manipulator
  $\varphi_1, \varphi_2 \rightarrow T \in SE(2)$
- $T_1 = T_y(l_0)$
- $T_2 = R(\varphi_1)T_x(l_1)$
- $T_3 = R(\varphi_2)T_x(l_2)$
- $T = T_1 T_2 T_3$



Motivation

- Consider FK for a planar 2-DoF manipulator
  $\varphi_1, \varphi_2 \to T \in SE(2)$
- $T_1 = T_y(l_0)$
- $T_2 = R(\varphi_1)T_x(l_1) \leftarrow$ **structure**
- $T_3 = R(\varphi_2)T_x(l_2)$
- $T = T_1 T_2 T_3$

Denavit–Hartenberg parameters

- Similar structure but for spatial manipulators

Denavit–Hartenberg parameters

- Similar structure but for spatial manipulators
- Four parameters for each transformation
  - $T_x(a)$, $T_z(d)$, $R_x(\alpha)$, $R_z(\theta)$
  - $T_{\mathsf{DH}} = R_z(\theta)T_z(d)R_x(\alpha)T_x(a)$

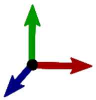
Denavit–Hartenberg parameters

- Similar structure but for spatial manipulators
- Four parameters for each transformation
  - $T_x(a)$, $T_z(d)$, $R_x(\alpha)$, $R_z(\theta)$
  - $T_{\mathsf{DH}} = R_z(\theta)T_z(d)R_x(\alpha)T_x(a)$
- Which of the following equals to T_{DH}?
  1. $T_{\mathsf{DH}} = T_z(d)R_z(\theta)T_x(a)R_x(\alpha)$
  2. $T_{\mathsf{DH}} = R_x(\alpha)T_x(a)R_z(\theta)T_z(d)$

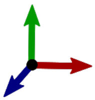
Denavit–Hartenberg parameters

- Similar structure but for spatial manipulators
- Four parameters for each transformation
  - $T_x(a)$, $T_z(d)$, $R_x(\alpha)$, $R_z(\theta)$
  - $T_{\text{DH}} = R_z(\theta)T_z(d)R_x(\alpha)T_x(a)$
- Which of the following equals to T_{DH}?
  1. $T_{\text{DH}} = T_z(d)R_z(\theta)T_x(a)R_x(\alpha)$
     yes, $T_xR_x = R_xT_x$
  2. $T_{\text{DH}} = R_x(\alpha)T_x(a)R_z(\theta)T_z(d)$

T_yR_y



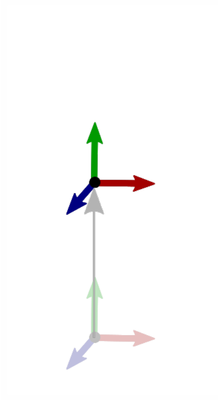
R_yT_y



Denavit–Hartenberg parameters

- Similar structure but for spatial manipulators
- Four parameters for each transformation
  - $T_x(a)$, $T_z(d)$, $R_x(\alpha)$, $R_z(\theta)$
  - $T_{\mathrm{DH}} = R_z(\theta)T_z(d)R_x(\alpha)T_x(a)$
- Which of the following equals to T_{DH}?
  1. $T_{\mathrm{DH}} = T_z(d)R_z(\theta)T_x(a)R_x(\alpha)$
     yes, $T_xR_x = R_xT_x$
  2. $T_{\mathrm{DH}} = R_x(\alpha)T_x(a)R_z(\theta)T_z(d)$

T_yR_y

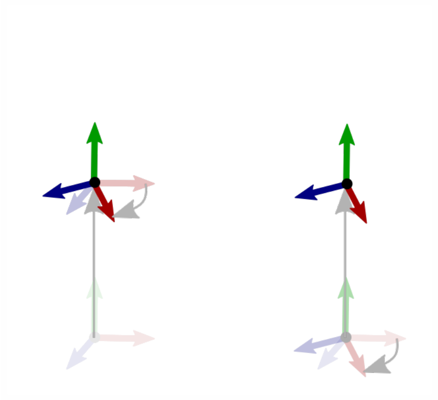


R_yT_y

Denavit–Hartenberg parameters

- Similar structure but for spatial manipulators
- Four parameters for each transformation
  - $T_x(a)$, $T_z(d)$, $R_x(\alpha)$, $R_z(\theta)$
  - $T_{\text{DH}} = R_z(\theta)T_z(d)R_x(\alpha)T_x(a)$
- Which of the following equals to T_{DH}?
  1. $T_{\text{DH}} = T_z(d)R_z(\theta)T_x(a)R_x(\alpha)$
     yes, $T_xR_x = R_xT_x$
  2. $T_{\text{DH}} = R_x(\alpha)T_x(a)R_z(\theta)T_z(d)$

