Physics-Based Animation

Kinematics

9 februari 2022

Kinematics

Basic Principles

- Study of motion of rigid bodies neglecting forces
- Forward Kinematics: End effector position from joint coordinates
- Inverse Kinematics: Joint coordinates from end effector position

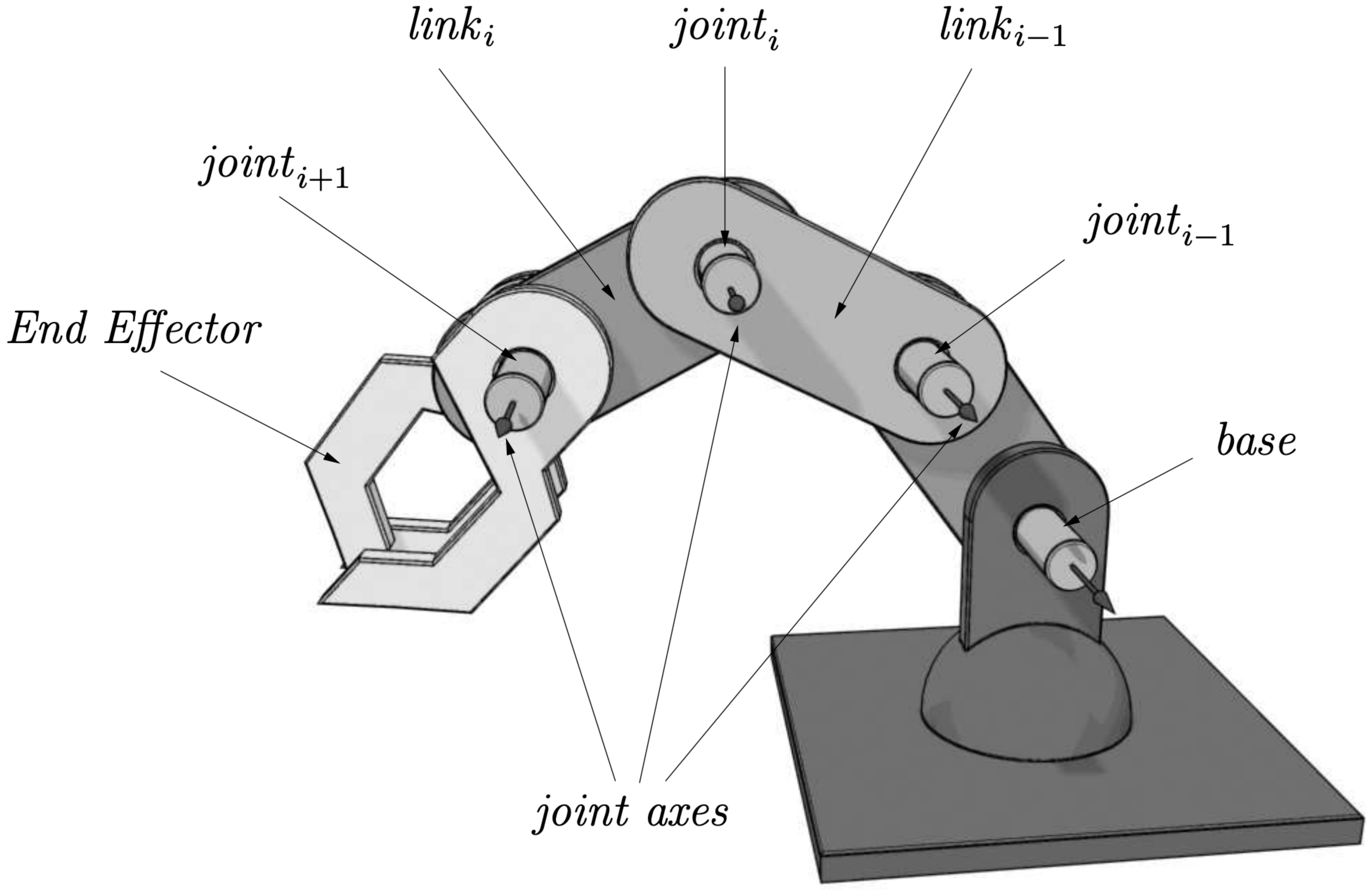
End Effector

- Last coordinate frame of an articulated figure

- Two expressions for the end effector transform:

$$^{0}\boldsymbol{T}_N = {}^{0}\boldsymbol{T}_N(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) = {}^{0}\boldsymbol{T}_1(\boldsymbol{\theta}_1)\,{}^{1}\boldsymbol{T}_2(\boldsymbol{\theta}_2) \cdots {}^{(N-1)}\boldsymbol{T}_N(\boldsymbol{\theta}_N) \tag{3.1a}$$

