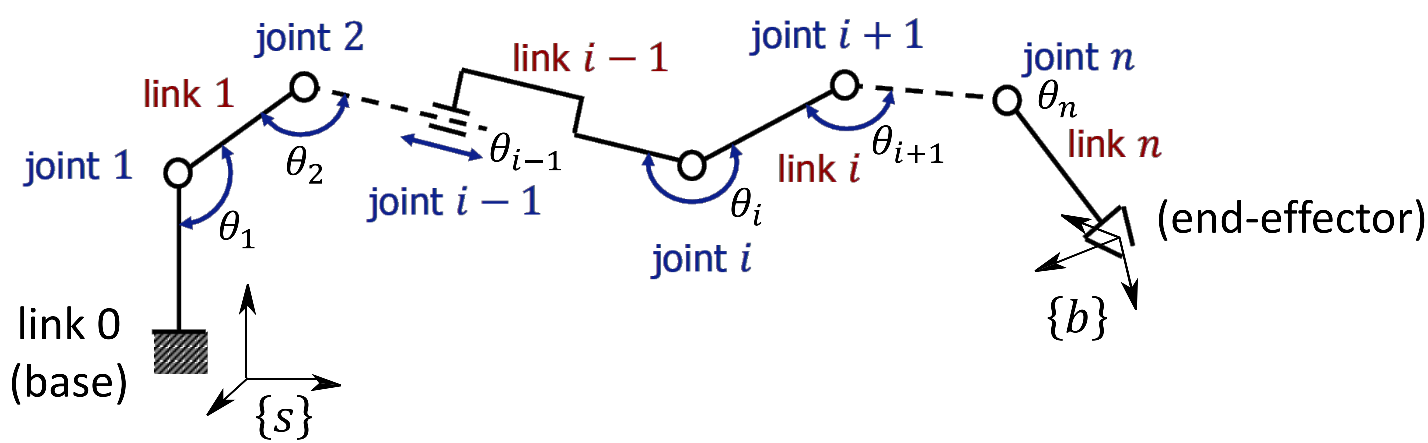
Ch5: Forward Kinematics

Forward Kinematics

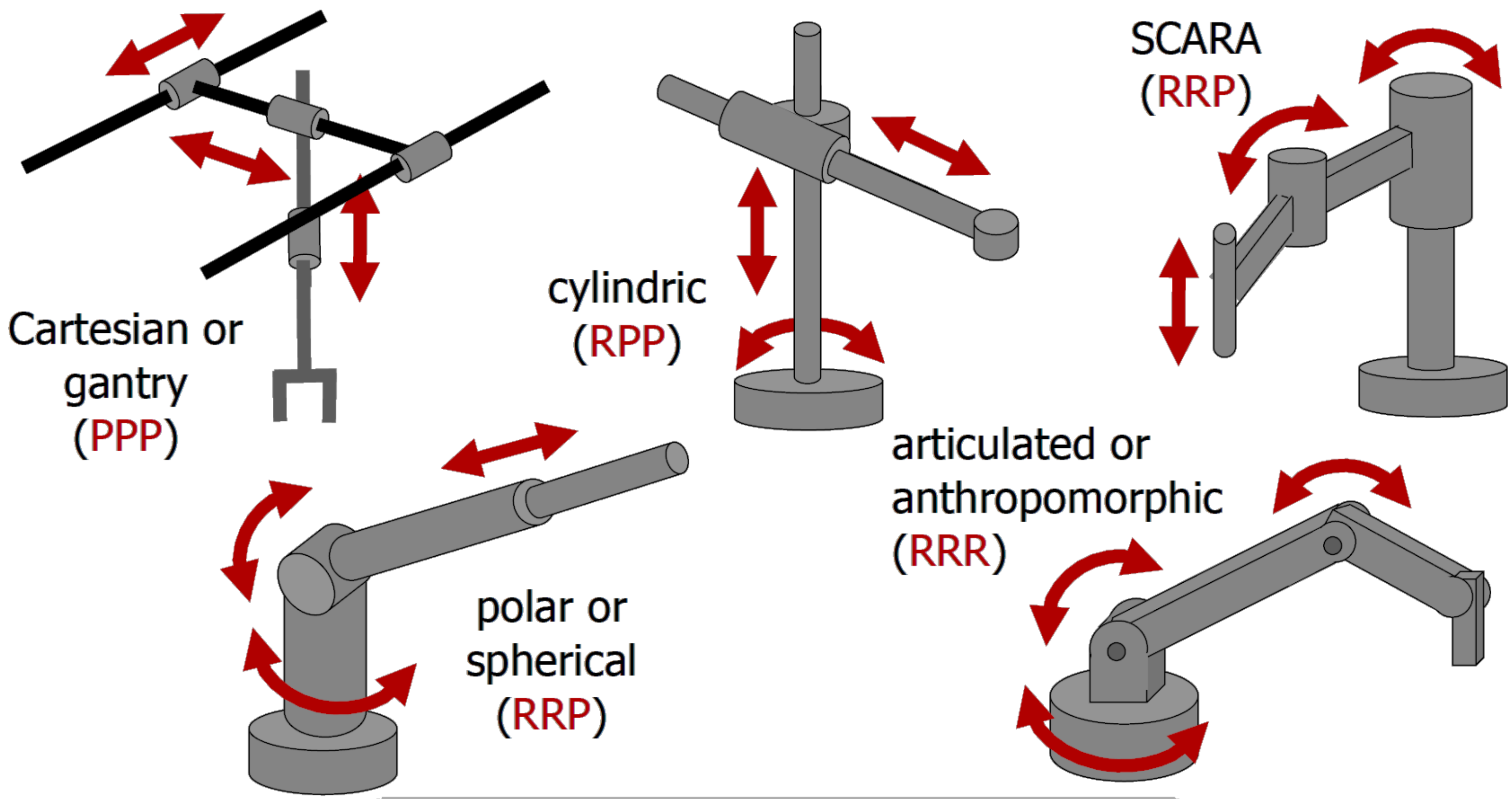
Assumptions: Open-Chain Robot Manipulators

Robot manipulators are articulated mechanical systems composed of links connected by joints. In this course, we consider only n-DOF open-chain (serial) robot manipulators with revolute and/or prismatic joints.

- The generalized joint coordinate (joint position) denoted by θ_i corresponds to the angular displacement of a revolute joint or the linear displacement of a prismatic joint. Thus, vector of joint positions: $\boldsymbol{\theta} \in \mathbb{R}^n$
- Each joint is independently controlled through an actuator.
- The joint positions are measured by sensors (encoders) placed at the actuators, that are usually located at the joints.

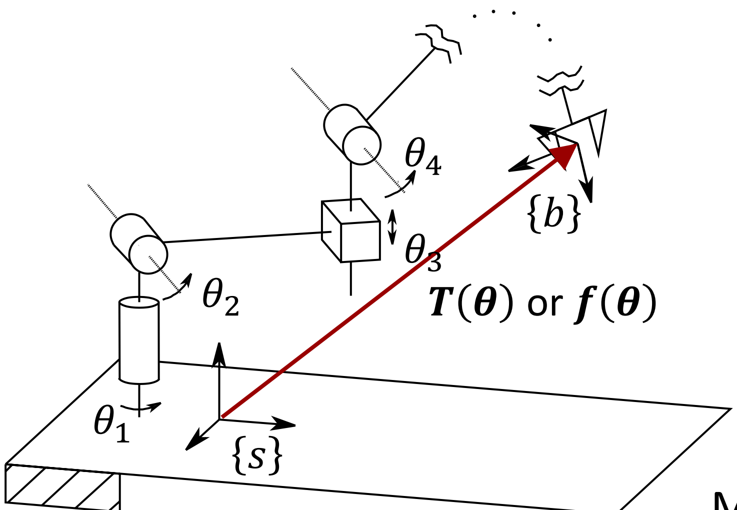


Classification of Robotic Arms by Kinematic Type of First 3 DOFs



Forward (Direct) Kinematics

The **Forward (Direct) Kinematics** of a robot refers to the calculation of the position and orientation (**pose**) of its end-effector frame from its joint positions $\boldsymbol{\theta}$ (the relative angle/displacment between a link and the following one).



