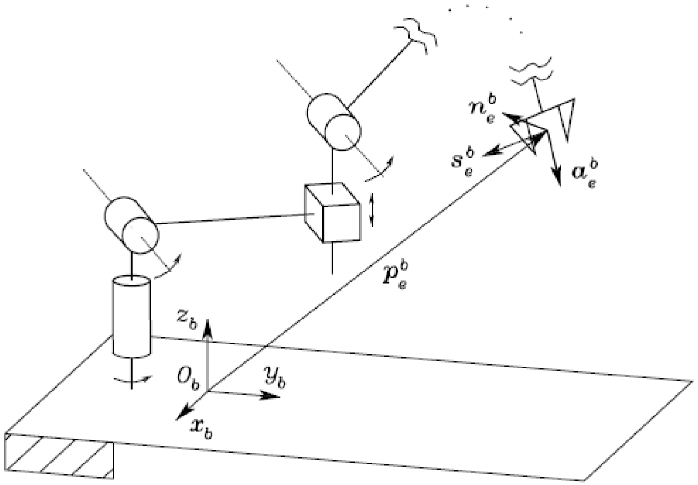
COMS W4733: Computational Aspects of Robotics

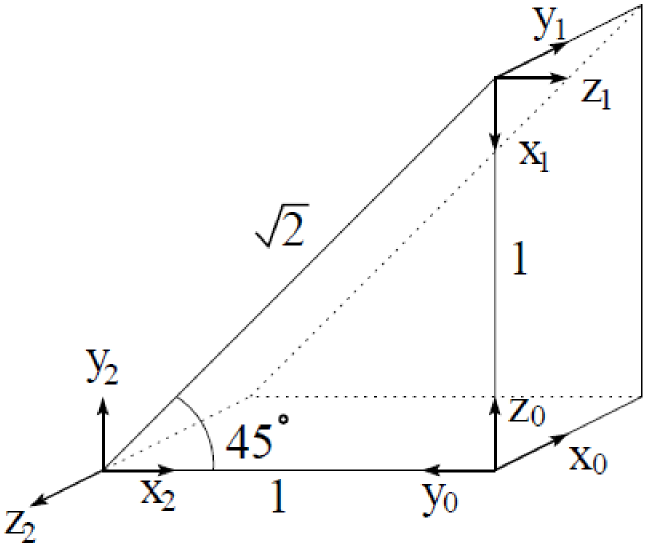
Lecture 3: Forward Kinematics



Instructor: Tony Dear

Homogeneous Transformations

- Break complex transformations down into multiple elementary transformations.
- Draw intermediate frames and compose elementary transformations together.
- The same transformation can often be achieved in several different ways.
- Pay attention to sub/superscripts!
- A_i^j: Pose of frame i relative to frame j

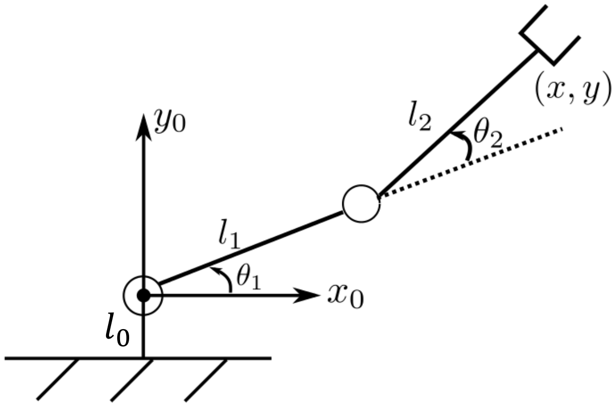


Manipulators

- *Open-chain* manipulators
- Fixed at a *base*; *end-effector* (e.g. gripper) at the end
- Series of *links* (rigid bodies) connected by *joints*

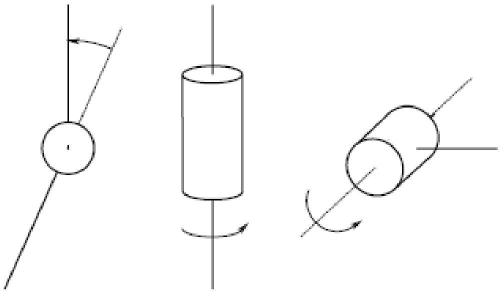
- Each joint is a single DOF, described by a *joint variable*