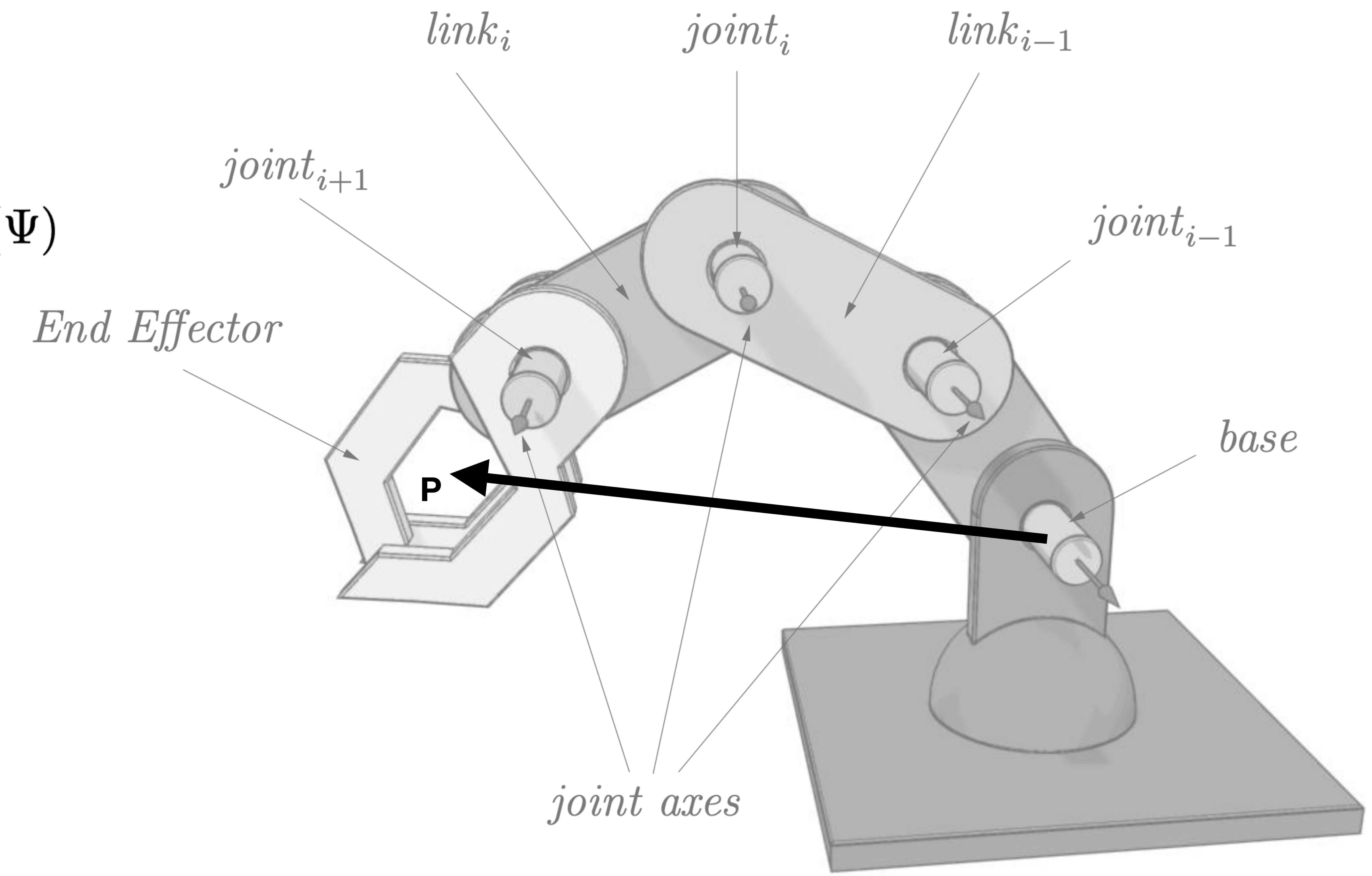
$$^{0}\boldsymbol{T}_N = {}^{0}\boldsymbol{T}_N(\boldsymbol{p}, \Phi, \Theta, \Psi) = \boldsymbol{T}(\boldsymbol{p})\boldsymbol{R}_z(\Phi)\boldsymbol{R}_y(\Theta)\boldsymbol{R}_x(\Psi) \tag{3.2}$$



End Effector

$$^{0}\boldsymbol{T}_N = {}^{0}\boldsymbol{T}_N(\boldsymbol{p}, \Phi, \Theta, \Psi) = \boldsymbol{T}(\boldsymbol{p})\boldsymbol{R}_z(\Phi)\boldsymbol{R}_y(\Theta)\boldsymbol{R}_x(\Psi)$$

$$\boldsymbol{T}(\boldsymbol{p}) = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