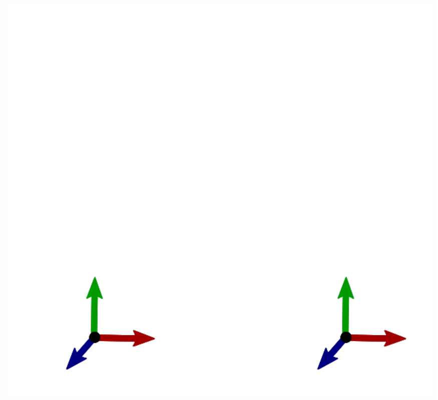
T_yR_y R_yT_y



Denavit–Hartenberg parameters

- Similar structure but for spatial manipulators
- Four parameters for each transformation
  - $T_x(a)$, $T_z(d)$, $R_x(\alpha)$, $R_z(\theta)$
  - $T_{\text{DH}} = R_z(\theta)T_z(d)R_x(\alpha)T_x(a)$
- Which of the following equals to T_{DH}?
  1. $T_{\text{DH}} = T_z(d)R_z(\theta)T_x(a)R_x(\alpha)$
     yes, $T_x R_x = R_x T_x$
  2. $T_{\text{DH}} = R_x(\alpha)T_x(a)R_z(\theta)T_z(d)$

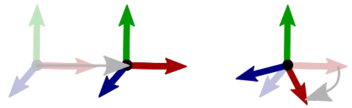
$T_x R_y$ $\qquad$ $R_y T_x$



Denavit–Hartenberg parameters

- Similar structure but for spatial manipulators
- Four parameters for each transformation
  - $T_x(a)$, $T_z(d)$, $R_x(\alpha)$, $R_z(\theta)$
  - $T_{\mathsf{DH}} = R_z(\theta)T_z(d)R_x(\alpha)T_x(a)$
- Which of the following equals to T_{DH}?
  1. $T_{\mathsf{DH}} = T_z(d)R_z(\theta)T_x(a)R_x(\alpha)$
     yes, $T_xR_x = R_xT_x$
  2. $T_{\mathsf{DH}} = R_x(\alpha)T_x(a)R_z(\theta)T_z(d)$

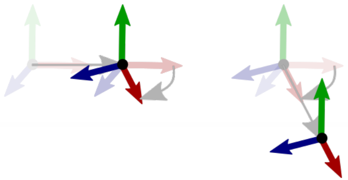
$$T_xR_y \qquad R_yT_x$$



Denavit–Hartenberg parameters

- Similar structure but for spatial manipulators
- Four parameters for each transformation
  - $T_x(a)$, $T_z(d)$, $R_x(\alpha)$, $R_z(\theta)$
  - $T_{\mathrm{DH}} = R_z(\theta)T_z(d)R_x(\alpha)T_x(a)$
- Which of the following equals to T_{DH}?
  1. $T_{\mathrm{DH}} = T_z(d)R_z(\theta)T_x(a)R_x(\alpha)$
     yes, $T_xR_x = R_xT_x$
  2. $T_{\mathrm{DH}} = R_x(\alpha)T_x(a)R_z(\theta)T_z(d)$

$$T_xR_y \qquad R_yT_x$$



Denavit–Hartenberg parameters

- Similar structure but for spatial manipulators
- Four parameters for each transformation
  - $T_x(a)$, $T_z(d)$, $R_x(\alpha)$, $R_z(\theta)$
  - $T_{\mathsf{DH}} = R_z(\theta)T_z(d)R_x(\alpha)T_x(a)$
- Which of the following equals to T_{DH}?
  1. $T_{\mathsf{DH}} = T_z(d)R_z(\theta)T_x(a)R_x(\alpha)$
     yes, $T_x R_x = R_x T_x$
  2. $T_{\mathsf{DH}} = R_x(\alpha)T_x(a)R_z(\theta)T_z(d)$
- Can we create arbitrary $SE(3)$ transformation with DH