- Joint variables go from 1 to n
- Link constants go from 0 to n
- **Joint i moves link i** (link 0 does not move)

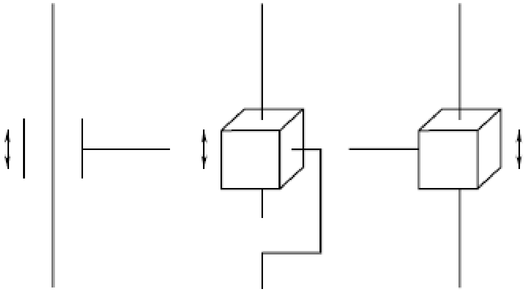


Joint Representations

- Joints encode *constraints* on relative link motions
- Typically only allow rotation about or translation in one specific direction



Revolute joints



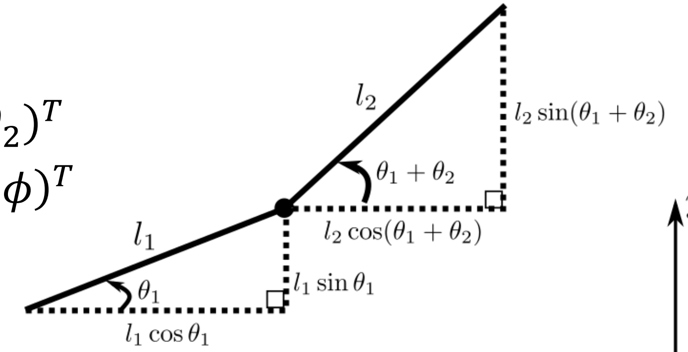
Prismatic joints

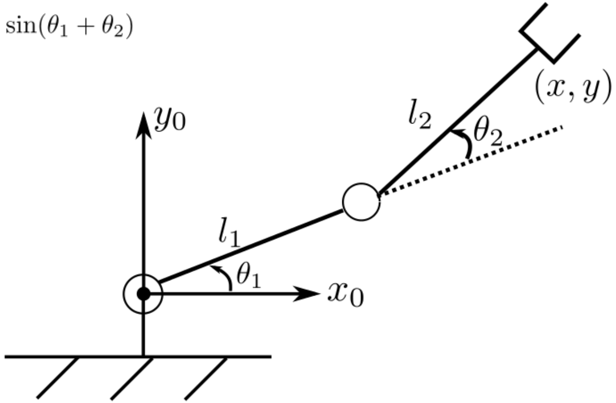
$$q_i = \begin{cases} \theta_i & \text{if joint } i \text{ is revolute} \\ d_i & \text{if joint } i \text{ is prismatic} \end{cases}$$

Forward Kinematics

- Description of the end-effect pose as function of the joint variables
  - We'll later consider the harder *inverse* problem
- Kinematics does **not** consider forces, torques, and mass (**dynamics**)

- Ex: Planar RR arm
  - Joints: $(q_1, q_2)^T = (\theta_1, \theta_2)^T$
  - End-effector pose: $(x, y, \phi)^T$





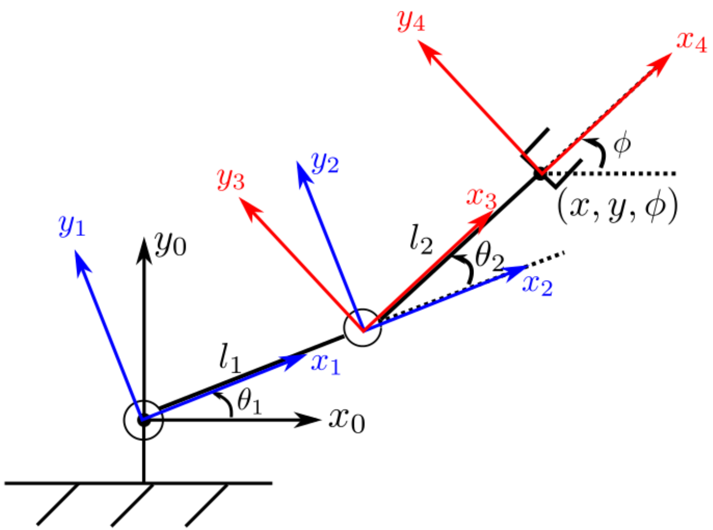
$$x = l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2)$$
$$y = l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2)$$