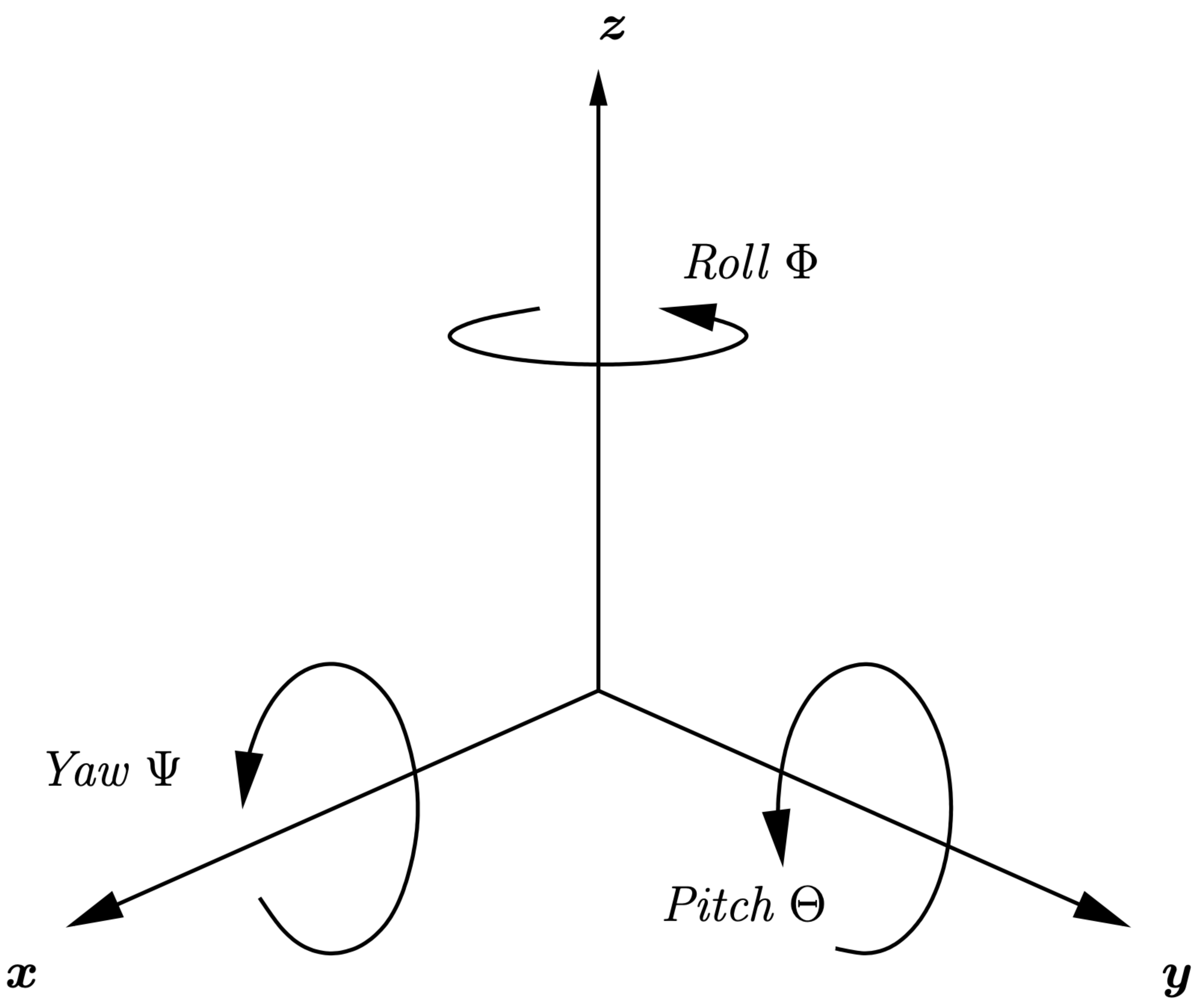
End Effector

$$ {}^0\boldsymbol{T}_N = {}^0\boldsymbol{T}_N(\boldsymbol{p}, \Phi, \Theta, \Psi) = \boldsymbol{T}(\boldsymbol{p})\boldsymbol{R}_z(\Phi)\boldsymbol{R}_y(\Theta)\boldsymbol{R}_x(\Psi) $$

$$ \boldsymbol{Y}(\Psi) = \boldsymbol{R}_x(\Psi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\Psi & -\sin\Psi & 0 \\ 0 & \sin\Psi & \cos\Psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ \boldsymbol{P}(\Theta) = \boldsymbol{R}_y(\Theta) = \begin{bmatrix} \cos\Theta & 0 & \sin\Theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\Theta & 0 & \cos\Theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ \boldsymbol{R}(\Phi) = \boldsymbol{R}_z(\Phi) = \begin{bmatrix} \cos\Phi & -\sin\Phi & 0 & 0 \\ \sin\Phi & \cos\Phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$



First:	Yaw	Ψ	Rotate the angle Ψ around the $\boldsymbol{x}$-axis
Second:	Pitch	Θ	Rotate the angle Θ around the $\boldsymbol{y}$-axis
Third:	Roll	Φ	Rotate the angle Φ around the $\boldsymbol{z}$-axis

Table 3.1: Order, name, and definition of the rotations.