- "Geometric" forward kinematics:

$$\text{Given } \boldsymbol{\theta} \in \mathbb{R}^n, \text{ Find } \boldsymbol{T}_{sb} = \boldsymbol{T}(\boldsymbol{\theta}) \in SE(3)$$

$$\boldsymbol{T} = (\boldsymbol{R}, \boldsymbol{p}) : \mathbb{R}^n \to SE(3)$$

- "Minimum-Coordinate" forward kinematics:

$$\text{Given } \boldsymbol{\theta} \in \mathbb{R}^n, \text{ Find } \boldsymbol{x}_e = \boldsymbol{f}(\boldsymbol{\theta}) \in \mathbb{R}^r$$

$$\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^r$$

$$\boldsymbol{x}_e = \begin{bmatrix} \boldsymbol{\phi} \\ \boldsymbol{p} \end{bmatrix}, \boldsymbol{x}_e = \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{p} \end{bmatrix}, \boldsymbol{x}_e = \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{p} \end{bmatrix} \quad \begin{aligned} \boldsymbol{\phi} &= (\alpha, \beta, \gamma) \\ \boldsymbol{q} &= (q_0, q_1, q_2, q_3) \\ \boldsymbol{r} &= \hat{\boldsymbol{\omega}}\theta \end{aligned}$$

Methods of forward kinematics:

- By inspection of geometry [only for simple robots]
- Product of Exponentials (PoE)
- Denavit-Hartenberg (DH)

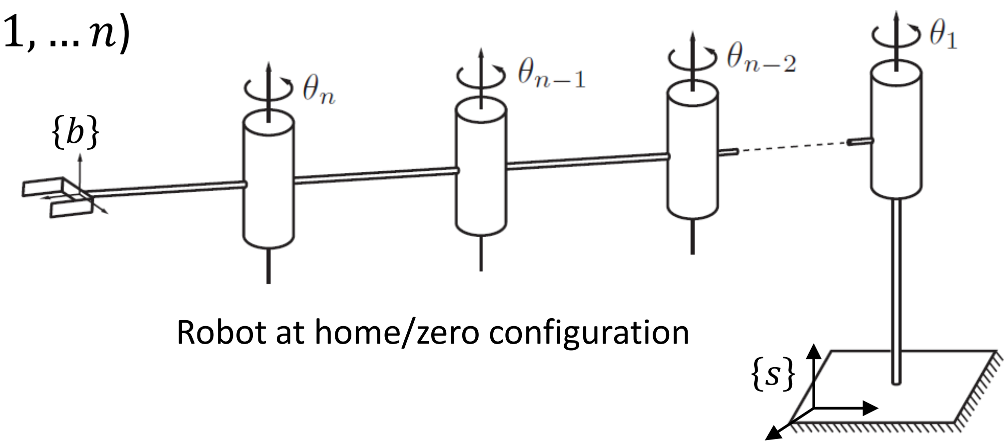
Product of Exponentials (PoE) Formulation in Base Frame

Forward Kinematics Using Product of Exponentials (PoE) Formulation in Base Frame

Calculating the forward kinematics of an open chain using the **space form** of the PoE formula:

- Assign a fixed base frame $\{s\}$.
- Assign an end-effector frame $\{b\}$.
- Let $\boldsymbol{M} = \boldsymbol{T}_{sb}(\mathbf{0}) \in SE(3)$ be the configuration of $\{b\}$ relative to $\{s\}$ when the robot is in its home or zero configuration ($\boldsymbol{\theta} = \mathbf{0}$).
- Specify the direction of positive displacement (rotation for revolute joints, translation for prismatic joints) for each joint.
- Find the screw axis $\boldsymbol{S}_i$ of the joints ($i = 1, \ldots n$) as expressed in $\{s\}$ when $\boldsymbol{\theta} = \mathbf{0}$.

$$\boldsymbol{S}_i = \begin{bmatrix} \boldsymbol{S}_{\omega,i} \\ \boldsymbol{S}_{v,i} \end{bmatrix} \in \mathbb{R}^6$$



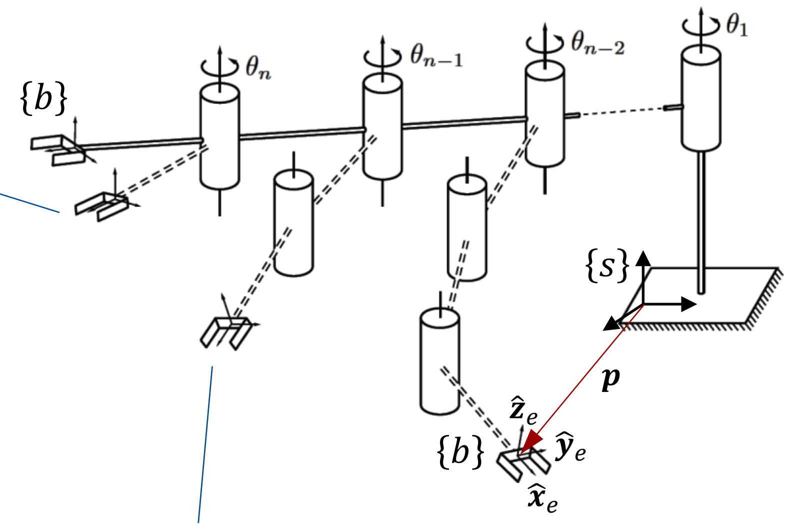
Robot at home/zero configuration

Forward Kinematics Using Product of Exponentials (PoE) Formulation in Base Frame (cont.)

Suppose that joint n is displaced by θ_n (for $\theta_1 = \cdots = \theta_{n-1} = 0$). Then, the new configuration of $\{b\}$ is

$$T_{sb} = e^{[S_n]\theta_n} M \in SE(3)$$

$$[S_n] = \begin{bmatrix} [S_{\omega,n}] & S_{v,n} \\ \mathbf{0} & 0 \end{bmatrix} \in se(3)$$



Now, suppose that joint $n-1$ is displaced by θ_{n-1} (for $\theta_1 = \cdots = \theta_{n-2} = 0$ and any fixed, but arbitrary, θ_n). Then, the new configuration of $\{b\}$ is

$$T_{sb} = e^{[S_{n-1}]\theta_{n-1}} \left(e^{[S_n]\theta_n} M \right)$$

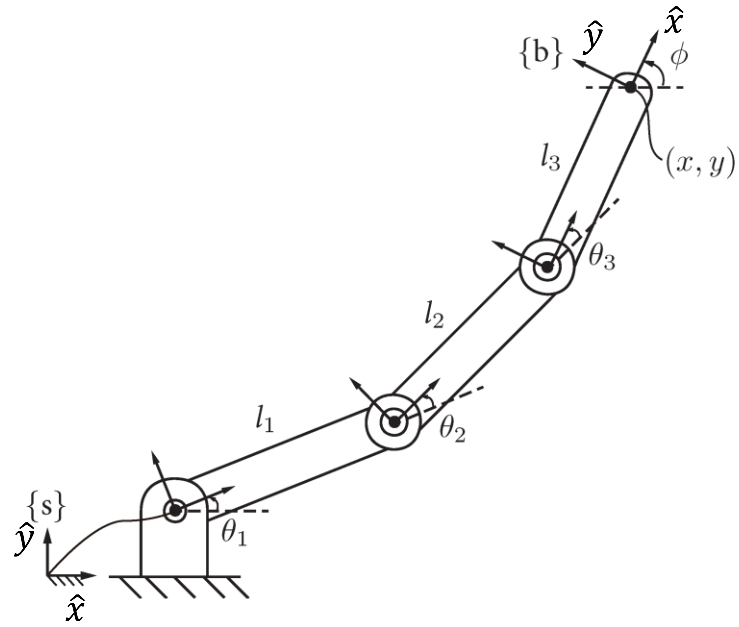
Continuing this for all the joints:

$$\boxed{T_{sb}(\theta) = e^{[S_1]\theta_1} \ldots e^{[S_{n-1}]\theta_{n-1}} e^{[S_n]\theta_n} M} = \begin{bmatrix} \hat{x}_e & \hat{y}_e & \hat{z}_e & p \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The screw axes $S_1, \ldots, S_n$ expressed in $\{s\}$, corresponding to the joint motions when the robot is at its home/zero configuration ($\theta = \mathbf{0}$).

Example: 3R Planar Robot

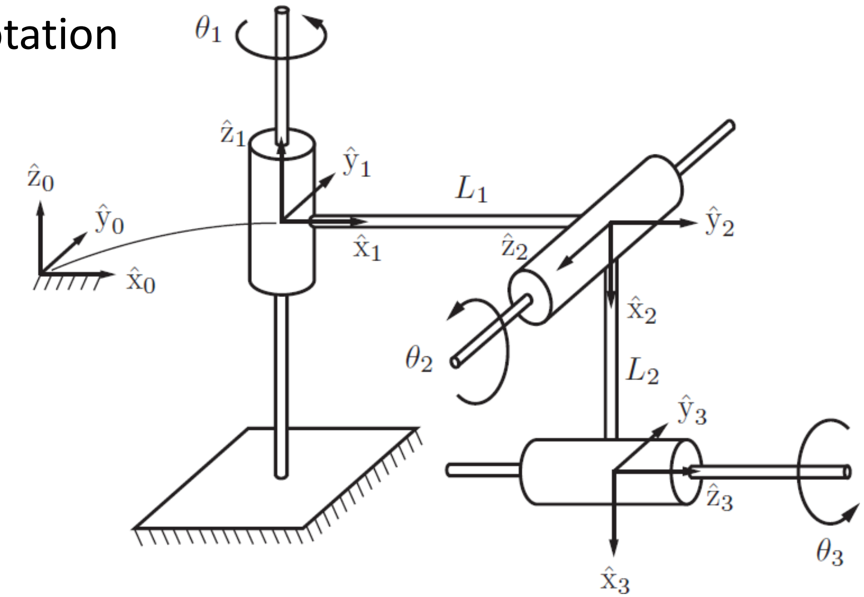
Find the geometric forward kinematics using the **space form** of the PoE formulation.