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

Coordinate Transformations

$$A_1^0 = \begin{bmatrix} c_1 & -s_1 & 0 \\ s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad A_2^1 = \begin{bmatrix} 1 & 0 & l_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$A_3^2 = \begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad A_4^3 = \begin{bmatrix} 1 & 0 & l_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



$c_i = \cos\theta_i$
$s_i = \sin\theta_i$

$$\boxed{A_4^0 = A_1^0 A_2^1 A_3^2 A_4^3}$$

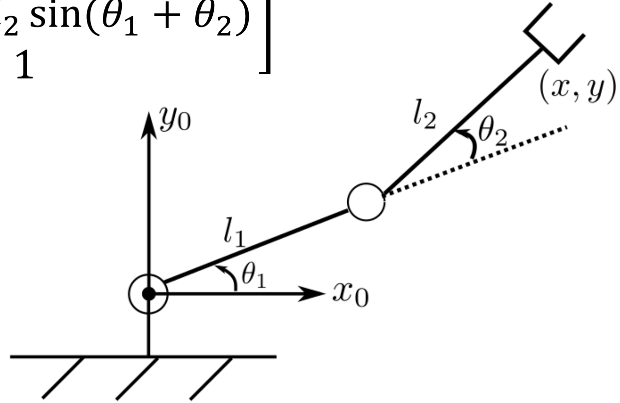
Describes movement of end-effector with respect to the base

Coordinate Transformations

$$\boxed{\boldsymbol{A}_4^0 = \boldsymbol{A}_1^0 \boldsymbol{A}_2^1 \boldsymbol{A}_3^2 \boldsymbol{A}_4^3}$$

$$\boldsymbol{A}_1^0 = \begin{bmatrix} c_1 & -s_1 & 0 \\ s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \boldsymbol{A}_2^1 = \begin{bmatrix} 1 & 0 & l_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \boldsymbol{A}_3^2 = \begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \boldsymbol{A}_4^3 = \begin{bmatrix} 1 & 0 & l_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{A}_4^0 = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) & l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) \\ 0 & 0 & 1 \end{bmatrix}$$



Denavit-Hartenberg Convention

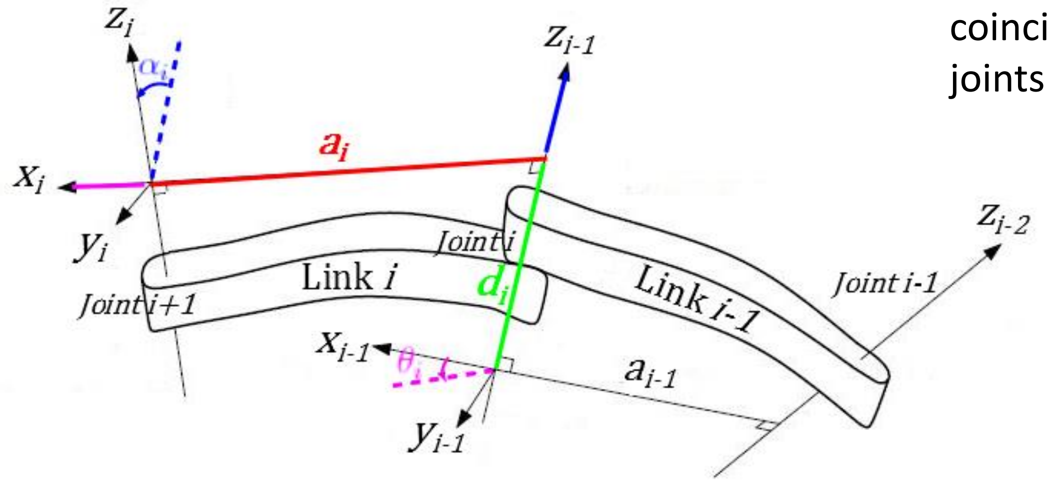
- Forward kinematics describe end effector relative to base frame
- No specification about intermediate frames

- **DH convention** is a universal language for describing any open chain
- Systematically summarizes each individual transformation with 4 parameters