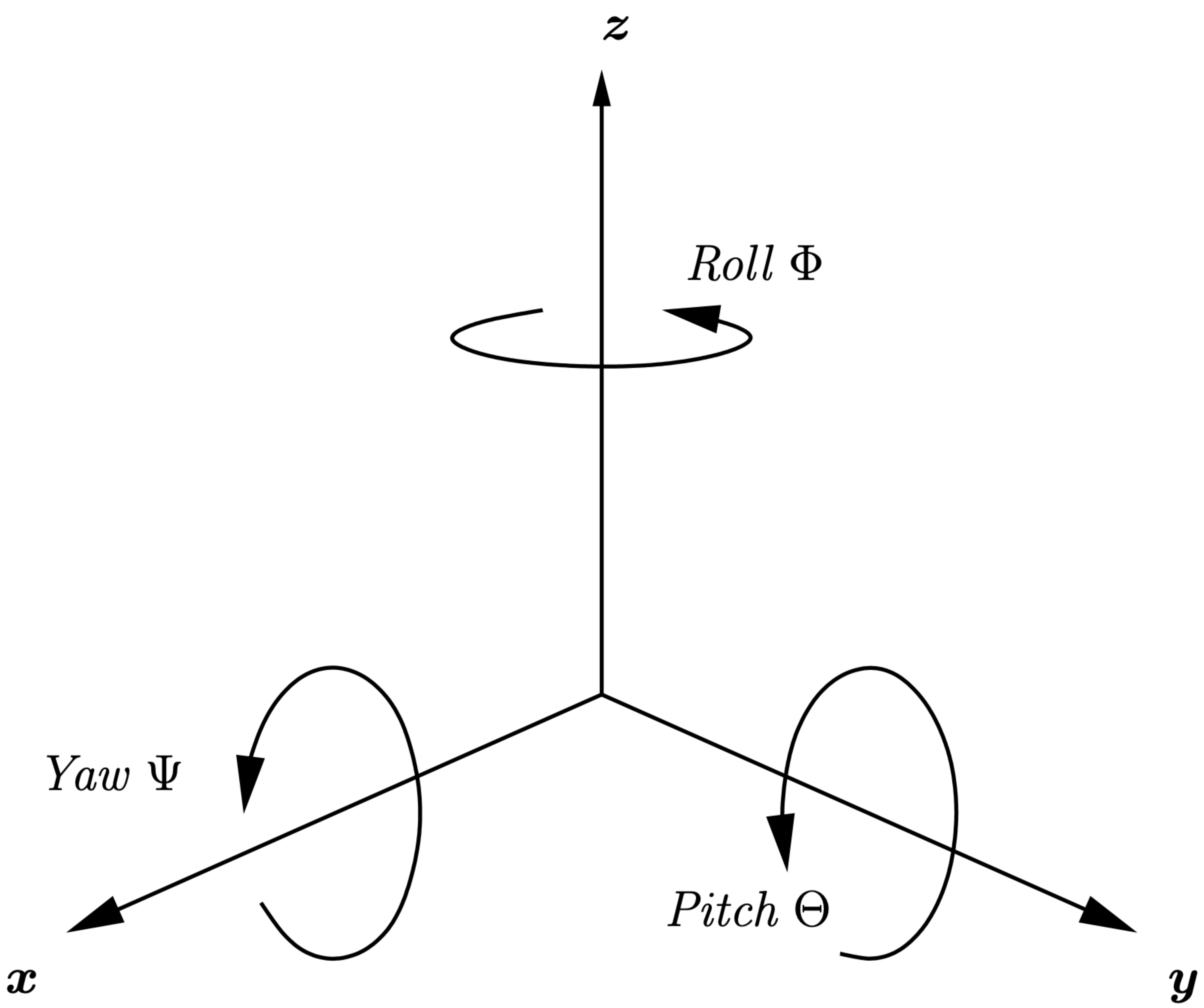
End Effector

$$^{0}\boldsymbol{T}_N = {}^{0}\boldsymbol{T}_N(\boldsymbol{p}, \Phi, \Theta, \Psi) = \boldsymbol{T}(\boldsymbol{p})\boldsymbol{R}_z(\Phi)\boldsymbol{R}_y(\Theta)\boldsymbol{R}_x(\Psi)$$