Example: 3R Spatial Robot

The zero position and the direction of positive rotation for each joint axis are as shown in the figure.

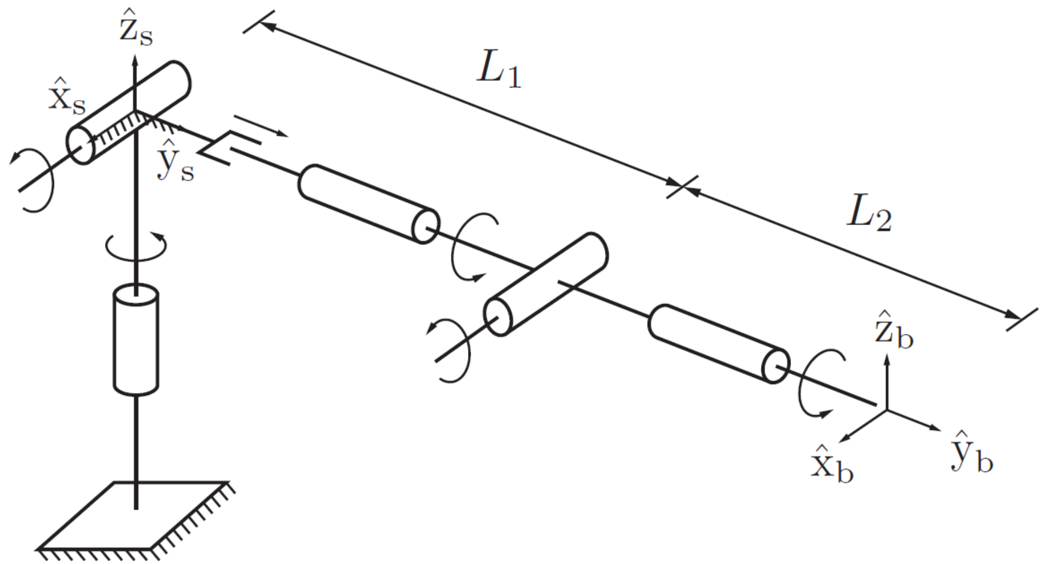
Find the geometric forward kinematics using the **space form** of the PoE formulation.



Example: RRPRRR Spatial Robot

The zero position and the direction of positive rotation for each joint axis are as shown in the figure.

Find the geometric forward kinematics using the **space form** of the PoE formulation.



Product of Exponentials (PoE) Formulation in End-Effector Frame

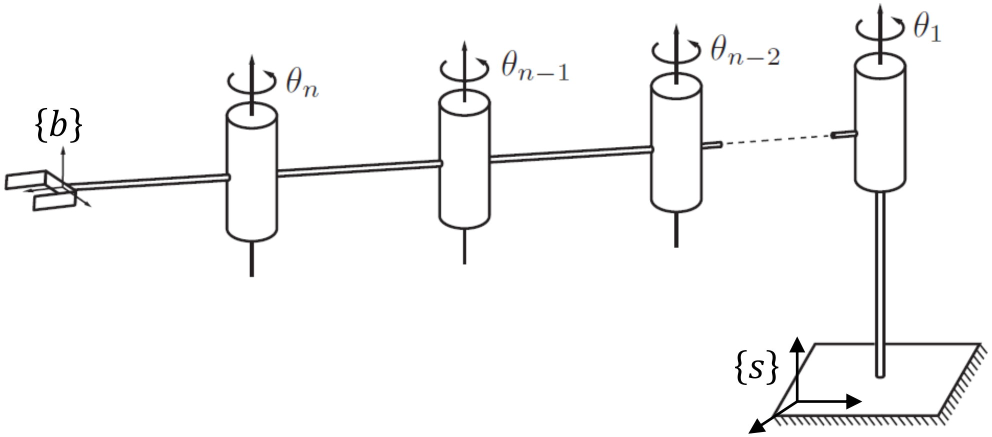
Forward Kinematics Using Product of Exponentials (PoE) Formulation in EE Frame

An alternative method to calculate the forward kinematics of an open chain is using the **body form** of the PoE formula.

- Assign a fixed base frame $\{s\}$.
- Assign an end-effector frame $\{b\}$.
- Let $\boldsymbol{M} \in SE(3)$ be the configuration of $\{b\}$ relative to $\{s\}$ when the robot is in its home or zero configuration.
- Specify the direction of positive displacement (rotation for revolute joints, translation for prismatic joints) for each joint.
- Find the screw axis $\boldsymbol{\mathcal{B}}_i$ of the joints $(i = 1, \ldots n)$ as expressed in $\{b\}$ when $\boldsymbol{\theta} = \boldsymbol{0}$.

$$\boldsymbol{\mathcal{B}}_i = \begin{bmatrix} \boldsymbol{\mathcal{B}}_{\omega,i} \\ \boldsymbol{\mathcal{B}}_{v,i} \end{bmatrix} \in \mathbb{R}^6$$

$$\boldsymbol{\mathcal{B}}_i = [\mathrm{Ad}_{\boldsymbol{M}^{-1}}]\boldsymbol{S}_i$$

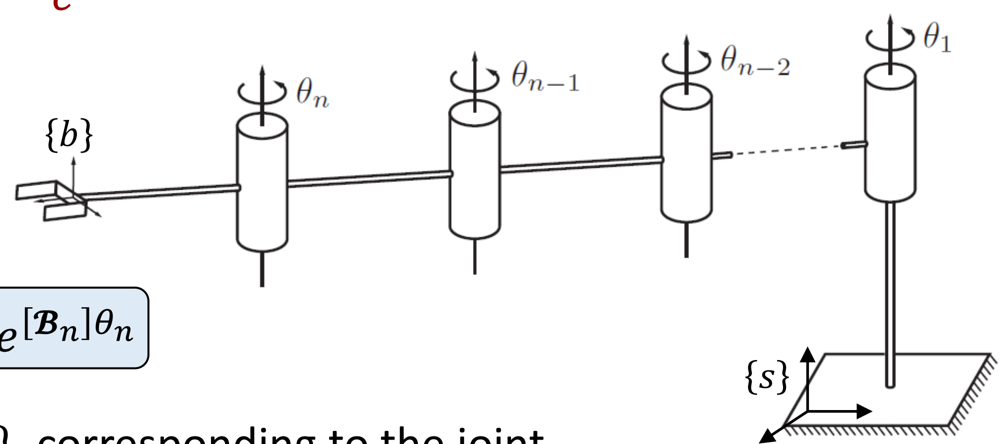


Forward Kinematics Using Product of Exponentials (PoE) Formulation in EE Frame (cont.)

We know that $e^{\boldsymbol{M}^{-1}\boldsymbol{P}\boldsymbol{M}} = \boldsymbol{M}^{-1}e^{\boldsymbol{P}}\boldsymbol{M}$, thus, $\boldsymbol{M}e^{\boldsymbol{M}^{-1}\boldsymbol{P}\boldsymbol{M}} = e^{\boldsymbol{P}}\boldsymbol{M}$.

$$\begin{aligned}
\boldsymbol{T}_{sb}(\boldsymbol{\theta}) &= e^{[\boldsymbol{\mathcal{S}}_1]\theta_1}\dots e^{[\boldsymbol{\mathcal{S}}_{n-1}]\theta_{n-1}}e^{[\boldsymbol{\mathcal{S}}_n]\theta_n}\boldsymbol{M}\\
&= e^{[\boldsymbol{\mathcal{S}}_1]\theta_1}\dots e^{[\boldsymbol{\mathcal{S}}_{n-1}]\theta_{n-1}}\boldsymbol{M}e^{\boldsymbol{M}^{-1}[\boldsymbol{\mathcal{S}}_n]\boldsymbol{M}\theta_n}\\
&= e^{[\boldsymbol{\mathcal{S}}_1]\theta_1}\dots \boldsymbol{M}e^{\boldsymbol{M}^{-1}[\boldsymbol{\mathcal{S}}_{n-1}]\boldsymbol{M}\theta_{n-1}}e^{\boldsymbol{M}^{-1}[\boldsymbol{\mathcal{S}}_n]\boldsymbol{M}\theta_n}\\
&= \boldsymbol{M}e^{\boldsymbol{M}^{-1}[\boldsymbol{\mathcal{S}}_1]\boldsymbol{M}\theta_1}\dots e^{\boldsymbol{M}^{-1}[\boldsymbol{\mathcal{S}}_{n-1}]\boldsymbol{M}\theta_{n-1}}e^{\boldsymbol{M}^{-1}[\boldsymbol{\mathcal{S}}_n]\boldsymbol{M}\theta_n}\\
&= \boldsymbol{M}e^{[\boldsymbol{\mathcal{B}}_1]\theta_1}\dots e^{[\boldsymbol{\mathcal{B}}_{n-1}]\theta_{n-1}}e^{[\boldsymbol{\mathcal{B}}_n]\theta_n}
\end{aligned}$$

$$\begin{cases}[\boldsymbol{\mathcal{B}}_i] = \boldsymbol{M}^{-1}[\boldsymbol{\mathcal{S}}_i]\boldsymbol{M}\\ \boldsymbol{\mathcal{B}}_i = [\mathrm{Ad}_{\boldsymbol{M}^{-1}}]\boldsymbol{\mathcal{S}}_i\end{cases}$$