- Two steps: Assign frames, then derive **DH parameters**

Frame Assignment

- Assign one frame per joint to satisfy the following:
  - z_i along rotation axis (revolute) or translation axis (prismatic) of joint $i+1$
  - O_i (origin) minimizes distance between z_{i-1} and z_i
  - x_i intersects and is perpendicular to z_{i-1}
  - y_i chosen to make frame right-handed



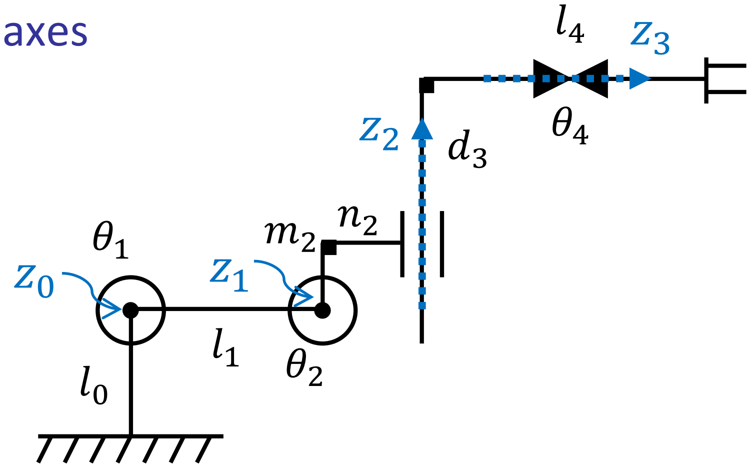
Frames not necessarily coincident with the joints themselves!

Example: z axes

- Step 0: Identify and label all joints and links
  - Links go from 0 to n; joints go from 1 to n. Joint i moves link i!
  - Label joint variables as either θ_i (revolute) or d_i (prismatic)

- Step 1: Place z_0 through z_{n-1} axes along joint axes
  - Positive along direction of positive displacement

- θ_1 and θ_2 revolute, z_0 and z_1 out of the plane
- d_3 prismatic, z_2 pointing upward
- θ_4 revolute, z_3 pointing rightward

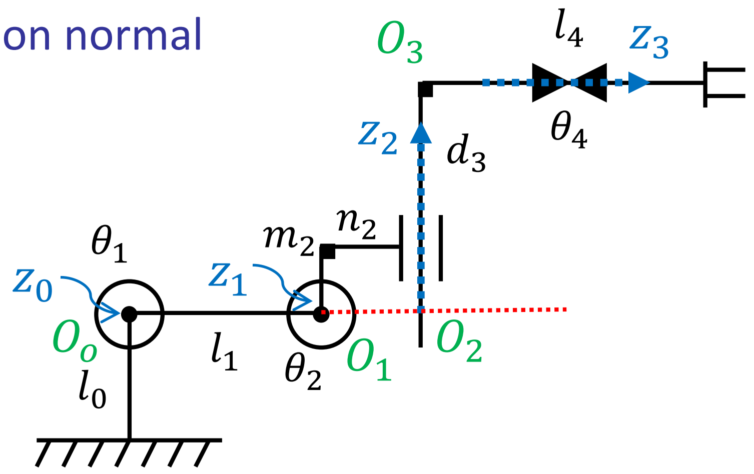


Example: Assigning Frame Origins

- Step 2: Locate origin of each coordinate frame
  - Base frame 0: Coincide origin O_0 with the first joint
  - Other frames: Choose O_i to minimize distance from O_{i-1}

- If z_{i-1} and z_i do not intersect, use their common normal
- Otherwise, use their intersection

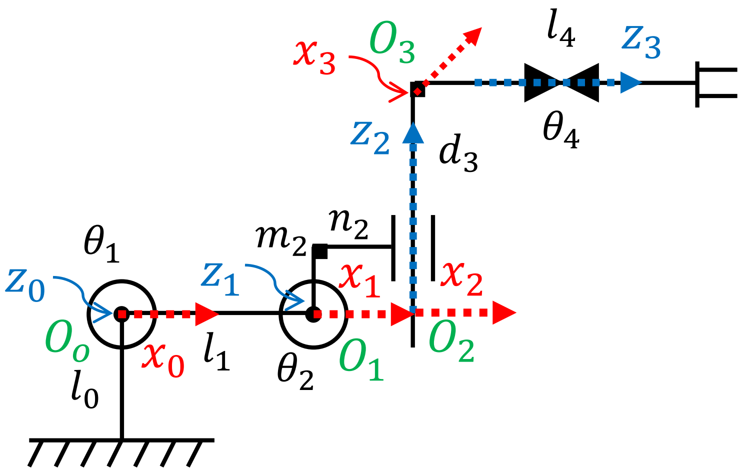
- z_0 and z_1 parallel; O_1 in the plane
- z_1 and z_2 are skew; O_2 on common normal
- z_2 and z_3 intersect; O_3 on intersection