$$\boldsymbol{T}_{RPY}(\Phi, \Theta, \Psi) = \boldsymbol{R}(\Phi)\boldsymbol{P}(\Theta)\boldsymbol{Y}(\Psi) = \boldsymbol{R}_z(\Phi)\boldsymbol{R}_y(\Theta)\boldsymbol{R}_x(\Psi)$$

$$= \begin{bmatrix} c\Phi & -s\Phi & 0 & 0 \\ s\Phi & c\Phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\Theta & 0 & s\Theta & 0 \\ 0 & 1 & 0 & 0 \\ -s\Theta & 0 & c\Theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\Psi & -s\Psi & 0 \\ 0 & s\Psi & c\Psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} c\Phi c\Theta & c\Phi s\Theta s\Psi - s\Phi c\Psi & c\Phi s\Theta c\Psi + s\Phi s\Psi & 0 \\ s\Phi c\Theta & s\Phi s\Theta s\Psi + c\Phi c\Psi & s\Phi s\Theta c\Psi - c\Phi s\Psi & 0 \\ -s\Theta & c\Theta s\Psi & c\Theta c\Psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



First:	Yaw	Ψ	Rotate the angle Ψ around the $\boldsymbol{x}$-axis
Second:	Pitch	Θ	Rotate the angle Θ around the $\boldsymbol{y}$-axis
Third:	Roll	Φ	Rotate the angle Φ around the $\boldsymbol{z}$-axis

Table 3.1: Order, name, and definition of the rotations.

End Effector

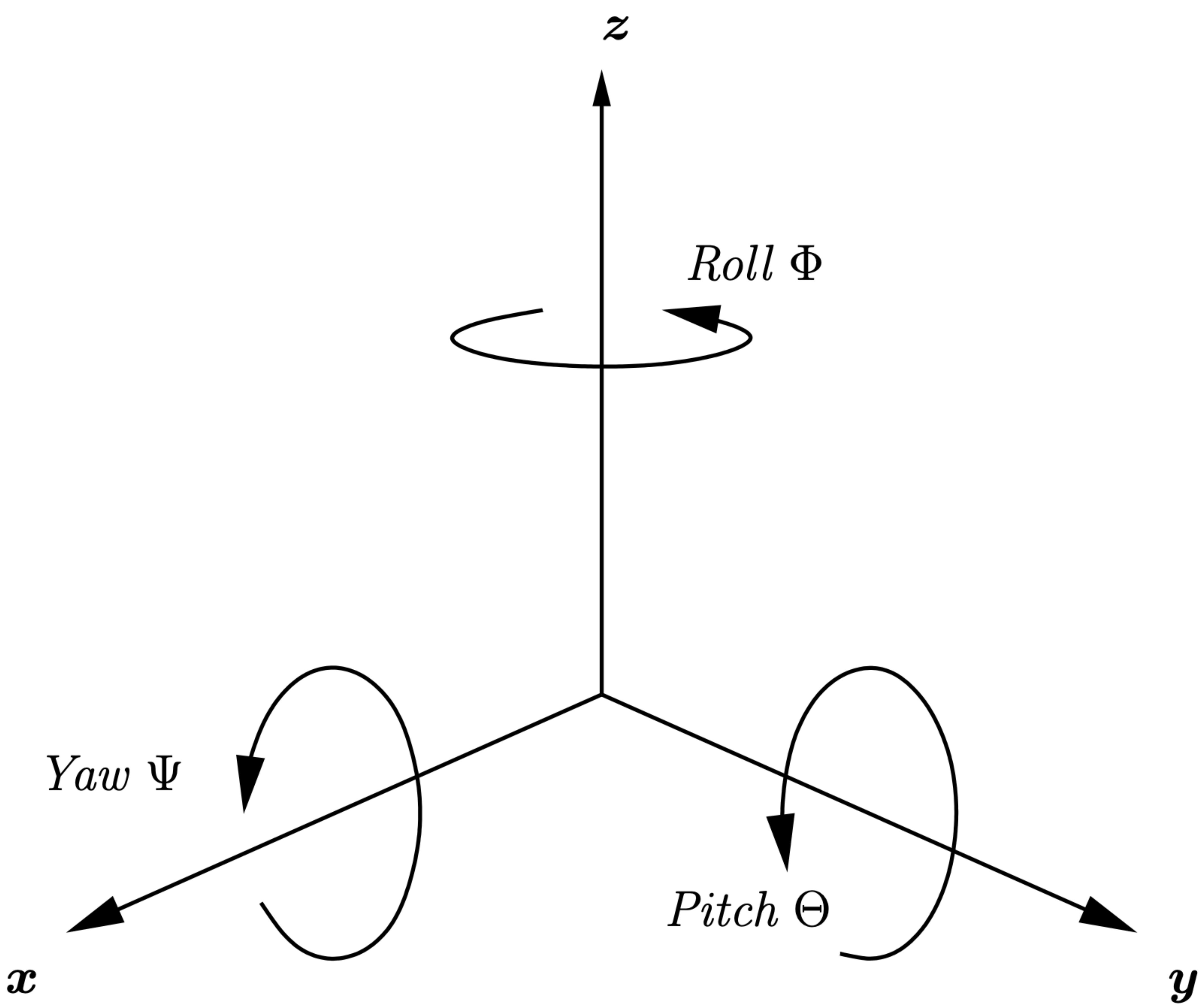
$$^0\boldsymbol{T}_N = {}^0\boldsymbol{T}_N(\boldsymbol{p}, \Phi, \Theta, \Psi) = \boldsymbol{T}(\boldsymbol{p})\boldsymbol{R}_z(\Phi)\boldsymbol{R}_y(\Theta)\boldsymbol{R}_x(\Psi)$$

