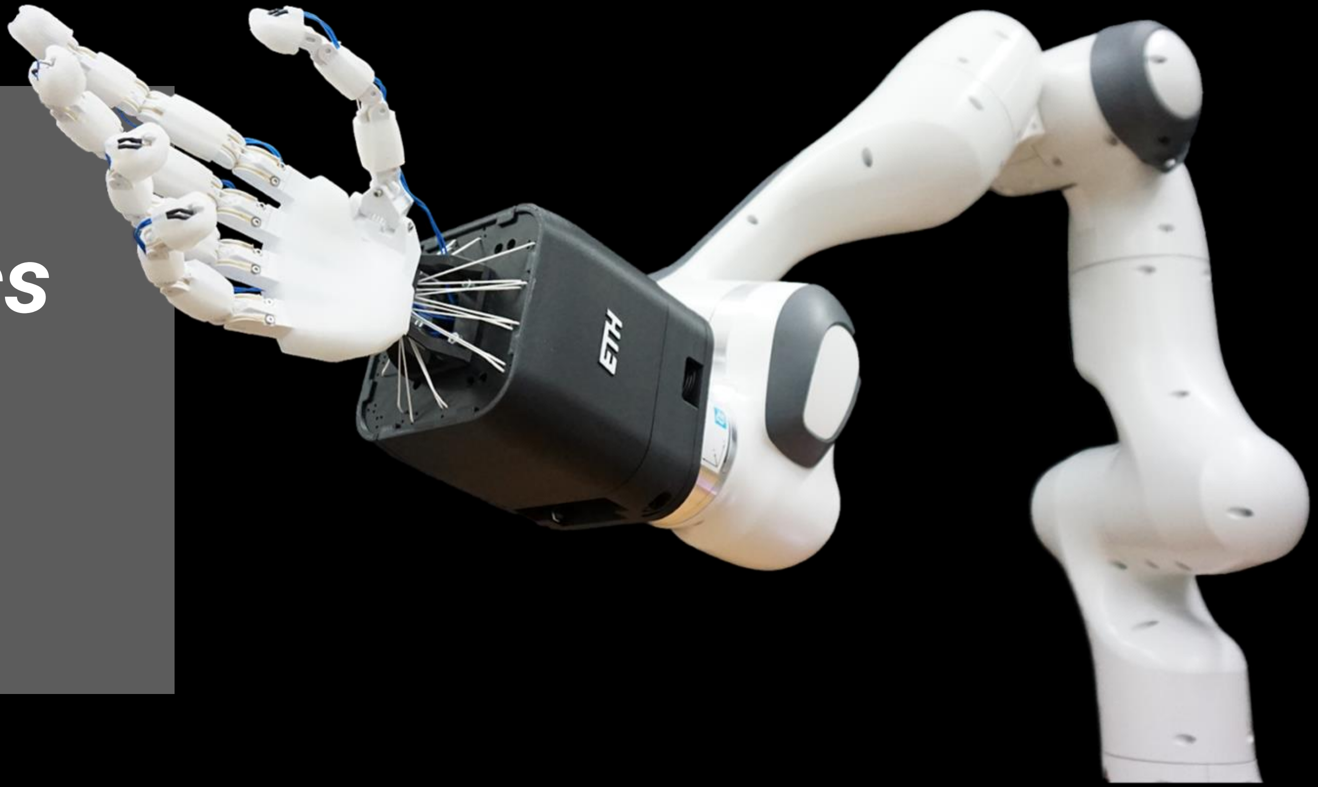
ETH zürich

Modelling the Robot Through Robot Kinematics and Dynamics

Robert Katzschmann

Assistant Professor of Robotics, Soft Robotics Lab



Faive Robotics

SoftRobotics Laboratory

Last Tutorial