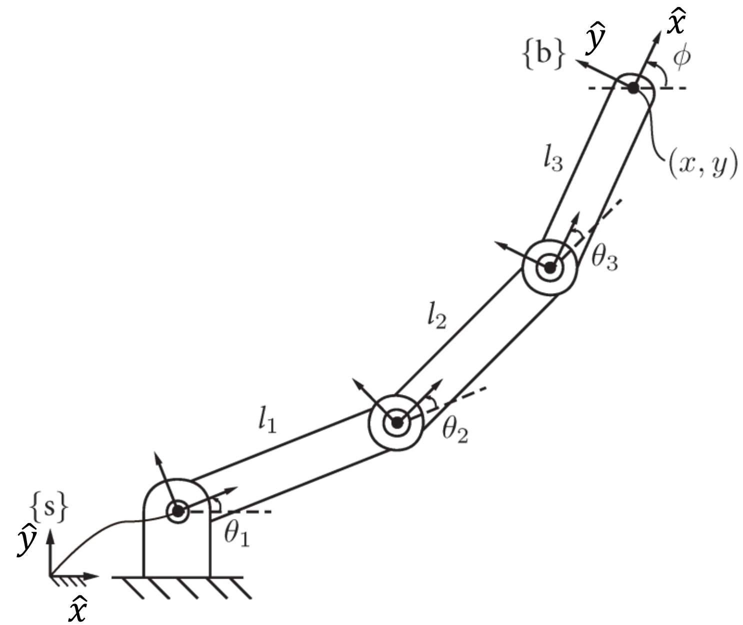


$$\boxed{\boldsymbol{T}_{sb}(\boldsymbol{\theta}) = \boldsymbol{M}e^{[\boldsymbol{\mathcal{B}}_1]\theta_1}\dots e^{[\boldsymbol{\mathcal{B}}_{n-1}]\theta_{n-1}}e^{[\boldsymbol{\mathcal{B}}_n]\theta_n}}$$

The screw axes $\boldsymbol{\mathcal{B}}_1, \dots, \boldsymbol{\mathcal{B}}_n$ expressed in $\{b\}$, corresponding to the joint motions when the robot is at its home/zero configuration ($\boldsymbol{\theta} = \boldsymbol{0}$).

Example: 3R Planar Robot

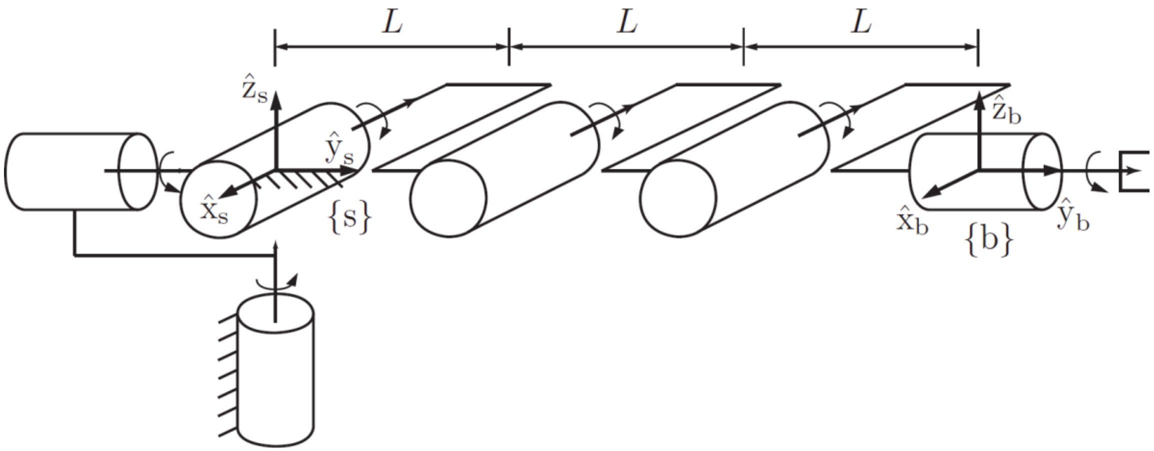
Find the geometric forward kinematics using the **body form** of the PoE formulation.



Example: 6R Spatial Robot

The zero position and the direction of positive rotation for each joint axis are as shown in the figure.

Find the geometric forward kinematics using the **body form** of the PoE formulation.

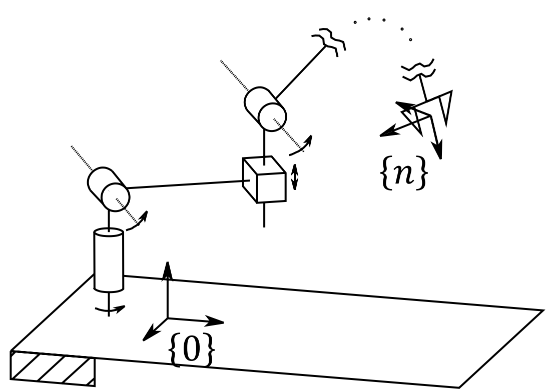


Denavit–Hartenberg (DH) Parameters

Denavit–Hartenberg (DH) Method

The basic idea of Denavit–Hartenberg (DH) method is to attach reference frames to each link of the open chain to derive the forward kinematics from the relative displacements between adjacent link frames.

Consider an n-link open chain manipulator in its home or zero configuration ($\boldsymbol{\theta} = \mathbf{0}$). Attach a frame $\{0\}$ to the base, frames $\{1\}$ to $\{n\}$ to the links 1 to n (end-effector), based on the following rules:



Assigning Link Frames

(Based on Modified DH Method)

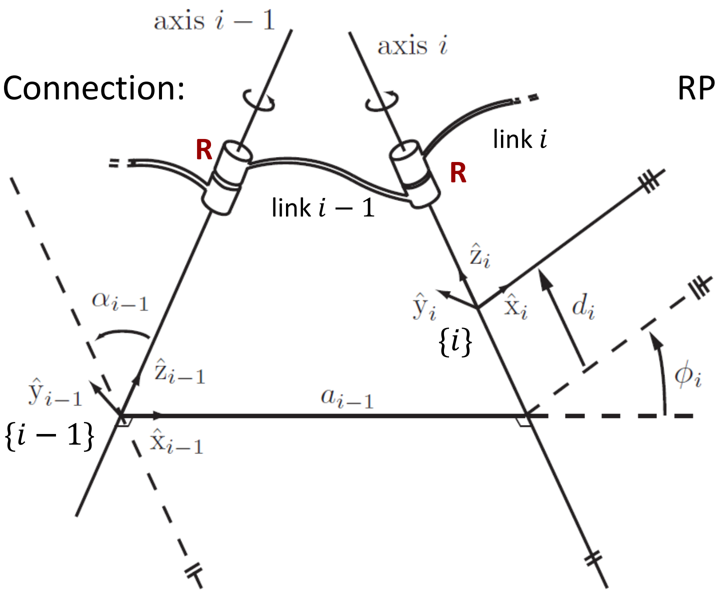
❶ Axis $\hat{z}_{i-1}$ coincides with joint axis $i-1$ and axis $\hat{z}_i$ coincides with joint axis i, along the positive direction of rotation (by the right-hand rule) or translation.

❷ Connect the axes $\hat{z}_{i-1}$ and $\hat{z}_i$ by a mutually perpendicular line (if this line is not unique or fails to exist, refer to the ***Special Cases***). The origin of frame $\{i-1\}$ is then located at the point where this line intersects joint axis $i-1$.