$$^0\boldsymbol{T}_N = {}^0\boldsymbol{T}_N(\boldsymbol{p}, \Phi, \Theta, \Psi) = \boldsymbol{T}(\boldsymbol{p})\boldsymbol{T}_{RPY}(\Phi, \Theta, \Psi)$$

$$= \begin{bmatrix} c\Phi c\Theta & c\Phi s\Theta s\Psi - s\Phi c\Psi & c\Phi s\Theta c\Psi + s\Phi s\Psi & p_x \\ s\Phi c\Theta & s\Phi s\Theta s\Psi + c\Phi c\Psi & s\Phi s\Theta c\Psi - c\Phi s\Psi & p_y \\ -s\Theta & c\Theta s\Psi & c\Theta c\Psi & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- State vector:

$$\boldsymbol{s} = \begin{bmatrix} Xposition \\ Yposition \\ Zposition \\ Yaw \\ Pitch \\ Roll \end{bmatrix} = \begin{bmatrix} p_x \\ p_y \\ p_z \\ \Psi \\ \Theta \\ \Phi \end{bmatrix} = \begin{bmatrix} p_x(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) \\ p_y(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) \\ p_z(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) \\ \Psi(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) \\ \Theta(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) \\ \Phi(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) \end{bmatrix}$$



First:	Yaw	Ψ	Rotate the angle Ψ around the $\boldsymbol{x}$-axis
Second:	Pitch	Θ	Rotate the angle Θ around the $\boldsymbol{y}$-axis
Third:	Roll	Φ	Rotate the angle Φ around the $\boldsymbol{z}$-axis

Table 3.1: Order, name, and definition of the rotations.

Forward Kinematics

- Given know joint parameters, calculate the end effector state vector:
  - $$\boldsymbol{s}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) = \begin{bmatrix} p_x(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) \\ p_y(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) \\ p_z(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) \\ \Psi(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) \\ \Theta(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) \\ \Phi(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) \end{bmatrix}$$
- Solution is well defined:
  - $${}^0\boldsymbol{T}_N = {}^0\boldsymbol{T}_N(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) = {}^0\boldsymbol{T}_1(\boldsymbol{\theta}_1)\,{}^1\boldsymbol{T}_2(\boldsymbol{\theta}_2) \cdots {}^{(N-1)}\boldsymbol{T}_N(\boldsymbol{\theta}_N) \qquad (3.1a)$$

  $\boldsymbol{s}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N)$ can be found from ${}^0\boldsymbol{T}_N$ using the methods in section 3.14

Inverse Kinematics

- Given a goal end effector state, $\boldsymbol{s}_g$, calculate joint parameters $\boldsymbol{\theta}_i, \quad i = 1, \ldots, N$

- $\boldsymbol{f}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) = \boldsymbol{s}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) - \boldsymbol{s}_g = 0.$

- No single unique solution, must be found iteratively —> Optimisation

Inverse Kinematics

Taylor series (for detail see chapter 20)

- Through Taylor expansion we can approximate $\boldsymbol{f}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N)$ as:

$$\boldsymbol{f}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) \approx -\frac{\partial \boldsymbol{f}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N)}{\partial(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N)} \Delta(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N)$$

$$\Delta(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) = \begin{bmatrix} \boldsymbol{\theta}_1 \\ \vdots \\ \boldsymbol{\theta}_N \end{bmatrix}_{new} - \begin{bmatrix} \boldsymbol{\theta}_1 \\ \vdots \\ \boldsymbol{\theta}_N \end{bmatrix}$$