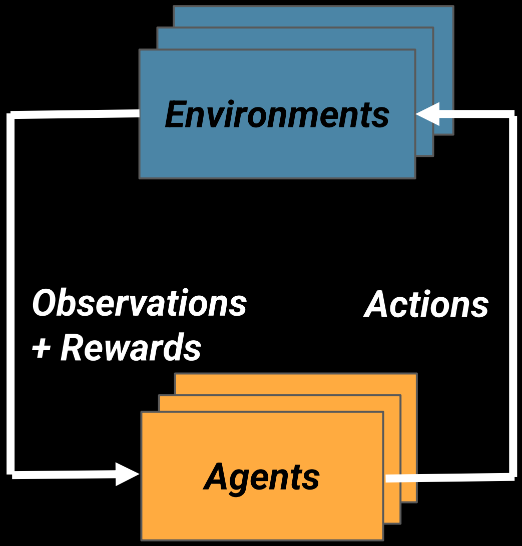
SOFA

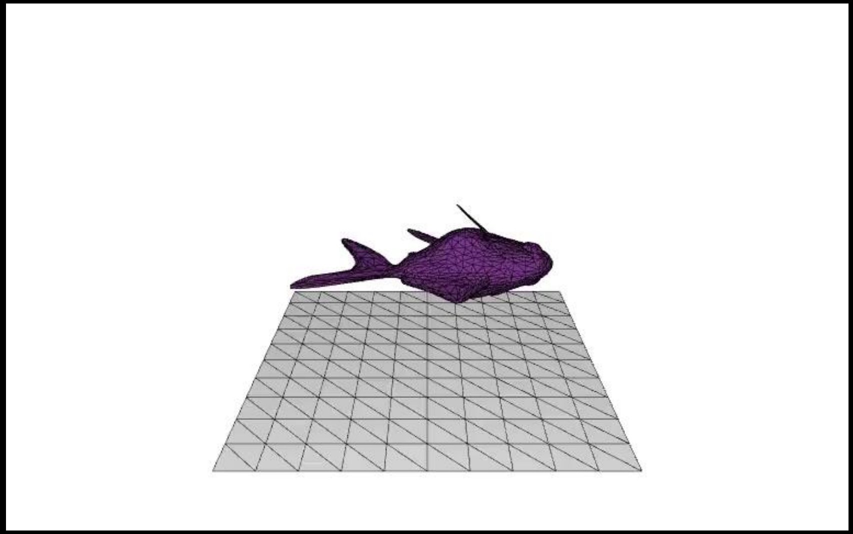
NVIDIA ISAAC

GAZEBO

MuJoCo

DRAKE



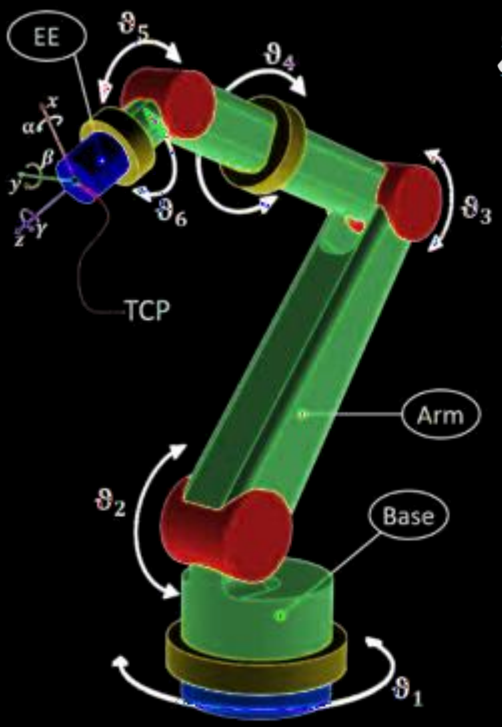


Last Tutorial

USA Toyz

How does it move?

Joint angles → end effector orientation?