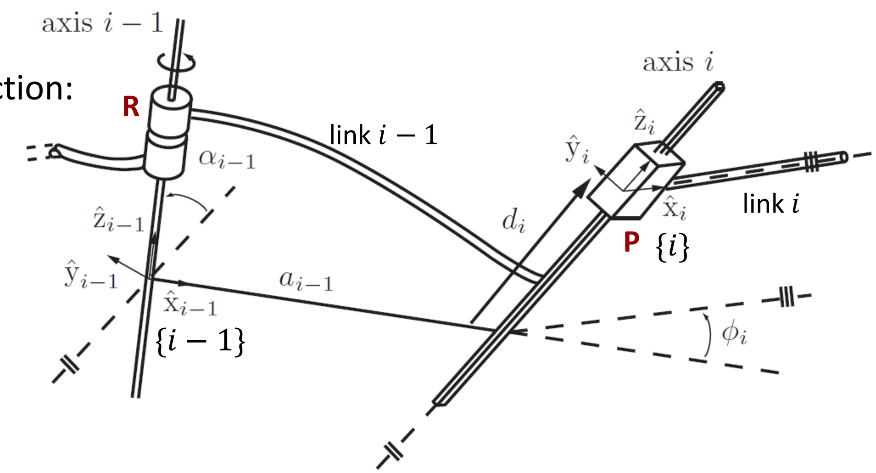
❸ Axis $\hat{x}_{i-1}$ is chosen to be in the direction of the mutually perpendicular line pointing from the axis $\hat{z}_{i-1}$ to the axis $\hat{z}_i$.

❹ $\hat{y}$-axis is determined from $\hat{x} \times \hat{y} = \hat{z}$ to obtain a right-handed frame.

RR Connection:



RP Connection:



Note: For prismatic joint i, the $\hat{z}_i$-direction is chosen to be along the positive direction of translation.

Denavit–Hartenberg (DH) Parameters

(Based on Modified DH Method)

Four DH parameters that exactly specify $\boldsymbol{T}_{i-1,i}$:

- a_{i-1} (link length): Length of the mutually perpendicular line.
- α_{i-1} (link twist): Angle from $\hat{z}_{i-1}$ to $\hat{z}_i$, measured about $\hat{x}_{i-1}$.
- d_i (link offset): Distance from the intersection of $\hat{x}_{i-1}$ and $\hat{z}_i$ to the origin of $\{i\}$ along $\hat{z}_i$.
- ϕ_i (joint angle): Angle from $\hat{x}_{i-1}$ to $\hat{x}_i$, measured about $\hat{z}_i$.

i	α_{i-1}	a_{i-1}	d_i	ϕ_i
1				
2				
3				
$\vdots$				

Note: For a revolute joint ϕ_i acts as the joint variable, and for a prismatic joint d_i acts as the joint variable, and the other 3 parameters are all constant.

Note: The frame $\{0\}$ is chosen to coincide with frame $\{1\}$ in its zero position. Frame $\{n\}$ is attached to a point on the end-effector that makes the description of the task intuitive and/or make as many of the DH parameters as possible zero.

Manipulator Forward Kinematics

(Based on Modified DH Method)

Transporting from $\{i-1\}$ to $\{i\}$:

i. A rotation of $\{i-1\}$ about its $\hat{x}$-axis by α_{i-1}.
ii. A translation of the frame along its $\hat{x}$-axis by a_{i-1}.
iii. A translation of the frame along its $\hat{z}$-axis by d_i.
iv. A rotation of the frame about its $\hat{z}$-axis by ϕ_i.

$$\boldsymbol{T}_{i-1,i} = \mathrm{Rot}(\hat{x}, \alpha_{i-1})\,\mathrm{Trans}(\hat{x}, a_{i-1})\,\mathrm{Trans}(\hat{z}, d_i)\,\mathrm{Rot}(\hat{z}, \phi_i) \in SE(3)$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_{i-1} & -\sin\alpha_{i-1} & 0 \\ 0 & \sin\alpha_{i-1} & \cos\alpha_{i-1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_{i-1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\phi_i & -\sin\phi_i & 0 & 0 \\ \sin\phi_i & \cos\phi_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\phi_i & -\sin\phi_i & 0 & a_{i-1} \\ \sin\phi_i\cos\alpha_{i-1} & \cos\phi_i\cos\alpha_{i-1} & -\sin\alpha_{i-1} & -d_i\sin\alpha_{i-1} \\ \sin\phi_i\sin\alpha_{i-1} & \cos\phi_i\sin\alpha_{i-1} & \cos\alpha_{i-1} & d_i\cos\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Thus, the forward kinematics can be expressed as

$$\boldsymbol{T}_{0n}(\theta_1, \ldots, \theta_n) = \boldsymbol{T}_{01}(\theta_1)\boldsymbol{T}_{12}(\theta_2)\cdots\boldsymbol{T}_{n-1,n}(\theta_n) \in SE(3)$$

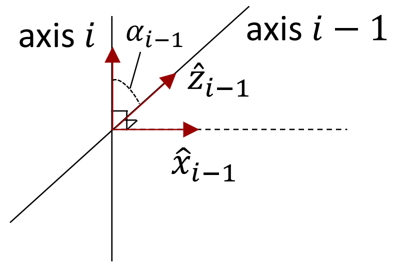
where θ_i is either ϕ_i (for revolute joint) or d_i (for a prismatic joint).

Assigning Link Frames: Special Cases

(Based on Modified DH Method)

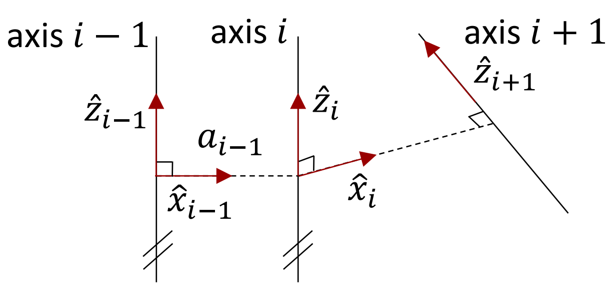
(1) When consecutive joint axes $i-1$ and i intersect (mutually perpendicular line is undefined):

In this case, $a_{i-1}=0$, and we choose $\hat{x}_{i-1}$ to be perpendicular to the plane spanned by axes $i-1$ and i (at intersection). There are two acceptable possibilities: one leads to a positive value of α_{i-1} while the other leads to a negative value.



(2) When consecutive joint axes $i-1$ and i are parallel (mutually perpendicular line is not unique):

In this case there exist many possibilities for a mutually perpendicular line, all of which are valid. Choose the line (and center of $\{i-1\}$) that is the <u>most physically intuitive</u> and that results in as <u>many zero DH parameters</u> as possible.



For instance, in this case $d_i=0$.

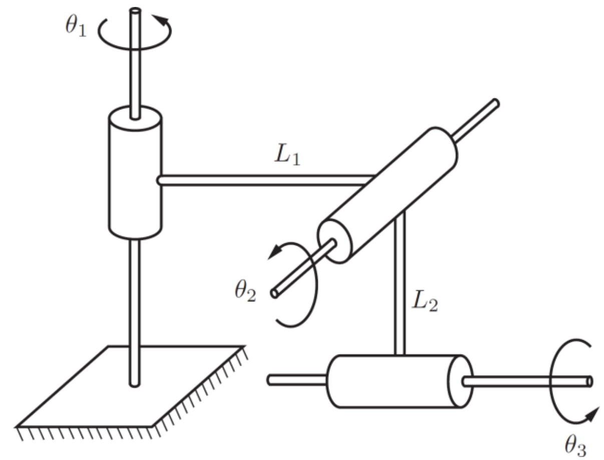
Some Remarks on DH Parameters

- In general, a minimum of six independent parameters are required to describe the relative transformation between two frames $\boldsymbol{T}_{i-1,i}$ in space (3 for the orientation and 3 for the position).
- In DH parameter representation, a minimum of four parameters are required for each transformation $\boldsymbol{T}_{i-1,i}$ (i.e., for an n-DOF open-chain robot, $4n$ DH parameters are sufficient to completely describe the forward kinematics).
- The reduction in the number of parameters is due to <u>carefully assigning link frames</u> based on some rules (where only rotations and translations along the $\hat{x}$- and $\hat{z}$-axes are allowed). If the link reference frames are assigned in arbitrary fashion, then more parameters are required.
- The interpretation of the parameters in the PoE is natural and intuitive, however, it needs more parameters than the DH representation ($6n$ parameters to describe the n screw axes, in addition to the n joint values).
- Given $\boldsymbol{T}_{s0}$ and $\boldsymbol{T}_{nb}$, we can compute $\boldsymbol{T}_{sb} = \boldsymbol{T}_{s0}\boldsymbol{T}_{0n}\boldsymbol{T}_{nb}$ that yields the pose of the end-effector frame $\{b\}$ with respect to the base frame $\{s\}$.