Example: x axes

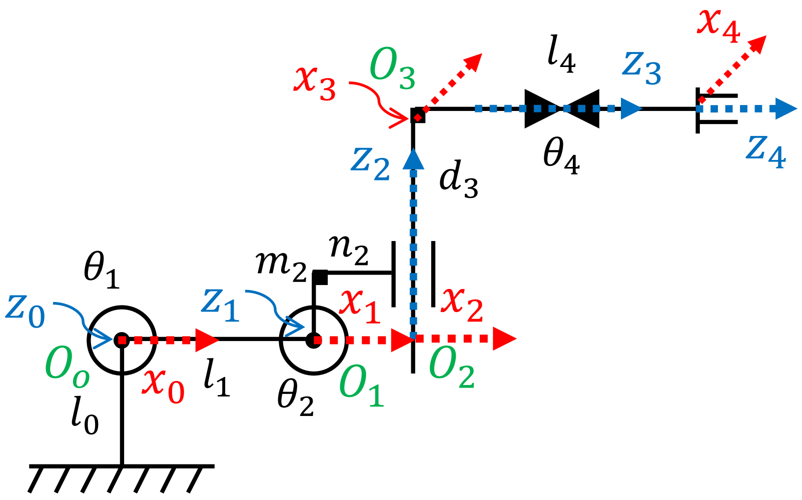
- Step 3: x_i intersects and is perpendicular to z_{i-1}
  - Usually have choice of two opposite directions
  - Try to choose a convenient direction and minimize frame transformations

- Arbitrarily choose x_0 pointing right
- x_1 points to the right, perpendicular to z_0
- x_2 points to the right, perpendicular to z_1
- x_3 must go into (or come out of) the plane



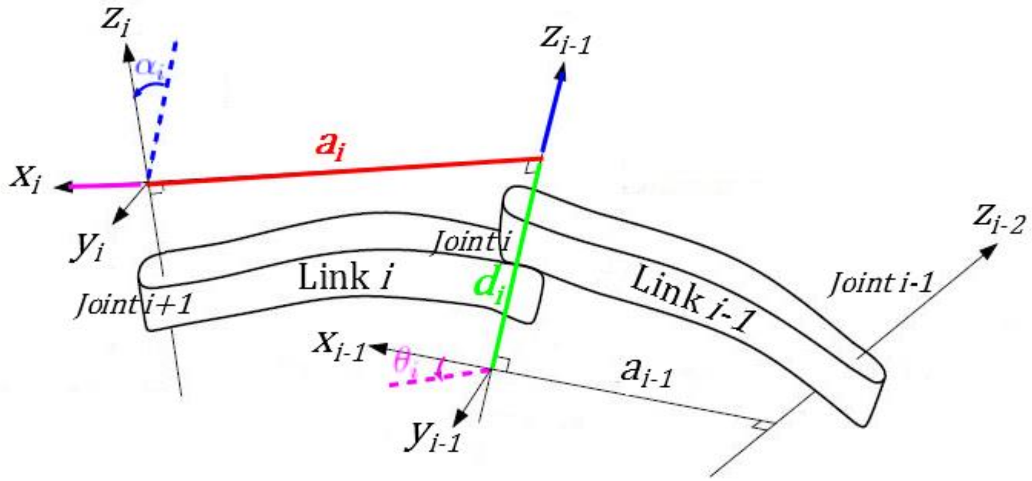
Tool Frame

- Last frame (frame n) typically goes on the end-effector
- Place origin O_n on center of gripper and align z_n with z_{n-1}
- Choose x_n so that it intersects z_{n-1}