- And since $\boldsymbol{s}_g$ is a constant:

$$\boldsymbol{s}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) - \boldsymbol{s}_g \approx -\frac{\partial \boldsymbol{s}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N)}{\partial(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N)} \left(\begin{bmatrix} \boldsymbol{\theta}_1 \\ \vdots \\ \boldsymbol{\theta}_N \end{bmatrix}_{new} - \begin{bmatrix} \boldsymbol{\theta}_1 \\ \vdots \\ \boldsymbol{\theta}_N \end{bmatrix} \right).$$

Inverse Kinematics

The regular case

- Define the Jacobian matrix

$$J(\theta) = \frac{\partial f(\theta)}{\partial \theta} = \frac{\partial s(\theta)}{\partial \theta}$$

- Then:

$$s(\theta) - s_g = -J(\theta)(\theta_{new} - \theta) \qquad (3.36^*)$$

- And iff J is invertible, iteratively:

$$\theta_{new} = \theta - J(\theta)^{-1}(s(\theta) - s_g)$$

Inverse Kinematics

The over determined case

$$s(\theta) - s_g = -J(\theta)(\theta_{new} - \theta) \qquad (3.36^*)$$

- The Jacobian has more rows than columns —> Not invertible!
- Pseudo inverse:

$$\boldsymbol{J}^{+} = \left(\boldsymbol{J}^{T}\boldsymbol{J}\right)^{-1}\boldsymbol{J}^{T}$$

- Left multiplying J in both sides of equation 3.36 and rearranging yields:

$$\theta_{new} = \theta - J^{+}(\theta)(s(\theta) - s_g)$$

Inverse Kinematics

The under determined case

$$s(\theta) - s_g = -J(\theta)(\theta_{new} - \theta) \qquad (3.36^*)$$

- The Jacobian has more columns than rows —> Not invertible!
- Pseudo inverse:

$$\boldsymbol{J}^{+} = \boldsymbol{J}^{T}\left(\boldsymbol{J}\boldsymbol{J}^{T}\right)^{-1}.$$

- Right multiplying J in both sides of equation 3.36 and rearranging yields:

$$\theta_{new} = \theta - J^{+}(\theta)(s(\theta) - s_g)$$

Inverse Kinematics

Computing Jacobians

$$J(\theta) = \begin{array}{cc} & k \longrightarrow \\ j \Big\downarrow & \begin{bmatrix} \frac{\partial p_x}{\partial \boldsymbol{\theta}_1}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) & & \frac{\partial p_x}{\partial \boldsymbol{\theta}_N}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) \\ \frac{\partial p_y}{\partial \boldsymbol{\theta}_1}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) & \cdots & \frac{\partial p_y}{\partial \boldsymbol{\theta}_N}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) \\ \frac{\partial p_z}{\partial \boldsymbol{\theta}_1}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) & \cdots & \frac{\partial p_z}{\partial \boldsymbol{\theta}_N}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) \\ \frac{\partial \Psi}{\partial \boldsymbol{\theta}_1}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) & \cdots & \frac{\partial \Psi}{\partial \boldsymbol{\theta}_N}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) \\ \frac{\partial \Theta}{\partial \boldsymbol{\theta}_1}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) & \cdots & \frac{\partial \Theta}{\partial \boldsymbol{\theta}_N}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) \\ \frac{\partial \Phi}{\partial \boldsymbol{\theta}_1}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) & \cdots & \frac{\partial \Phi}{\partial \boldsymbol{\theta}_N}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) \end{bmatrix} \end{array}$$

$$\frac{\partial \left({}^0\boldsymbol{T}_N(\boldsymbol{\theta}_i)\right)_{kj}}{\partial \boldsymbol{\theta}_i}$$

Inverse Kinematics

Computing Jacobians

- Recall: $${}^{0}\boldsymbol{T}_N = {}^{0}\boldsymbol{T}_N(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) = {}^{0}\boldsymbol{T}_1(\boldsymbol{\theta}_1)\,{}^{1}\boldsymbol{T}_2(\boldsymbol{\theta}_2) \cdots {}^{(N-1)}\boldsymbol{T}_N(\boldsymbol{\theta}_N)$$
- Then we can define for some joint i < N:

$$\boldsymbol{P} = {}^{0}\boldsymbol{T}_{(i-1)}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{i-1}) = {}^{0}\boldsymbol{T}_1(\boldsymbol{\theta}_1) \cdots {}^{(i-2)}\boldsymbol{T}_i(\boldsymbol{\theta}_{i-1})$$
$$\boldsymbol{C} = {}^{i}\boldsymbol{T}_N(\boldsymbol{\theta}_{(i+1)}, \ldots, \boldsymbol{\theta}_N) = {}^{i}\boldsymbol{T}_{(i+1)}(\boldsymbol{\theta}_{(i+1)}) \cdots {}^{(N-1)}\boldsymbol{T}_N(\boldsymbol{\theta}_N)$$