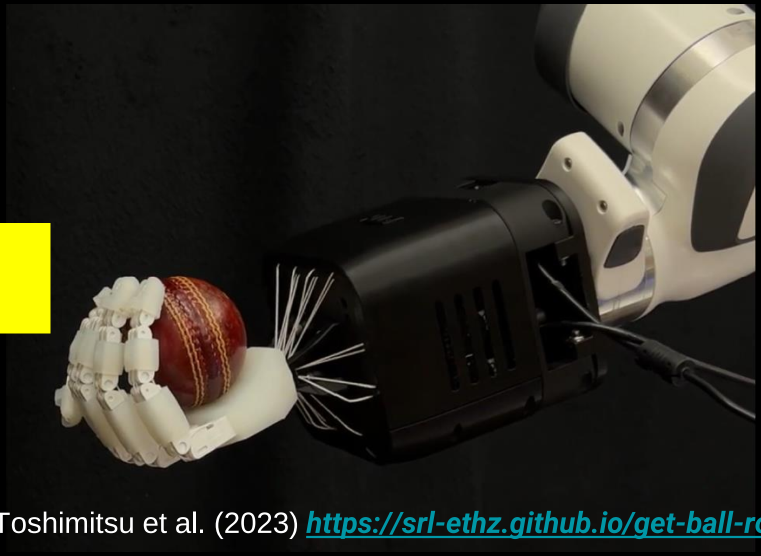


researchgate.net

Robot Kinematics & Robot Dynamics

Robotis

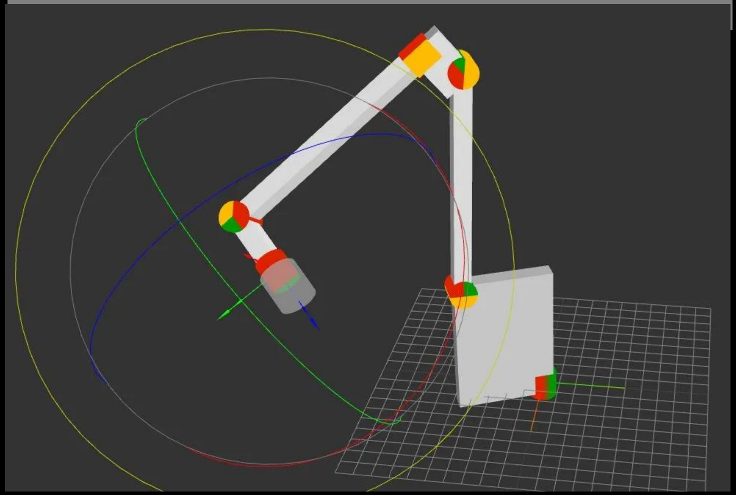
Motor ↔ Fingertips
Input? Force?



