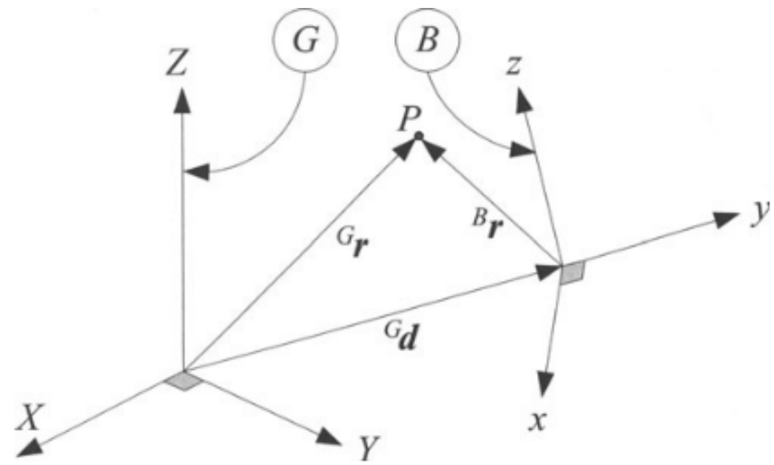
Last Time

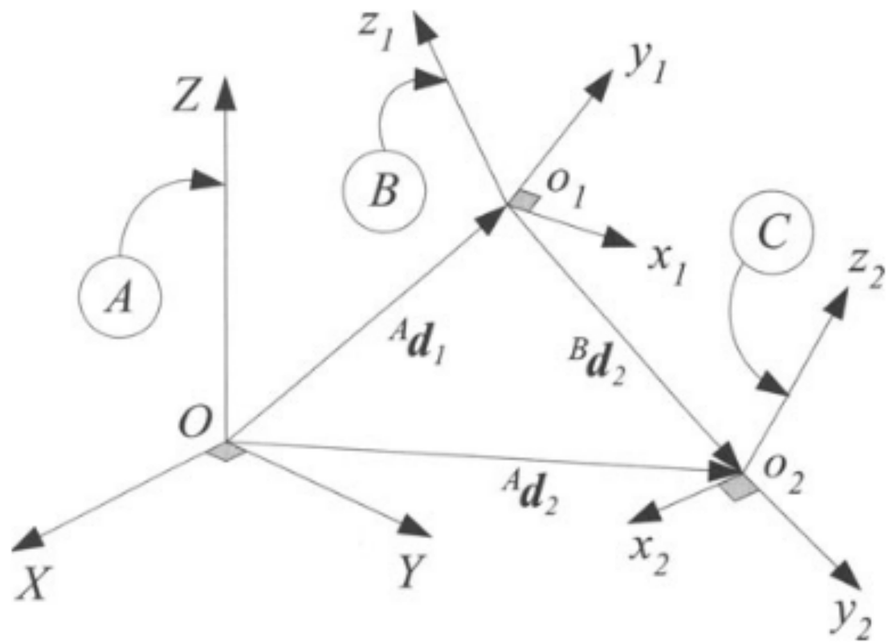
$$ {}^{G}T_{B} = \begin{bmatrix} {}^{G}R_{B} & {}^{G}\mathbf{d} \\ 0 & 1 \end{bmatrix}, \qquad {}^{G}T_{B}^{-1} = {}^{B}T_{G} = \begin{bmatrix} {}^{G}R_{B}^{T} & -{}^{G}R_{B}^{T}\mathbf{d} \\ 0 & 1 \end{bmatrix} $$

$$ {}^{G}\mathbf{r}_{p} = {}^{G}T_{B}\,{}^{B}\mathbf{r}_{p} $$



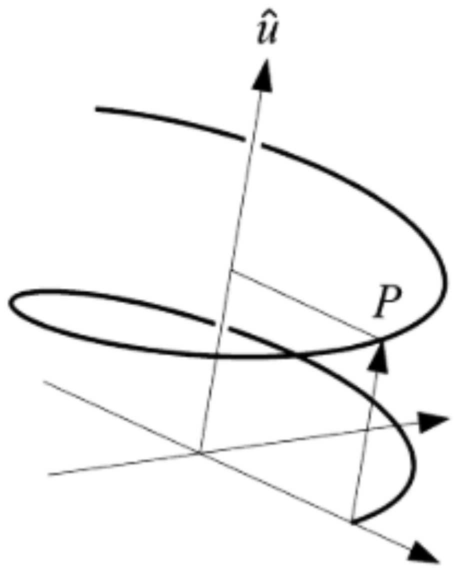
Homogenous Transformations combines Rotation and Translation into matrix operation