Denavit–Hartenberg parameters

- Similar structure but for spatial manipulators
- Four parameters for each transformation
  - $T_x(a)$, $T_z(d)$, $R_x(\alpha)$, $R_z(\theta)$
  - $T_{\mathsf{DH}} = R_z(\theta)T_z(d)R_x(\alpha)T_x(a)$
- Which of the following equals to T_{DH}?
  1. $T_{\mathsf{DH}} = T_z(d)R_z(\theta)T_x(a)R_x(\alpha)$
     yes, $T_x R_x = R_x T_x$
  2. $T_{\mathsf{DH}} = R_x(\alpha)T_x(a)R_z(\theta)T_z(d)$
- Can we create arbitrary $SE(3)$ transformation with DH
  - No, only 4 DoF

Denavit–Hartenberg parameters

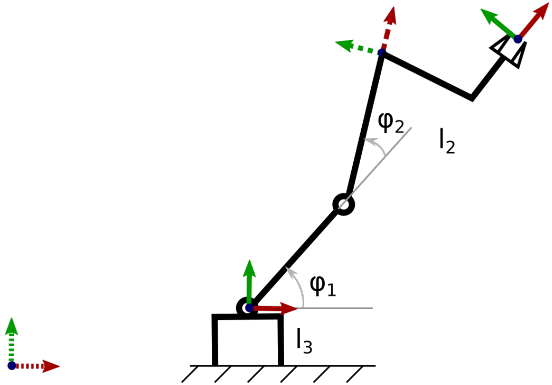
- Similar structure but for spatial manipulators
- Four parameters for each transformation
  - $T_x(a)$, $T_z(d)$, $R_x(\alpha)$, $R_z(\theta)$
  - $T_{\mathsf{DH}} = R_z(\theta)T_z(d)R_x(\alpha)T_x(a)$
- Which of the following equals to T_{DH}?
  1. $T_{\mathsf{DH}} = T_z(d)R_z(\theta)T_x(a)R_x(\alpha)$
     yes, $T_xR_x = R_xT_x$
  2. $T_{\mathsf{DH}} = R_x(\alpha)T_x(a)R_z(\theta)T_z(d)$
- Can we create arbitrary $SE(3)$ transformation with DH
  - No, only 4 DoF
  - Designed for open kinematic chains with revolute and prismatic joints

Denavit–Hartenberg parameters

- Similar structure but for spatial manipulators
- Four parameters for each transformation
  - $T_x(a)$, $T_z(d)$, $R_x(\alpha)$, $R_z(\theta)$
  - $T_{\mathsf{DH}} = R_z(\theta)T_z(d)R_x(\alpha)T_x(a)$
- Which of the following equals to T_{DH}?
  1. $T_{\mathsf{DH}} = T_z(d)R_z(\theta)T_x(a)R_x(\alpha)$
     yes, $T_xR_x = R_xT_x$
  2. $T_{\mathsf{DH}} = R_x(\alpha)T_x(a)R_z(\theta)T_z(d)$
- Can we create arbitrary $SE(3)$ transformation with DH
  - No, only 4 DoF
  - Designed for open kinematic chains with revolute and prismatic joints
- Coordinate frames need to be placed appropriately
  - z-axis is in axis of rotation/translation
  - x_1 is perpendicular to z_0 and z_1
  - x_1 intersects z_0 and z_1

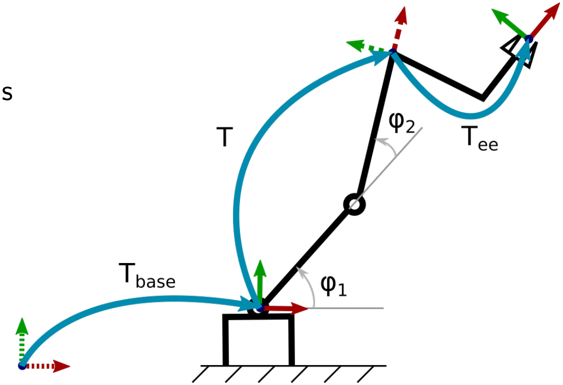
Initial and final transforms

- We cannot create arbitrary $SE(3)$ transformation with DH
  - Mount gripper on different location
  - Defining different reference frame