Toshimitsu et al. (2023) https://srl-ethz.github.io/get-ball-rolling/

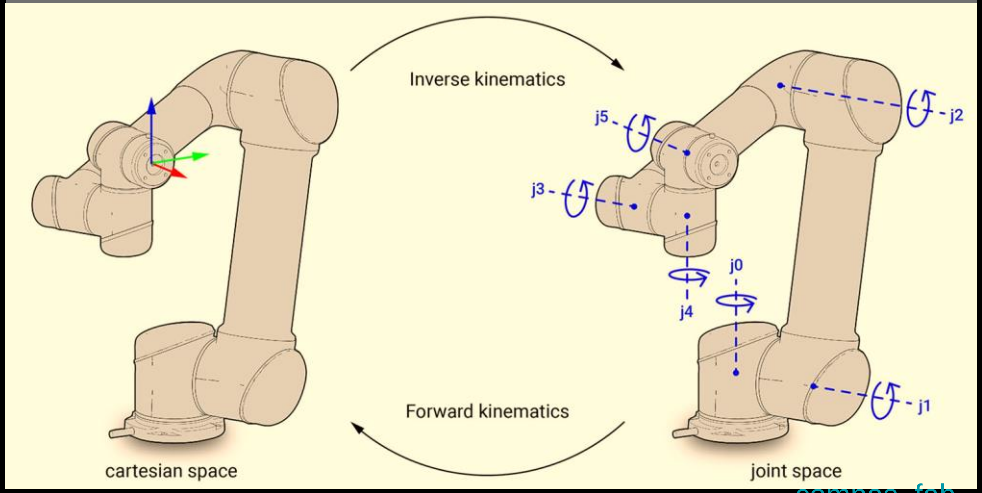
Plan for Today

1. Robot Kinematics and Dynamics



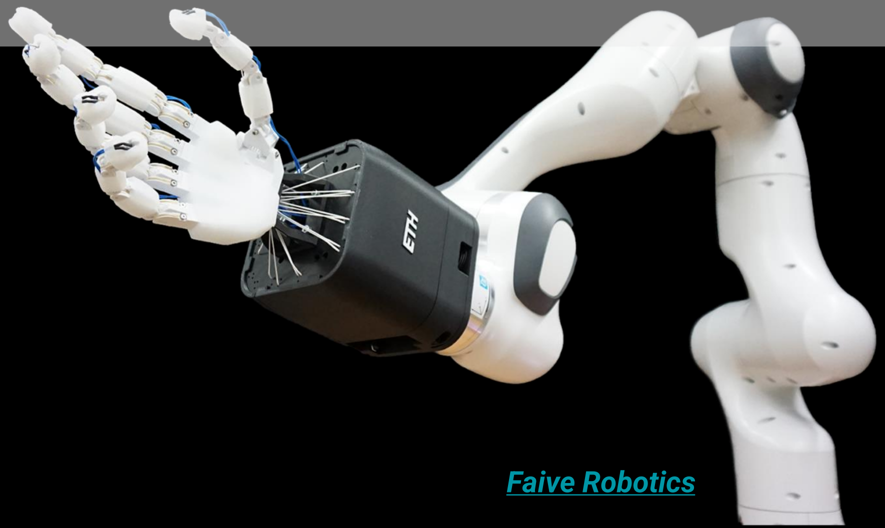
Marginally Clever Robots

2. Forward and Inverse Kinematics



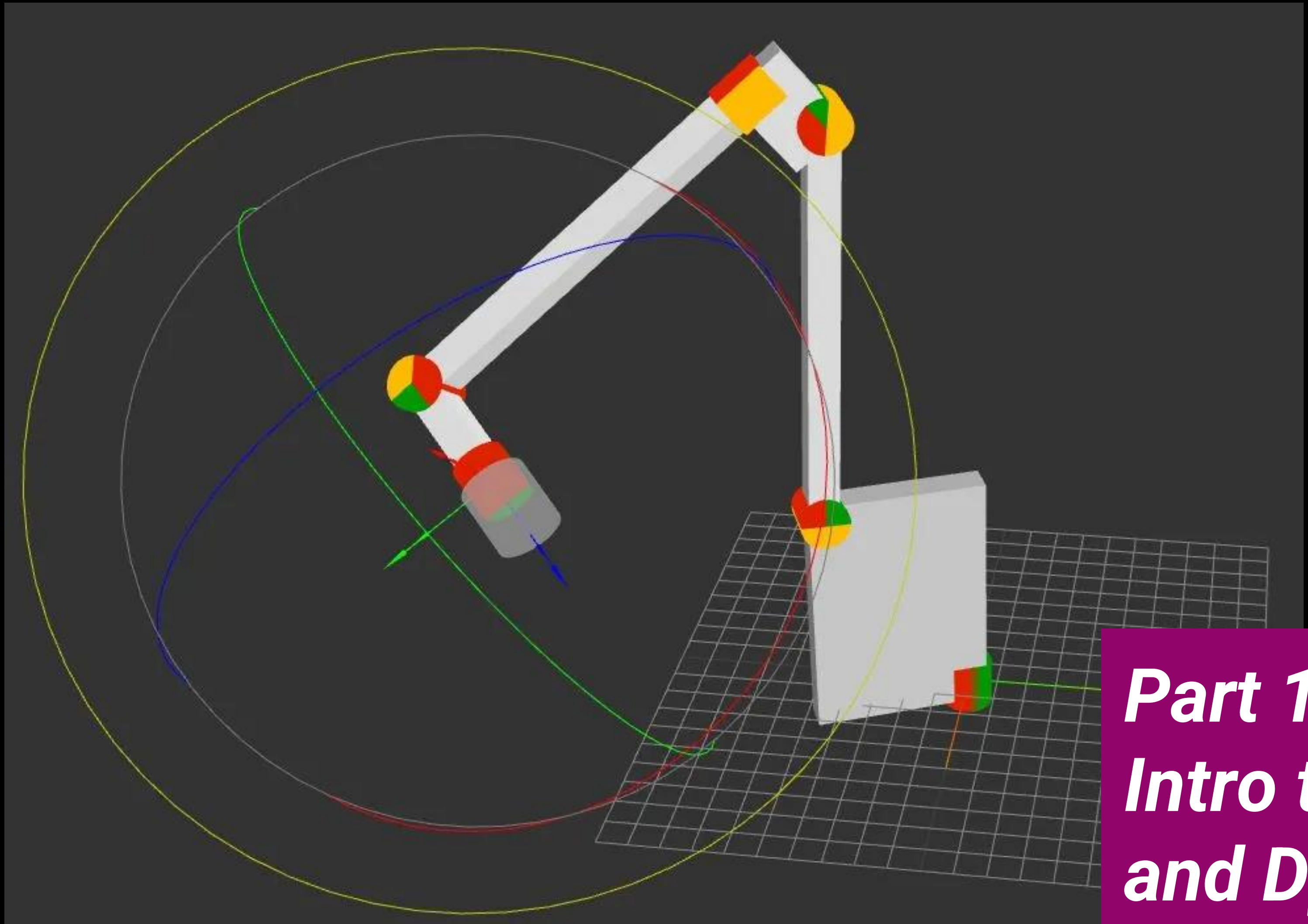
compas_fab

3. Kinematics and Dynamics for hand joints



Faive Robotics

ETH zürich

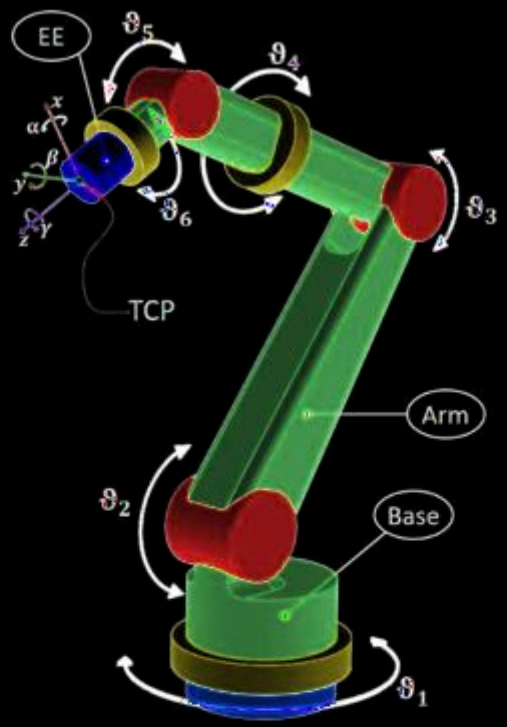