Examples

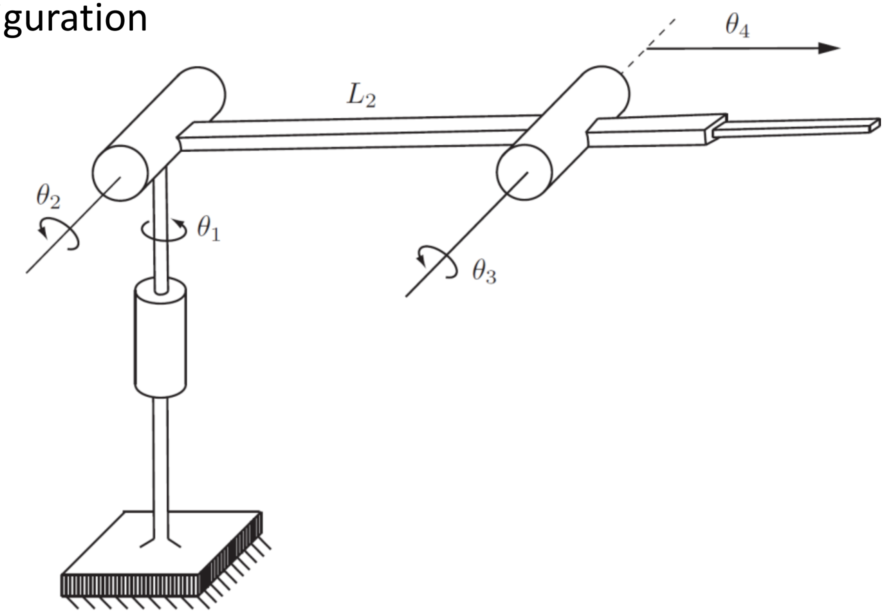
Example 1: A 3R spatial open chain in its zero configuration.

i	α_{i-1}	a_{i-1}	d_i	ϕ_i
1				
2				
3				



Example 2: A spatial RRRP open chain in its zero configuration

i	α_{i-1}	a_{i-1}	d_i	ϕ_i
1				
2				
3				
4				



Redundancy

Intrinsic and Kinematic Redundancy in Open-Chain Robot Manipulators

For an open-chain manipulator, we can define:

$\dim(\text{C–Space}) = \dim(\text{J–Space}) = n$ (Configuration Space or Joint Space)

$\dim(\text{O–Space}) = m$ (Operational Space) $[m = 3$ for planar, $m = 6$ for spatial$]$

$\dim(\text{T–Space}) = r$ (Task Space) $[r \leq m]$

- A manipulator is **intrinsically redundant** when the dimension of the joint space is greater than the dimension of the operational space (i.e., $n > m$). Ex.: 4R Planar Robot $(n = 4, m = 3)$

- A manipulator is **kinematically redundant** when the dimension of the joint space is greater than the dimension of the task space (i.e, $n > r$) and there exist $n - r$ redundant DOFs (or degrees of redundancy (DOR)).

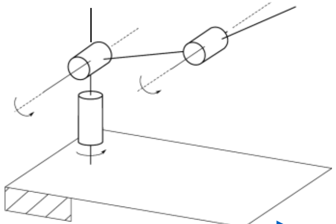
Note: A manipulator can be redundant with respect to a task and nonredundant with respect to another.

Ex: 3R Planar Robot
$$\begin{cases} n = m = 3, r = 2 \quad (x, y) & \text{redundant} \\ n = m = r = 3 \quad (x, y, \phi) & \text{nonredundant} \end{cases}$$

Ex: 3R Spatial Robot

$n = m = r = 3\ (x, y, z)$

nonredundant



- Redundancy can provide the manipulator with dexterity and versatility in its motion. Thus, it is possible to avoid obstacles in the workspace or to optimize some objective function such as minimizing the motor power needed to hold the end-effector at that configuration.

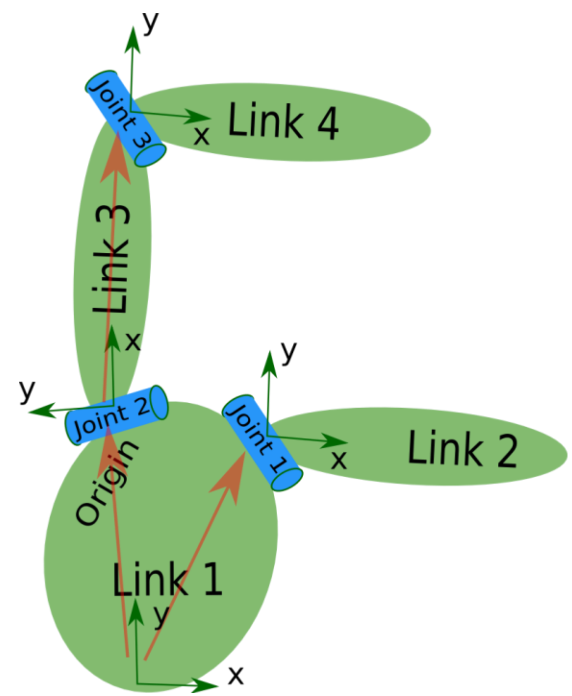
Universal Robot Description Format (URDF)

Universal Robot Description Format

The Universal Robot Description Format (URDF) is an XML (eXtensible Markup Language) file format used by the Robot Operating System (ROS) to describe the kinematics (in defining joints), inertial properties, and link geometry of robots (in defining links) of open-chain robots in home/zero configuration.

```
<joint name="joint1" type="continuous">
        <parent link="link1"/>
        <child link="link2"/>
        <origin xyz="0.5 0.3 0" rpy="0 0 0" />
        <axis xyz="-0.9 0.15 0" />
</joint>

<link name="link1">
        <inertial>
                <mass value="1"/>
                <origin rpy="0.1 0 0" xyz="0 0 0"/>
                <inertia ixx="0.004" ixy="0" ixz="0"
                        iyy="0.004" iyz="0" izz="0.007"/>
        </inertial>
</link>
```



(Home Configuration)

URL

URDF: Defining Joints

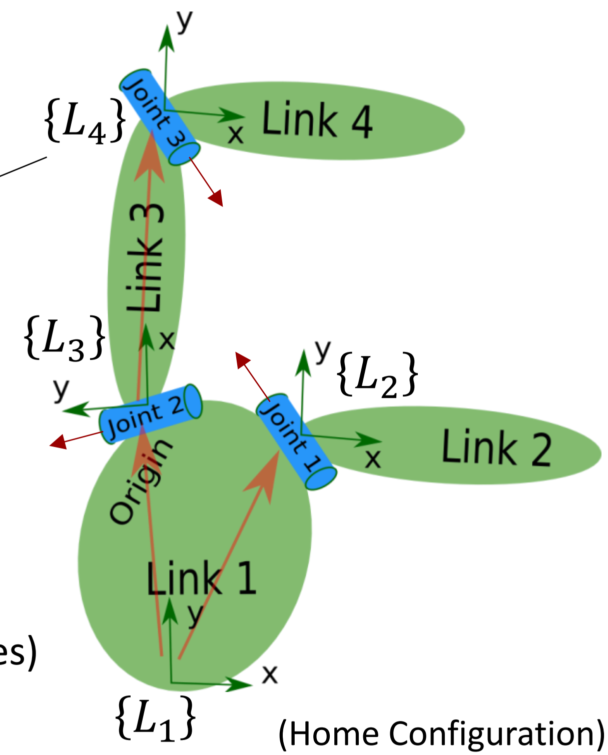
Joints connect two links: a parent link and a child link. The reference frame of each (child) link $\{L_i\}$ is located (at the bottom of the link) on the joint's axis.

(for example)

```
<joint name="joint3" type="continuous">
        <parent link="link3"/>
        <child link="link4"/>
        <origin xyz="0.5 0 0" rpy="0 0 -1.57" />
        <axis xyz="0.707 -0.707 0" />
</joint>
```

$: \{L_4\}$ w.r.t. $\{L_3\}$

$:$ in $\{L_4\}$



(Home Configuration)

"rpy" Roll–Pitch–Yaw Angles (XYZ about fixed frames)

"origin" defines the pose of the child link frame relative to the parent link frame when the joint variable is zero.