Last Time



$$^{A}T_{C} = {}^{A}T_{B}\,{}^{B}T_{C}$$

There is another method for generalized displacement (Ch 4.5-4.10)



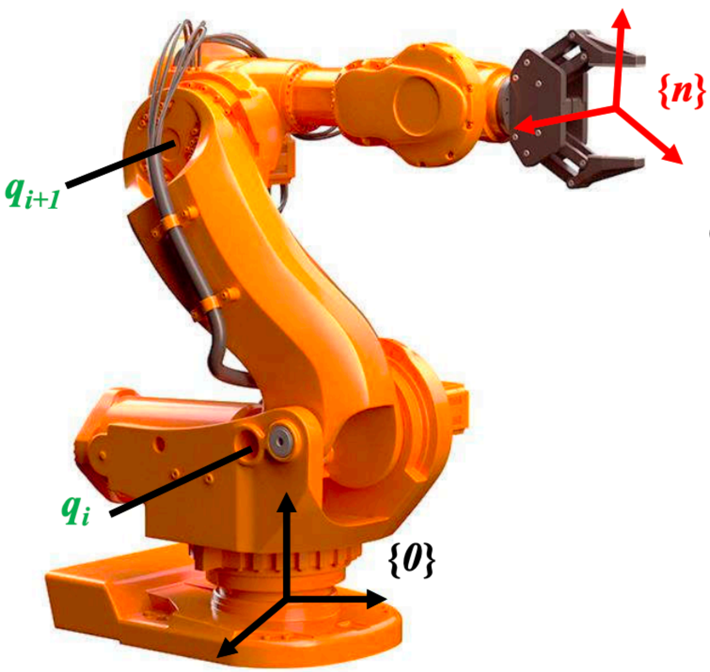
Screw Coordinates—essentially a generalization of Euler's Theorem.

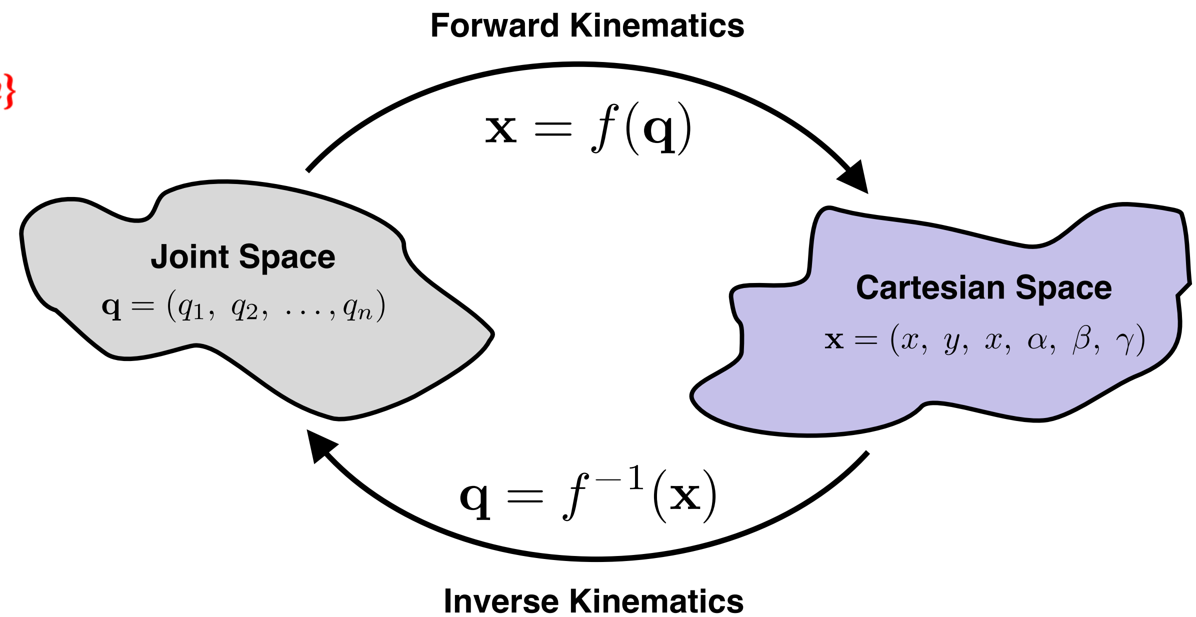
Chasles Theorem: Any rigid body displacement can be produced by a translation along an axis combined with a unique rotation about that axis

We will discuss next lecture.