Initial and final transforms

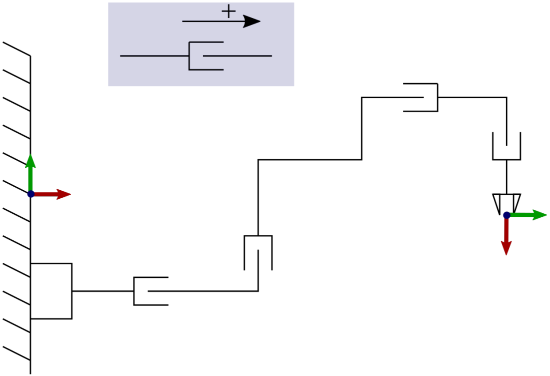
- We cannot create arbitrary $SE(3)$ transformation with DH
  - Mount gripper on different location
  - Defining different reference frame
- Usually we define initial and final transforms
  - $T = T^1_{\mathsf{DH}} T^2_{\mathsf{DH}} \dots T^n_{\mathsf{DH}}$
  - $T_{\mathsf{FK}} = T_{\mathsf{base}} T T_{\mathsf{ee}}$



Example: spatial robot in plane

xyz=rgb

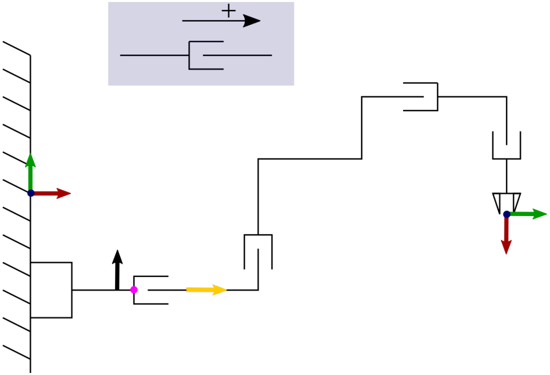
- Four prismatic joints
- Solve FK with DH notation



Example: spatial robot in plane

xyz=rgb

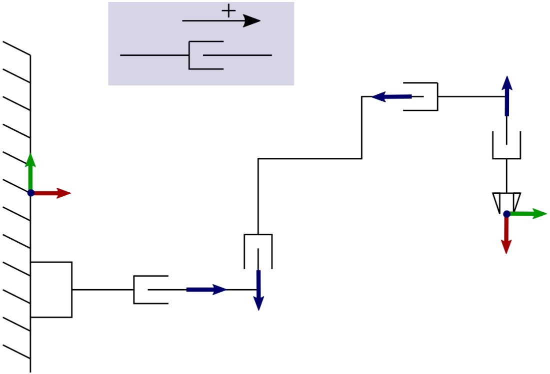
- z -axis is in axis of rotation/translation
- Where will be z -axis?
  1. black
  2. yellow
  3. pink



Example: spatial robot in plane

xyz=rgb

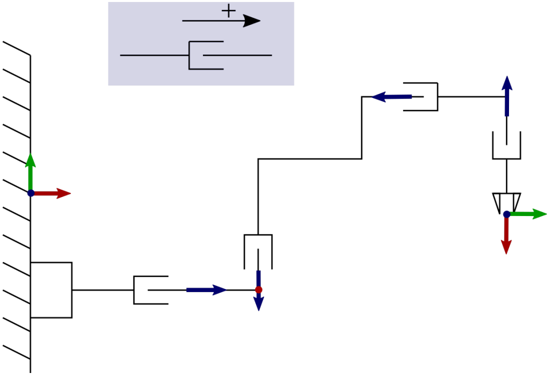
- Be careful with orientation



Example: spatial robot in plane

xyz=rgb

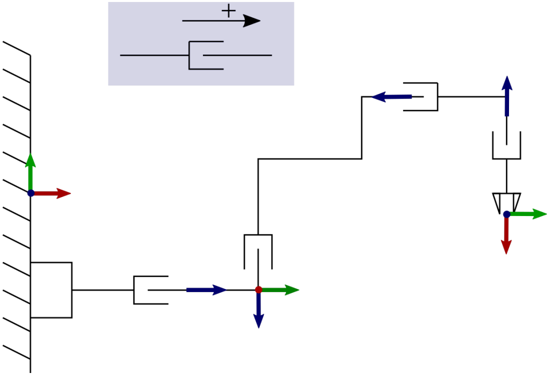
- We know:
  - x_1 is perpendicular to z_0 and z_1
  - x_1 intersects z_0 and z_1
- x -axis of the first frame:
  1. axis points out of the screen
  2. axis points into the screen
  3. both in/out is correct