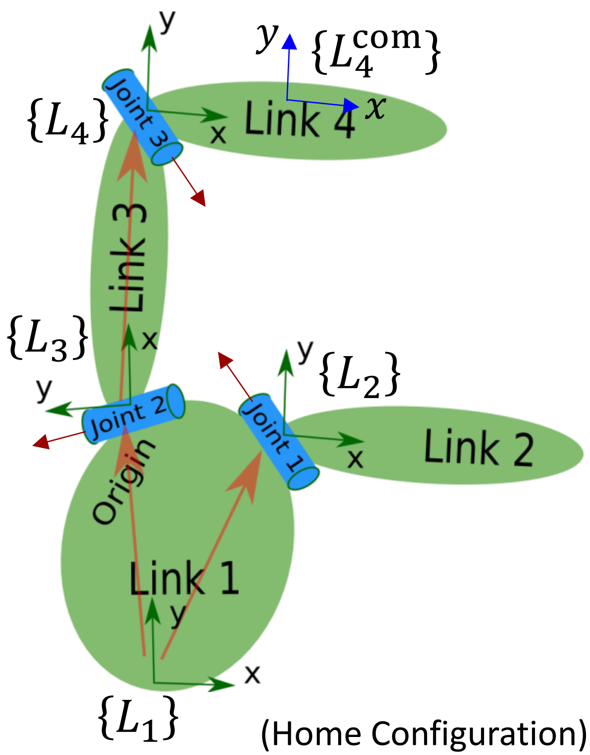
"axis" defines the joint's axis, a <u>unit vector</u> expressed in the child link's frame, in the direction of <u>positive rotation</u> for a revolute joint or <u>positive translation</u> for a prismatic joint.

URDF: Defining Links

```
<link name=" link4">
    <inertial>
        <mass value="1"/>
        <origin xyz="0.1 0 0" rpy="0 0 0"/>
        <inertia ixx="0.004" ixy="0" ixz="0"
            iyy="0.004" iyz="0" izz="0.007"/>
    </inertial>
    <visual>
        <geometry>
            <mesh filename="…/link1.stl" />
        </geometry>
        <material name="DarkGrey">
            <color rgba="0.3 0.3 0.3 1.0"/>
        </material>
    </visual>
</link>
```

$\boxed{:\ \{L_4^{\text{com}}\}\ \text{w.r.t.}\ \{L_4\}}$

$\boxed{:\ \text{in}\ \{L_4^{\text{com}}\}}$



(Home Configuration)

"inertia" defines six elements of inertia matrix relative to a frame at the link's center of mass.

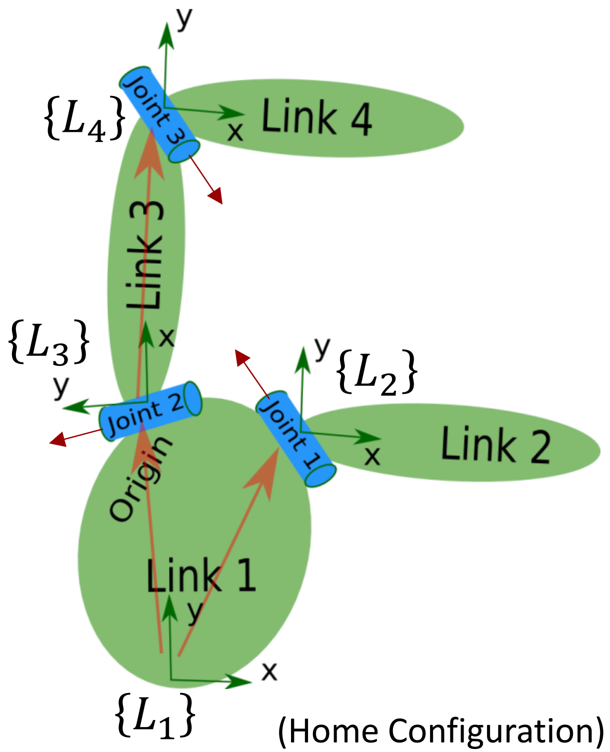
"origin" defines the position and orientation of a frame at the link's center of mass (COM) relative to the link's frame at its joint.

URDF

```
<robot name="test_robot">
  <link name="link1" />
  <link name="link2" />
  <link name="link3" />
  <link name="link4" />

  <joint name="joint1" type="continuous">
    <parent link="link1"/>
    <child link="link2"/>
    <origin xyz="5 3 0" rpy="0 0 0" />
    <axis xyz="-0.9 0.15 0" />
  </joint>

  <joint name="joint2" type="continuous">
    <parent link="link1"/>
    <child link="link3"/>
    <origin xyz="-2 5 0" rpy="0 0 1.57" />
    <axis xyz="-0.707 0.707 0" />
  </joint>
```

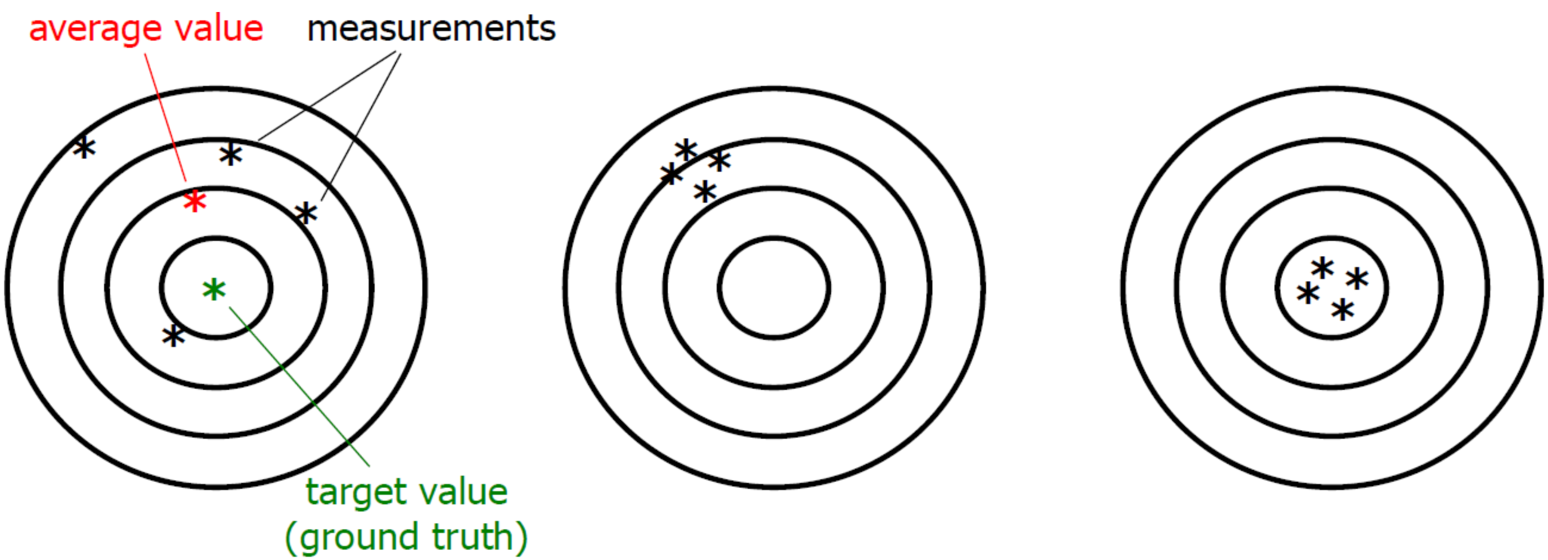


(Home Configuration)

```
  <joint name="joint3" type="continuous">
    <parent link="link3"/>
    <child link="link4"/>
    <origin xyz="5 0 0" rpy="0 0 -1.57" />
    <axis xyz="0.707 -0.707 0" />
  </joint>
</robot>
```

Kinematic Calibration

Accuracy and Repeatability



low accuracy low repeatability	low accuracy high repeatability	high accuracy high repeatability

better components!

calibration!

Kinematic Calibration Using DH Parameters

Kinematic Calibration techniques are used to find accurate estimates of DH parameters for forward kinematics from a series of precise measurements on the manipulator's end-effector pose to improve manipulator accuracy.

By defining the $n \times 1$ vectors of DH parameters for the whole structure as

$$\boldsymbol{a} = [a_1 \dots a_n]^T, \qquad \boldsymbol{\alpha} = [\alpha_1 \dots \alpha_n]^T, \qquad \boldsymbol{d} = [d_1 \dots d_n]^T, \qquad \boldsymbol{\theta} = [\theta_1 \dots \theta_n]^T,$$

the forward kinematics equation can be rewritten by emphasizing the dependence on DH parameters as

$$\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{\theta}) \in \mathbb{R}^r \quad \rightarrow \quad \boldsymbol{x} = \boldsymbol{f}(\boldsymbol{a}, \boldsymbol{\alpha}, \boldsymbol{d}, \boldsymbol{\theta}) \in \mathbb{R}^r \quad (1)$$

(for simplicity, suppose all joints are revolute)

Let $\boldsymbol{x}_m$ be the measured pose. We are now interested in solving the following where $(\boldsymbol{a}, \boldsymbol{\alpha}, \boldsymbol{d}, \boldsymbol{\theta})$ are unknowns.

$$\boldsymbol{x}_m = \boldsymbol{f}(\boldsymbol{a}, \boldsymbol{\alpha}, \boldsymbol{d}, \boldsymbol{\theta}) \qquad \text{or} \qquad \underset{\boldsymbol{a},\boldsymbol{\alpha},\boldsymbol{d},\boldsymbol{\theta}}{\text{minimize}} \ \|\boldsymbol{x}_m - \boldsymbol{f}(\boldsymbol{a}, \boldsymbol{\alpha}, \boldsymbol{d}, \boldsymbol{\theta})\|_2^2$$

A numerical method to solve nonlinear least squares problem is using Gauss–Newton method after linearization.