Ch. 5.1-5.3

Today: Forward Kinematics





Agenda

1. Denavit-Hartenberg notation
2. Transformation between adjacent frames
3. Forward kinematics
4. Universal Robot Description Format (URDF)
5. Matlab Examples

DH Method Overview

- Basic Idea: use four parameters that encode reference frame orientation for each joint

- Use these parameters to generate the homogenous transformation between consecutive joints:

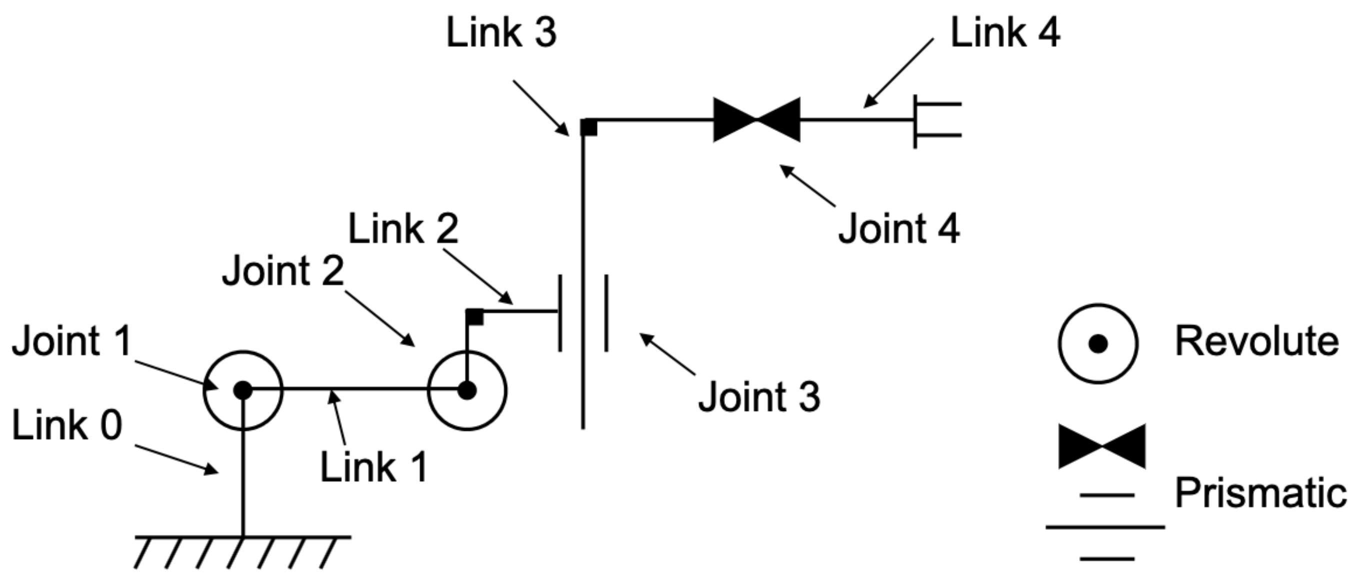
$$^{i-1}T_i$$

- Combine all transformation matrices:

$$^{B}T_e = \ ^{B}T_1 \ ^{1}T_2 \ ^{2}T_3 \ldots \ ^{n-1}T_n \ ^{n}T_e$$

Step 0: Identify joints and links

- Joints: 1, …, n
- Links: 0, …, n



Step 1: assign <u>all</u> z-axes

Assign axis z_0 to z_{n-1} to joints 1 through n

Joint i	**Axis** z_{i-1}	**Positive Direction**
Revolute	axis of rotation	positive angle RH-rule
prismatic	axis of translation	positive displacement

- z-axis represent the degree of freedom of each joint!
- base frame can be located anywhere along z0-axis