Marginally Clever Robots

Part 1: Intro to Robot Kinematics and Dynamics

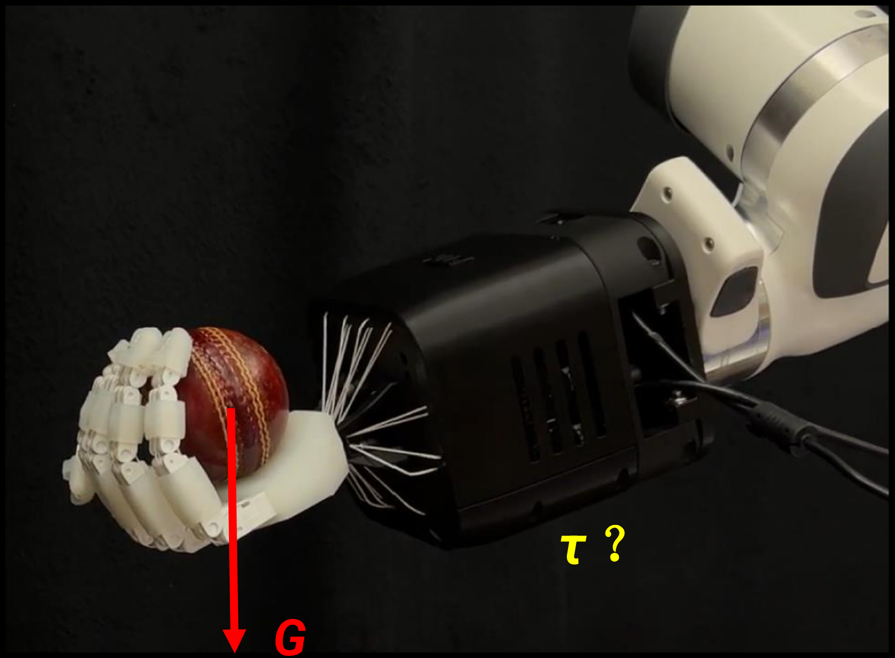
SoftRobotics Laboratory

Robot Kinematics and Dynamics



researchgate.net

Kinematics



Toshimitsu et al. (2023) https://srl-ethz.github.io/get-ball-rolling/

Dynamics

Simulation
reaction to certain actuator commands

Control
invert of simulation, if I want to get somewhere, what command to give?