Linearization

On the assumption of small deviations and using first-order Taylor expansion in (1):

$$\Delta \boldsymbol{x} = \boldsymbol{x}_m - \boldsymbol{x}_{\mathrm{nom}} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{a}} \Delta \boldsymbol{a} + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\alpha}} \Delta \boldsymbol{\alpha} + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{d}} \Delta \boldsymbol{d} + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}} \Delta \boldsymbol{\theta} \qquad (2)$$

measured pose

A measure of accuracy at the given configuration

Nominal pose computed with the nominal values, i.e., $\boldsymbol{x}_{\mathrm{nom}} = \boldsymbol{f}(\boldsymbol{a}_{\mathrm{nom}}, \boldsymbol{\alpha}_{\mathrm{nom}}, \boldsymbol{d}_{\mathrm{nom}}, \boldsymbol{\theta}_{\mathrm{nom}})$

First-order variations between the real values and the nominal ones $((\cdot)_{\mathrm{real}} - (\cdot)_{\mathrm{nom}})$

Partial derivatives ($r \times n$ Jacobian matrices) evaluated in nominal values of $\boldsymbol{a}_{\mathrm{nom}}, \boldsymbol{\alpha}_{\mathrm{nom}}, \boldsymbol{d}_{\mathrm{nom}}, \boldsymbol{\theta}_{\mathrm{nom}}$

Note: The nominal values for the fixed parameters $\boldsymbol{a}_{\mathrm{nom}}, \boldsymbol{\alpha}_{\mathrm{nom}}, \boldsymbol{d}_{\mathrm{nom}}$ are set equal to the design data of the mechanical structure and for the joint variables $\boldsymbol{\theta}_{\mathrm{nom}}$ are set equal to the data provided by the position sensors (encoders) at the given manipulator configuration.

Linearization

By defining the concatenated vector $\boldsymbol{\zeta} = [\boldsymbol{a}^T, \boldsymbol{\alpha}^T, \boldsymbol{d}^T, \boldsymbol{\theta}^T] \in \mathbb{R}^{4n}$ and the **Kinematic Calibration Matrix** $\boldsymbol{\Phi}(\boldsymbol{\zeta}_{\text{nom}}) = [\partial \boldsymbol{f}/\partial \boldsymbol{a}, \partial \boldsymbol{f}/\partial \boldsymbol{\alpha}, \partial \boldsymbol{f}/\partial \boldsymbol{d}, \partial \boldsymbol{f}/\partial \boldsymbol{\theta}] \in \mathbb{R}^{r\times 4n}$ computed at the nominal values of the parameters $\boldsymbol{\zeta}_{\text{nom}}$, (2) can be compactly rewritten as

$$\boldsymbol{x}_m - \boldsymbol{x}_{\text{nom}} = \boldsymbol{\Phi}(\boldsymbol{\zeta}_{\text{nom}})(\boldsymbol{\zeta}_{\text{real}} - \boldsymbol{\zeta}_{\text{nom}}) \quad \text{or} \quad \boxed{\Delta\boldsymbol{x} = \boldsymbol{\Phi}(\boldsymbol{\zeta}_{\text{nom}})\underbrace{\Delta\boldsymbol{\zeta}}_{\text{unknowns}}} \qquad (3)$$

Since this constitutes a system of r equations and $4n$ unknowns with $r < 4n$, a sufficient number (say l) of different end-effector pose measurements $\boldsymbol{x}_{m,i}$ $(i = 1, \dots, l)$ must be performed to obtain a system of at least $4n$ equations:

$$\begin{bmatrix} \boldsymbol{x}_{m,1} - \boldsymbol{f}(\boldsymbol{\zeta}_{\text{nom},1}) \\ \vdots \\ \boldsymbol{x}_{m,l} - \boldsymbol{f}(\boldsymbol{\zeta}_{\text{nom},l}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}(\boldsymbol{\zeta}_{\text{nom},1}) \\ \vdots \\ \boldsymbol{\Phi}(\boldsymbol{\zeta}_{\text{nom},l}) \end{bmatrix} \Delta\boldsymbol{\zeta} \quad \text{or} \quad \boxed{\overline{\Delta\boldsymbol{x}} = \overline{\boldsymbol{\Phi}}\Delta\boldsymbol{\zeta}}$$

$\overline{\Delta\boldsymbol{x}} \in \mathbb{R}^{rl}$
$\overline{\boldsymbol{\Phi}} \in \mathbb{R}^{lr\times 4n}$
$\Delta\boldsymbol{\zeta} \in \mathbb{R}^{4n}$

Regressor Matrix is full column rank for sufficiently large l, i.e., $lr \gg 4n$.

Note: $\boldsymbol{\zeta}_{\text{nom},i} = [\boldsymbol{a}_{\text{nom}}^T, \boldsymbol{\alpha}_{\text{nom}}^T, \boldsymbol{d}_{\text{nom}}^T, \boldsymbol{\theta}_{\text{nom},i}^T]$, i.e., the geometric parameters are constant whereas the joint variables depend on the manipulator configuration at pose i.

Gauss–Newton Iteration for Linearized Least Squares Problem

Since the original problem is a nonlinear parameter estimate problem in the unknowns, after linearization, the procedure should be iterated until $\Delta\boldsymbol{\zeta}$ converges.

a) Initialization: Given $\boldsymbol{x}_m = (\boldsymbol{x}_{m,1}, \ldots, \boldsymbol{x}_{m,l})$, $\boldsymbol{z}_{\text{nom}} = (\boldsymbol{\zeta}_{\text{nom},1}, \ldots, \boldsymbol{\zeta}_{\text{nom},l}) \in \mathbb{R}^{4nl}$,

- Set $\boldsymbol{z}^{(0)} = \boldsymbol{z}_{\text{nom}}$
- Set $\overline{\boldsymbol{\Phi}}^{(0)} = \overline{\boldsymbol{\Phi}}(\boldsymbol{z}^{(0)})$
- Set $\overline{\Delta\boldsymbol{x}}^{(0)} = \boldsymbol{x}_m - \boldsymbol{f}(\boldsymbol{z}^{(0)})$
- Set $k = 0$

b) Iteration: While $\left\|\overline{\Delta\boldsymbol{x}}^{(k)}\right\| > \epsilon$ for some small $\epsilon \in \mathbb{R}$ and $k < k_{\text{MaxIteration}}$:

- Set $\Delta\boldsymbol{\zeta}^{(k)} = \left(\overline{\boldsymbol{\Phi}}^{(k)}\right)^{+}\overline{\Delta\boldsymbol{x}}^{(k)} = \underbrace{\left(\overline{\boldsymbol{\Phi}}^{(k)^T}\overline{\boldsymbol{\Phi}}^{(k)}\right)^{-1}\overline{\boldsymbol{\Phi}}^{(k)^T}}_{\text{left pseudo-inverse matrix of } \overline{\boldsymbol{\Phi}}^{(k)}}\overline{\Delta\boldsymbol{x}}^{(k)} \quad \Leftrightarrow \quad \min_{\Delta\boldsymbol{\zeta}} \left\|\overline{\boldsymbol{\Phi}}^{(k)}\Delta\boldsymbol{\zeta} - \overline{\Delta\boldsymbol{x}}^{(k)}\right\|_2^2$
- Set $\boldsymbol{z}^{(k+1)} = \boldsymbol{z}^{(k)} + \Delta\boldsymbol{z}^{(k)}$
- Set $\overline{\boldsymbol{\Phi}}^{(k+1)} = \overline{\boldsymbol{\Phi}}(\boldsymbol{z}^{(k+1)})$
- Set $\overline{\Delta\boldsymbol{x}}^{(k+1)} = \boldsymbol{x}_m - \boldsymbol{f}(\boldsymbol{z}^{(k+1)})$
- Set $k = k + 1$

$\Delta\boldsymbol{z}^{(k)} = \left(\Delta\boldsymbol{\zeta}^{(k)}, \ldots, \Delta\boldsymbol{\zeta}^{(k)}\right) \in \mathbb{R}^{4nl}$

c) Return: $\boldsymbol{z}^{(k)}, \Delta\boldsymbol{\zeta}^{(k-1)}$

As a result, an accurate estimates of the nominal geometric parameters $(\boldsymbol{a}_{\text{nom}}, \boldsymbol{\alpha}_{\text{nom}}, \boldsymbol{d}_{\text{nom}})$ as well as possible corrections on the encoders measurements $(\Delta\boldsymbol{\theta})$ are obtained.