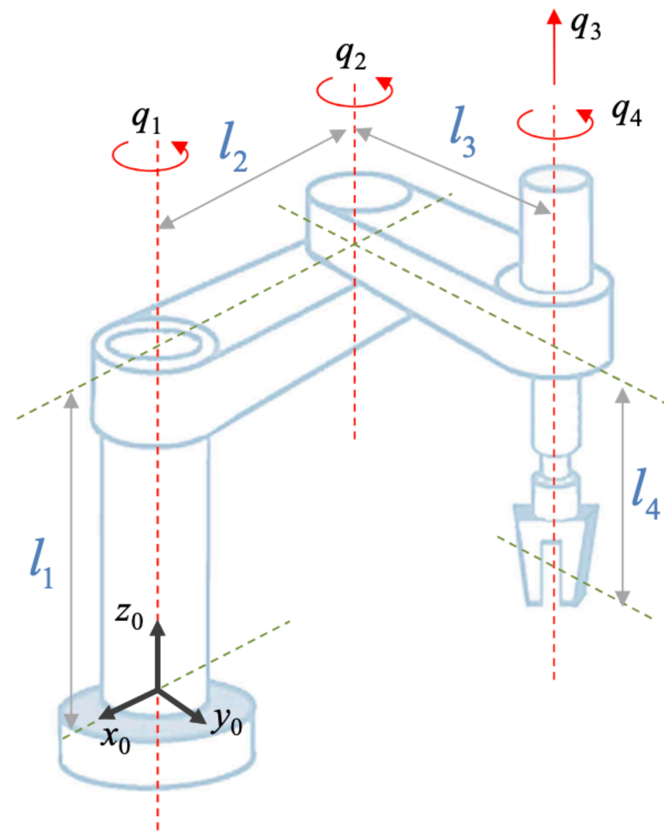
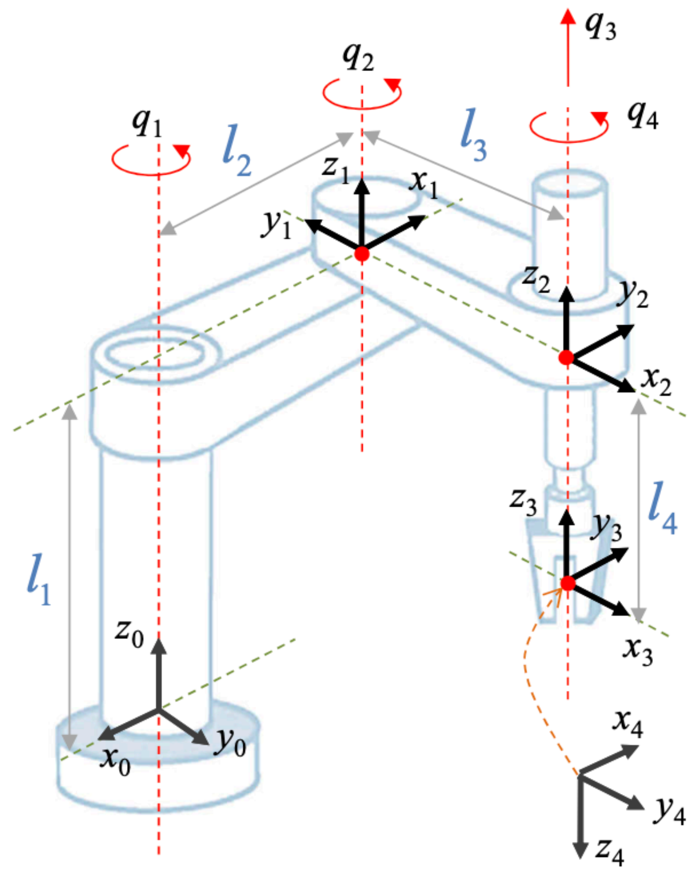