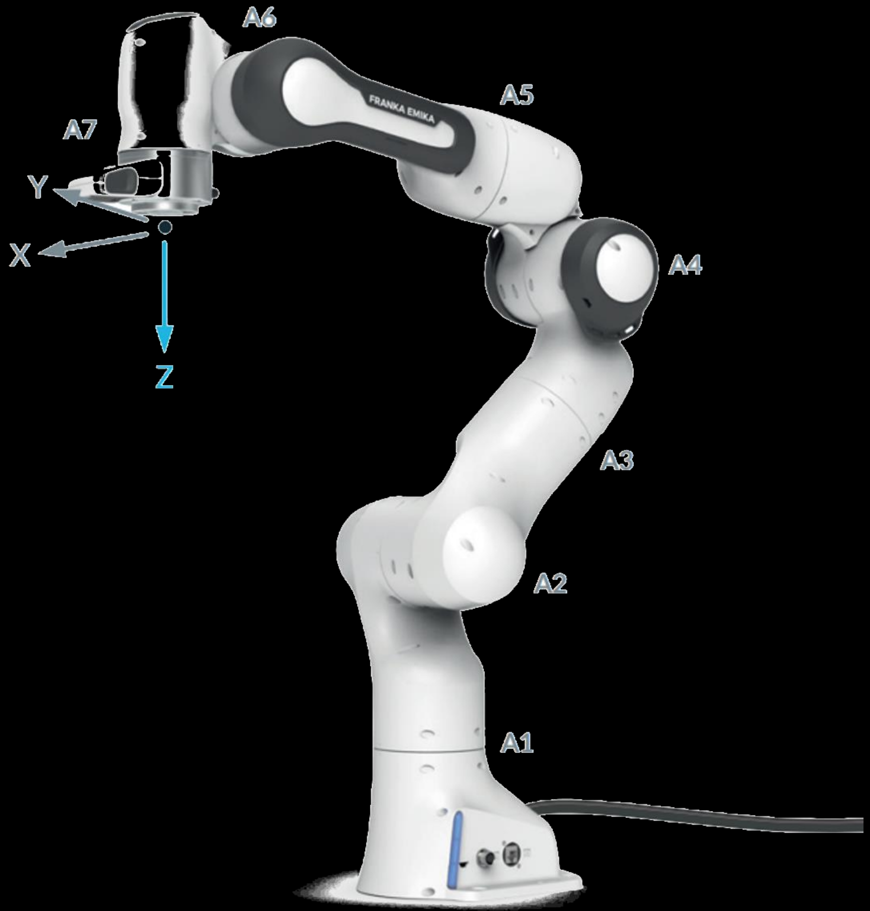
Design
how are the loads distributed

Optimization
what dimension should I have

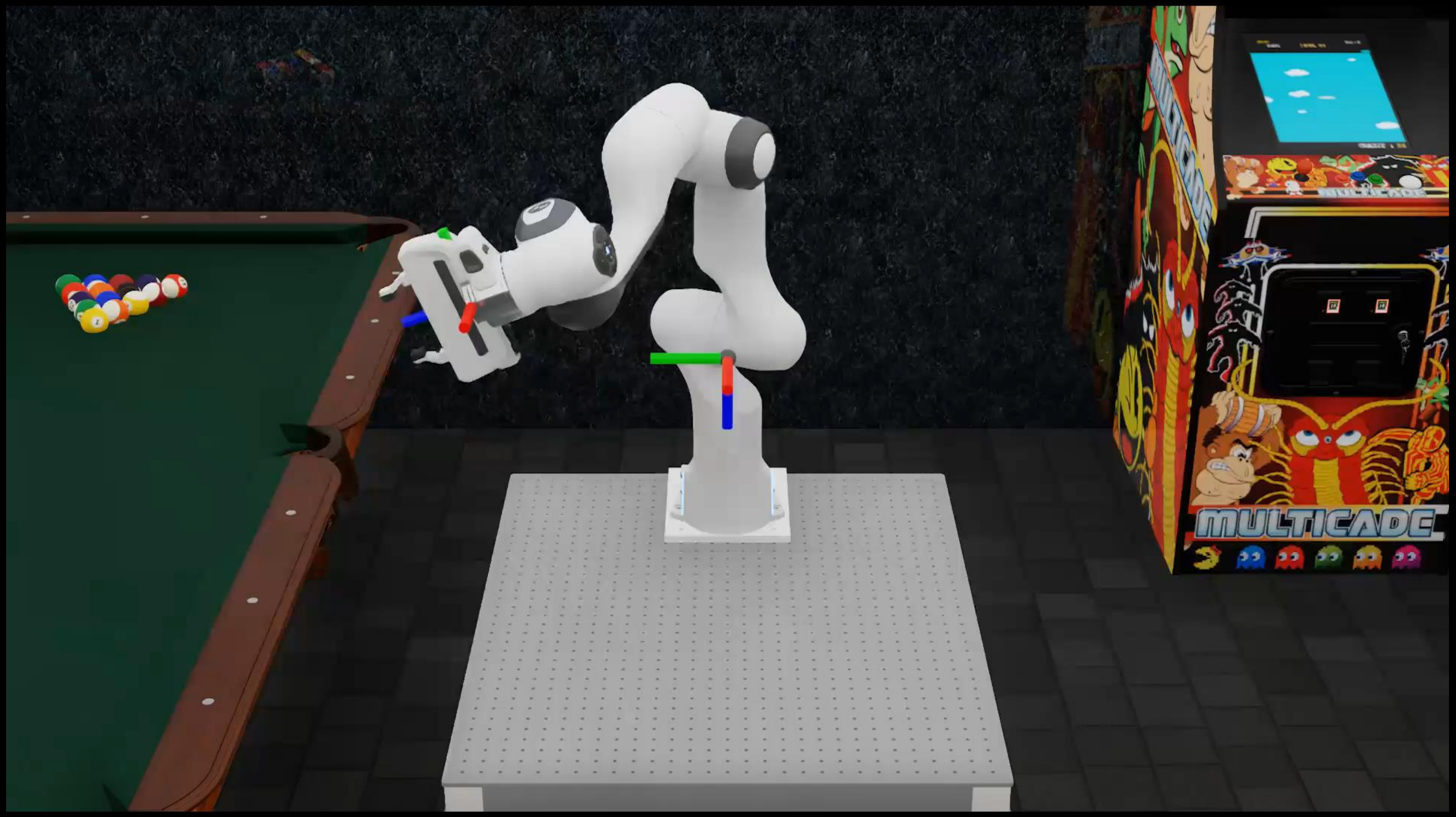
Actuation
torque, speed, powder etc.