Example: spatial robot in plane

xyz=rgb

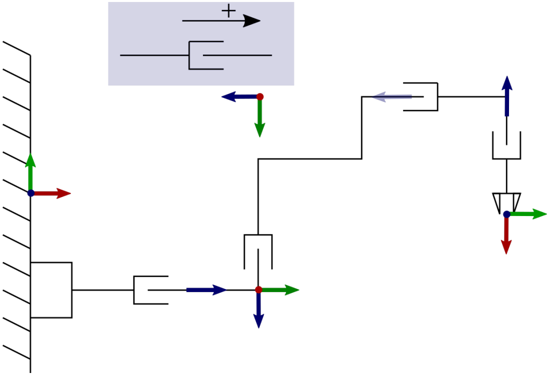
- We know x and z,
  we can determine origin and y



Example: spatial robot in plane

xyz=rgb

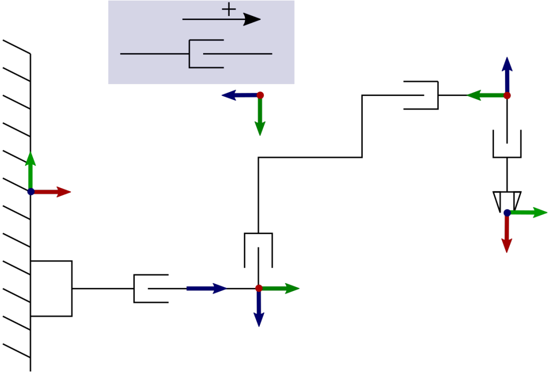
- Some frames could be located 'outside' the robot



Example: spatial robot in plane

xyz=rgb

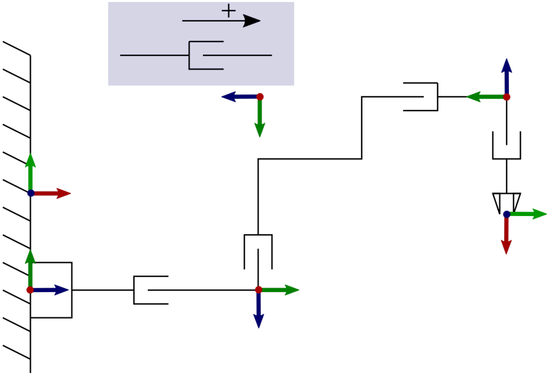
- Only the last frame is missing



Example: spatial robot in plane

xyz=rgb

- We have 6 frames
  - Initial transformation
  - 4 DH transformations
  - Final transformation
- It remains to determine DH parameters



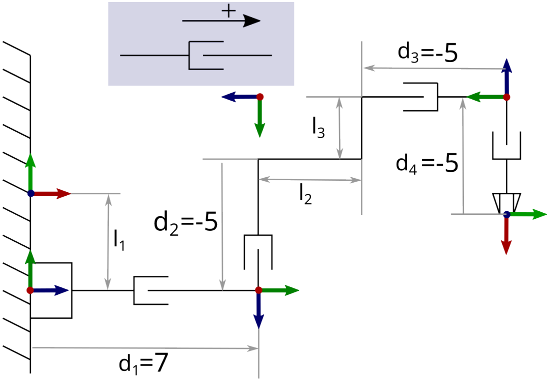
Example: spatial robot in plane

xyz=rgb

- Initial transformation: $T_y(-l_1)R_y(90^\circ)$

JointType	θ	d	a	α
P	0	d_1	0	90°
P	0	$d_2 - l_3$	0	90°
P	0	$d_3 - l_2$	0	90°
P	0	d_4	0	0

- We need to include helper frame before the gripper
  - x_1 is not perpendicular to z_0 and z_1
- Final transformation: $R_y(90^\circ)R_x(180^\circ)$



Conclusion

- What is DH notation
  - $T_{\mathsf{DH}} = R_z(\theta)T_z(d)R_x(\alpha)T_x(a)$
  - Designed for open kinematic chains with revolute and prismatic joints
- How to assign frames