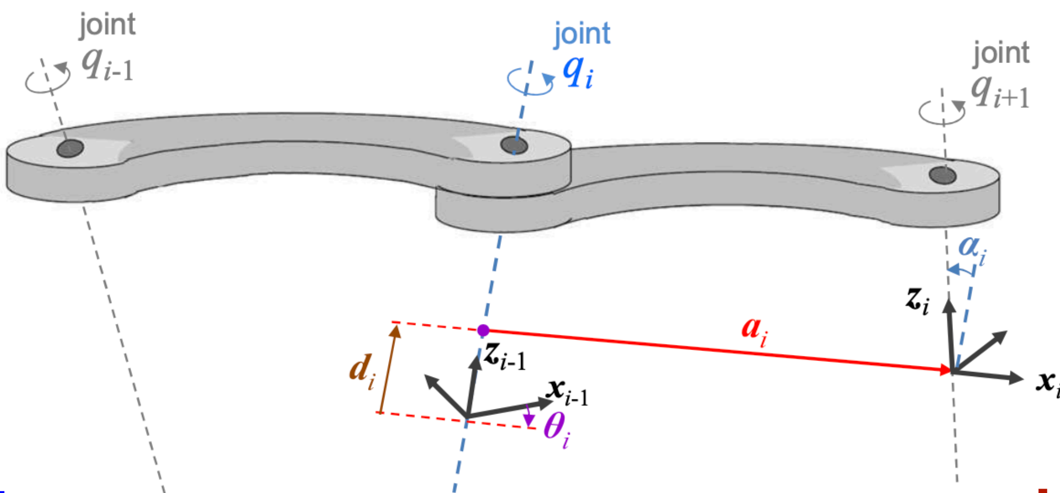
image credit: Oscar Ramos, UTEC

Step 2: frame origin

- assign axis x_i in the direction of $z_{i-1} \times z_i$. If they are parallel assign along common normal between z_{i-1} & z_i
- assign axis y_i to complete the frame following right hand rule
- tool frame (end effector frame)
  - $x_n$orthogonal to z_{n-1}
  - z_n pointing outwards



Step 3: DH parameters



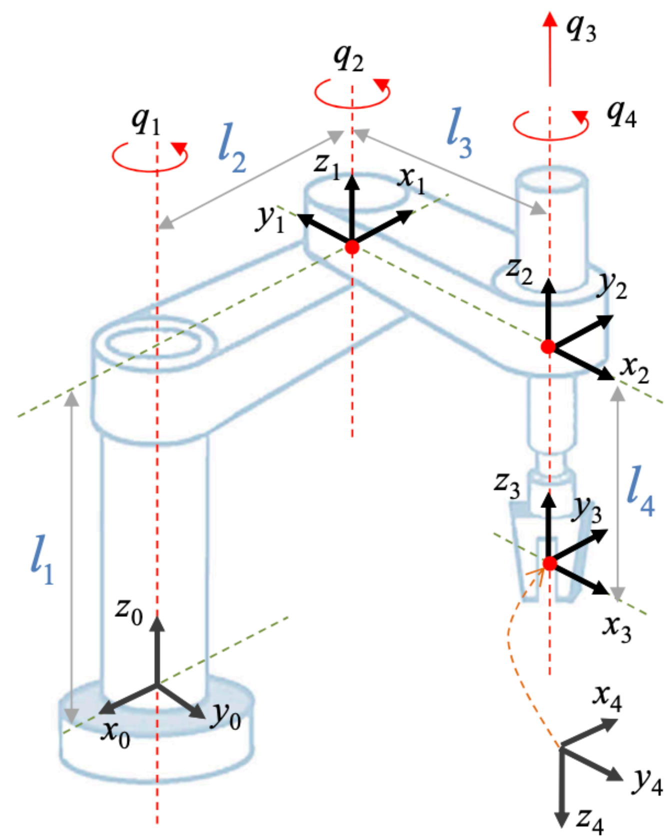
Joint Parameters

Link Parameters

- d_i: distance from the origin of {i-1} to the intersection of z_{i-1} with x_i along z_{i-1}
- θ_i: rotation angle from x_{i-1} with x_i about z_{i-1}
- a_i: distance from the intersection of z_{i-1} with x_i along x_i
- α_i: angle from z_{i-1} with z_i about x_i

Step 3: DH parameters

Joint i	d_i	θ_i	a_i	α_i
1				
2				
3				
4				



- d_i: distance from the origin of {i-1} to the intersection of z_{i-1} with x_i along z_{i-1}
- θ_i: rotation angle from x_{i-1} with x_i about z_{i-1}
- a_i: distance from the intersection of z_{i-1} with x_i along x_i
- α_i: angle from z_{i-1} with z_i about x_i

Step 4: generate transformations

$$^{i-1}T_i = D_{z_{i-1},d_i} R_{z-1,\theta_i} D_{x_{i-1},a_i} R_{x_{i-1},\alpha_i}$$

$$R_{x_{i-1},\alpha_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad D_{x_{i-1},a_i} = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_{z_{i-1},\theta_i} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad D_{z_{i-1},d_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Each $^{i-1}T_i$ **must be a function of either** θ_i **or** d_i

$$^{G}T_n(\mathbf{q}) = ^{G}T_1(q_1)\,^{1}T_2(q_2)\,^{2}T_3(q_3)\ldots ^{n-1}T_n(q_n)$$