ETH zürich SoftRobotics Laboratory

Franka Arm

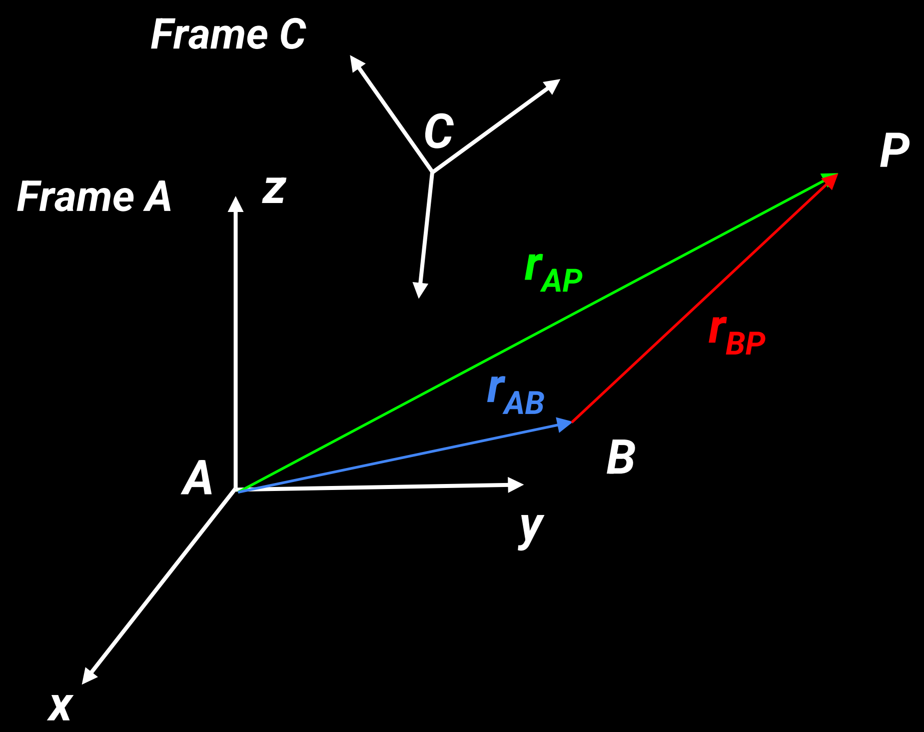


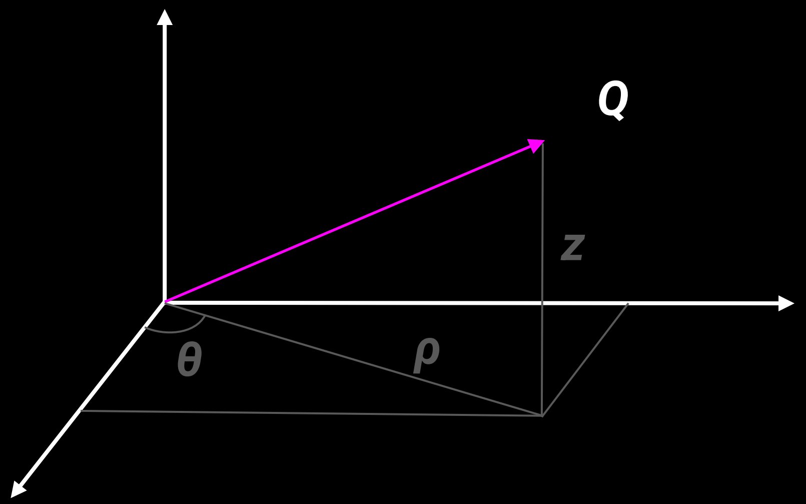
franka.de



Videos from Orbit

Points, Lines, and Coordinates





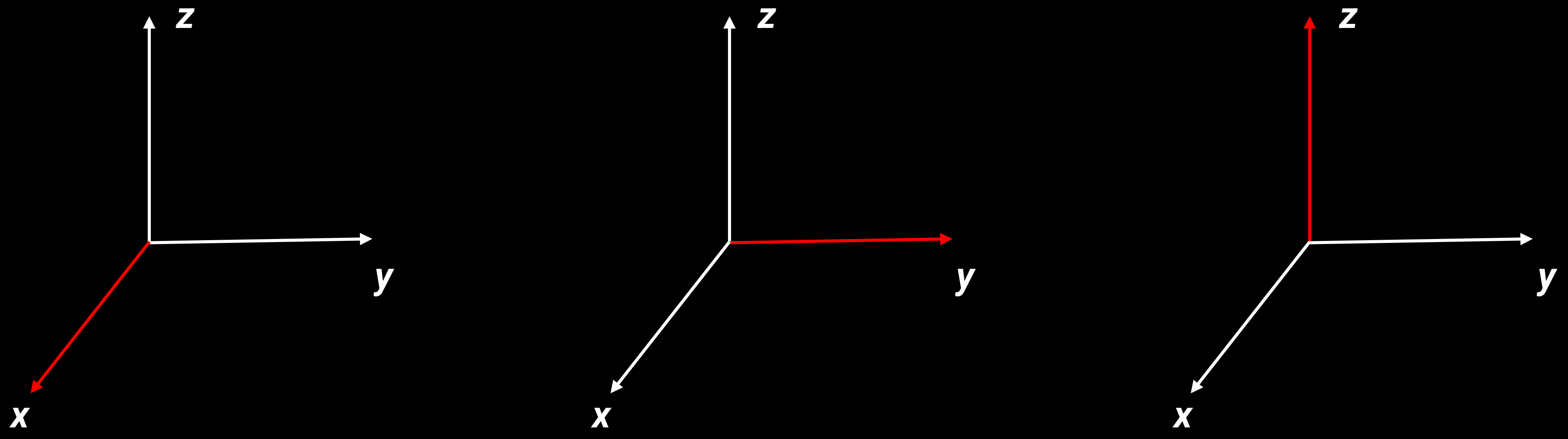
Point P in Cartesian Coordinates Frame A: $_{A}X_{P} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$