Denavit-Hartenberg Parameters

- Our frame definition ensures that each transformation between frame O_i and O_{i-1} can be summarized by the same four parameters!
  - Rotate about z_{i-1} by θ_i (**joint angle**) and translate along z_{i-1} by d_i (**link offset**)
  - Translate along x_i by a_i (**link length**) and rotate about x_i by α_i (**link twist**)



Forward Kinematic Map

- Our frame definition ensures that each transformation between frame O_i and O_{i-1} can be summarized by the same four parameters!
  - Rotate about z_{i-1} by θ_i (**joint angle**) and translate along z_{i-1} by d_i (**link offset**)
  - Translate along x_i by a_i (**link length**) and rotate about x_i by α_i (**link twist**)

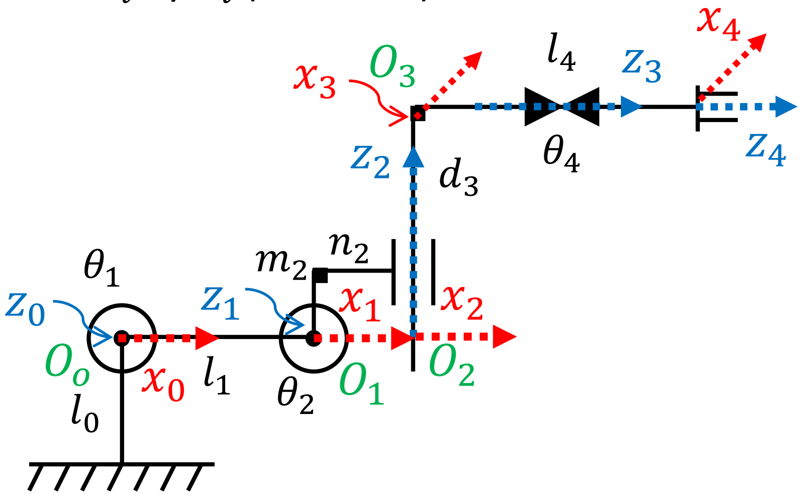
$$\boldsymbol{A}_i^{i-1} = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Overall forward kinematic map then found as $\boldsymbol{T}_n^0 = \boldsymbol{A}_1^0 \boldsymbol{A}_2^1 \cdots \boldsymbol{A}_{n-1}^{n-2} \boldsymbol{A}_n^{n-1}$

Example: DH Parameters

- Instead of carrying around a bunch of homogeneous transform matrices, we can summarize a manipulator's configuration with a table of $4n$ parameters
  - Rotate about z_{i-1} by θ_i (**joint angle**) and translate along z_{i-1} by d_i (**link offset**)
  - Translate along x_i by a_i (**link length**) and rotate about x_i by α_i (**link twist**)

Link	a_i	α_i	d_i	θ_i
1	l_1	0	0	θ_1
2	n_2	-90	0	θ_2
3	0	90	$m_2 + d_3$	90
4	0	0	l_4	θ_4



Example: DH Parameters

- Each row has exactly one joint variable, either revolute or prismatic
- Row i corresponds to a homogeneous transformation between frames $i-1$ and i
- Composition of all n transformations gets us the FK map

Link	a_i	α_i	d_i	θ_i
1	l_1	0	0	θ_1
2	n_2	-90	0	θ_2
3	0	90	$m_2 + d_3$	90
4	0	0	l_4	θ_4

$$\boldsymbol{T}_4^0 = \begin{bmatrix} c_1 & -s_1 & 0 & l_1 c_1 \\ s_1 & c_1 & 0 & l_1 s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} c_2 & 0 & -s_2 & n_2 c_2 \\ s_2 & 0 & c_2 & n_2 s_2 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$\times \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & m_2 + d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} c_4 & -s_4 & 0 & 0 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 1 & l_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Considerations

- FK map $\boldsymbol{T}_n^0 = \begin{bmatrix} \boldsymbol{R}_n^0 & \boldsymbol{o}_n^0 \\ \boldsymbol{0}^T & 1 \end{bmatrix}$ provides both position $\boldsymbol{o}_n^0$ and orientation $\boldsymbol{R}_n^0$ of end-effector
- Sometimes only require o_n^0: simply extract right column from matrix

- $\boldsymbol{T}_n^0$ depends only on definition of base and end-effector frames
- Possible to define different intermediate frames
- Sometimes may be helpful to draw manipulator in an easier reference configuration