Transformation from base to end-effector is a function of the joint <u>all</u> parameters

Consecutive transformation are a function of respective joint parameter

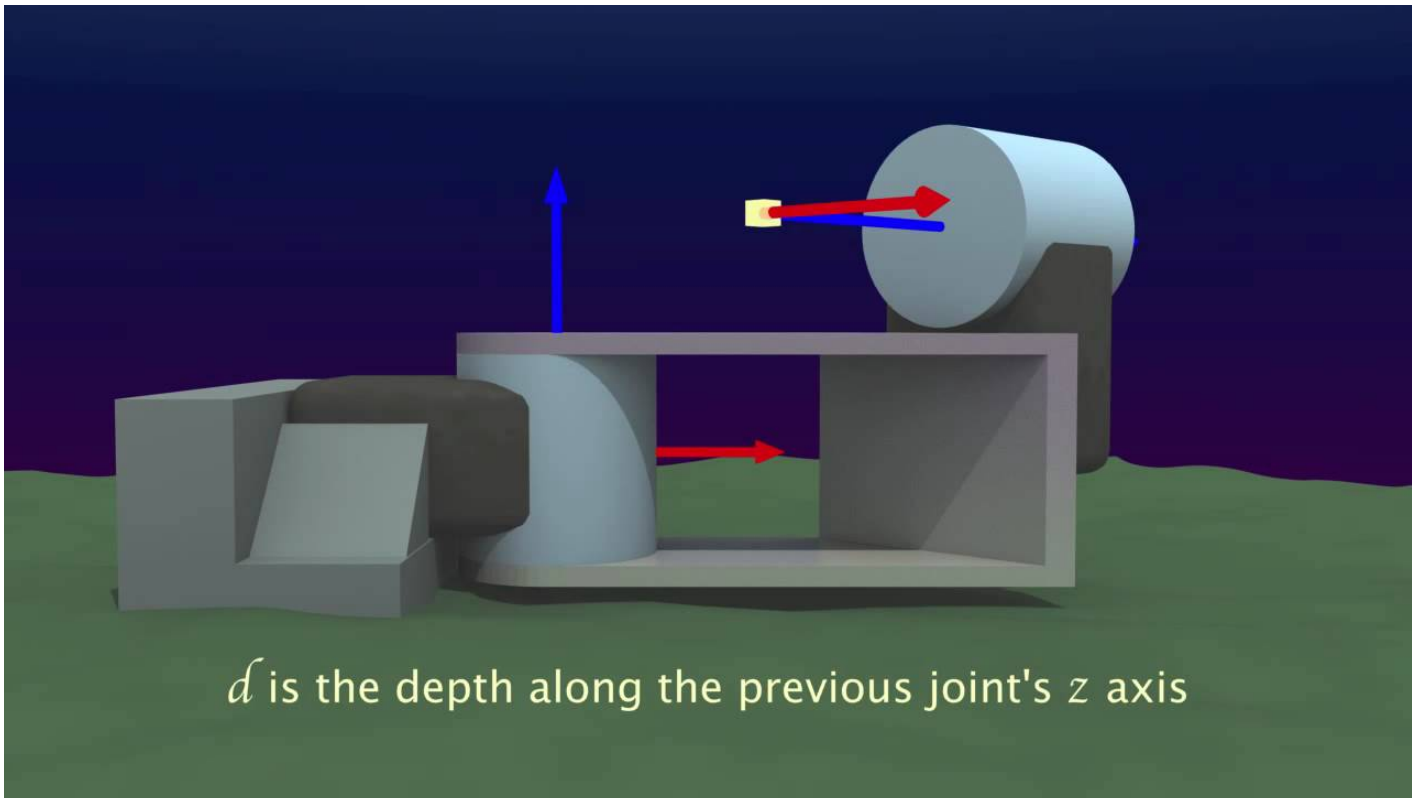
$$q_i = \theta_i \quad \text{or} \quad d_i$$

Orientation of end-effector

The origin of the end-effector in the base frame

$$^{G}T_n(\mathbf{q}) = \begin{bmatrix} ^{G}R_n(\mathbf{q}) & ^{G}\mathbf{r}_n(\mathbf{q}) \\ 0 & 1 \end{bmatrix}$$

Very nice illustration



https://www.youtube.com/watch?v=rA9tm0gTIn8