- So that:

$${}^{0}\boldsymbol{T}_N(\boldsymbol{\theta}_i) = \boldsymbol{P}\left({}^{(i-1)}\boldsymbol{T}_i(\boldsymbol{\theta}_i)\right)\boldsymbol{C}$$

$$\frac{\partial\left({}^{0}\boldsymbol{T}_N(\boldsymbol{\theta}_i)\right)_{kj}}{\partial\boldsymbol{\theta}_i} = \boldsymbol{P}_{k*}\left(\frac{\partial\left({}^{(i-1)}\boldsymbol{T}_i(\boldsymbol{\theta}_i)\right)}{\partial\boldsymbol{\theta}_i}\right)\boldsymbol{C}_{*j}$$

Inverse Kinematics

Computing Jacobians

$$\frac{\partial \left({}^{0}\boldsymbol{T}_N(\boldsymbol{\theta}_i) \right)_{kj}}{\partial \boldsymbol{\theta}_i} = \boldsymbol{P}_{k*} \left(\frac{\partial \left({}^{(i-1)}\boldsymbol{T}_i(\boldsymbol{\theta}_i) \right)}{\partial \boldsymbol{\theta}_i} \right) \boldsymbol{C}_{*j}$$

$$\begin{bmatrix} p_x(\boldsymbol{\theta}_i) \\ p_y(\boldsymbol{\theta}_i) \\ p_z(\boldsymbol{\theta}_i) \\ \Psi(\boldsymbol{\theta}_i) \\ \Theta(\boldsymbol{\theta}_i) \\ \Phi(\boldsymbol{\theta}_i) \end{bmatrix} = \begin{bmatrix} m_{14}(\boldsymbol{\theta}_i) \\ m_{24}(\boldsymbol{\theta}_i) \\ m_{34}(\boldsymbol{\theta}_i) \\ \arctan\left(\dfrac{m_{32}(\boldsymbol{\theta}_i)}{m_{33}(\boldsymbol{\theta}_i)} \right) \\ \arctan\left(\dfrac{-m_{31}(\boldsymbol{\theta}_i)}{m_{32}(\boldsymbol{\theta}_i) \sin \Psi(\boldsymbol{\theta}_i) + m_{33}(\boldsymbol{\theta}_i) \cos \Psi(\boldsymbol{\theta}_i)} \right) \\ \arctan\left(\dfrac{m_{21}(\boldsymbol{\theta}_i)}{m_{11}(\boldsymbol{\theta}_i)} \right) \end{bmatrix}$$

$$J(\theta) = \begin{bmatrix} \frac{\partial p_x}{\partial \boldsymbol{\theta}_1}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) & \cdots & \frac{\partial p_x}{\partial \boldsymbol{\theta}_N}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) \\ \frac{\partial p_y}{\partial \boldsymbol{\theta}_1}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) & \cdots & \frac{\partial p_y}{\partial \boldsymbol{\theta}_N}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) \\ \frac{\partial p_z}{\partial \boldsymbol{\theta}_1}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) & \cdots & \frac{\partial p_z}{\partial \boldsymbol{\theta}_N}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) \\ \frac{\partial \Psi}{\partial \boldsymbol{\theta}_1}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) & \cdots & \frac{\partial \Psi}{\partial \boldsymbol{\theta}_N}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) \\ \frac{\partial \Theta}{\partial \boldsymbol{\theta}_1}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) & \cdots & \frac{\partial \Theta}{\partial \boldsymbol{\theta}_N}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) \\ \frac{\partial \Phi}{\partial \boldsymbol{\theta}_1}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) & \cdots & \frac{\partial \Phi}{\partial \boldsymbol{\theta}_N}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) \end{bmatrix}$$

Inverse Kinematics

Computing Jacobians (Denavit-Hartenberg)

- Prismatic joint:

$$\frac{\partial\left({}^{(i-1)}\boldsymbol{T}_i(\boldsymbol{\theta}_i)\right)}{\partial d_i} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\sin\alpha_{i-1} \\ 0 & 0 & 0 & \cos\alpha_{i-1} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

- Revolute joint:

$$\frac{\partial\left({}^{(i-1)}\boldsymbol{T}_i(\boldsymbol{\theta}_i)\right)}{\partial d_i} = \begin{bmatrix} -\sin\varphi_i & -\cos\varphi_i & 0 & 0 \\ \cos\alpha_{i-1}\cos\varphi_i & -\cos\alpha_{i-1}\sin\varphi_i & 0 & 0 \\ \sin\alpha_{i-1}\cos\varphi_i & -\sin\alpha_{i-1}\sin\varphi_i & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$