Point Q in Cylindrical Coordinate: $X_{Q} = \begin{pmatrix} \rho \\ \theta \\ z \end{pmatrix}$

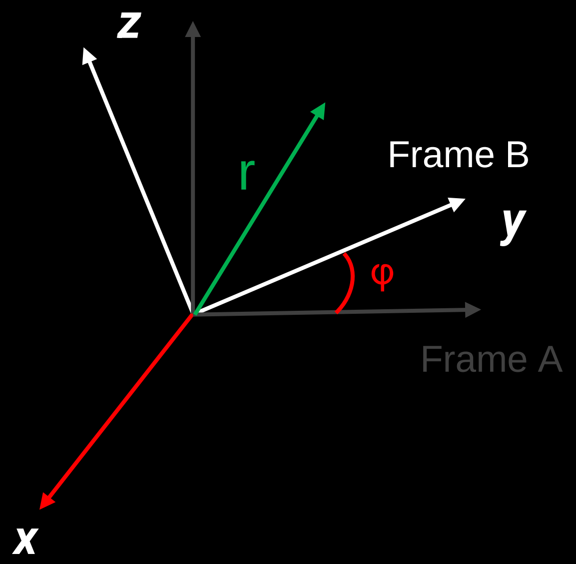
$$_{A}r_{AP} = {}_{A}r_{AB} + {}_{A}r_{BP}$$

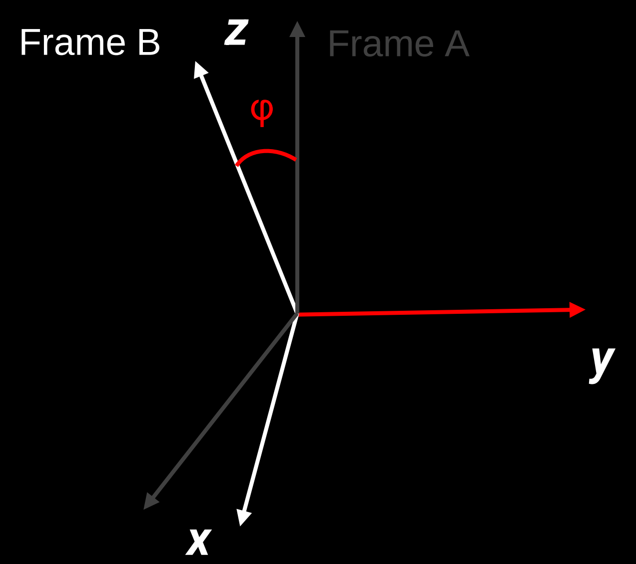
$$_{A}r_{AP} \neq {}_{A}r_{AB} + {}_{C}r_{BP}$$

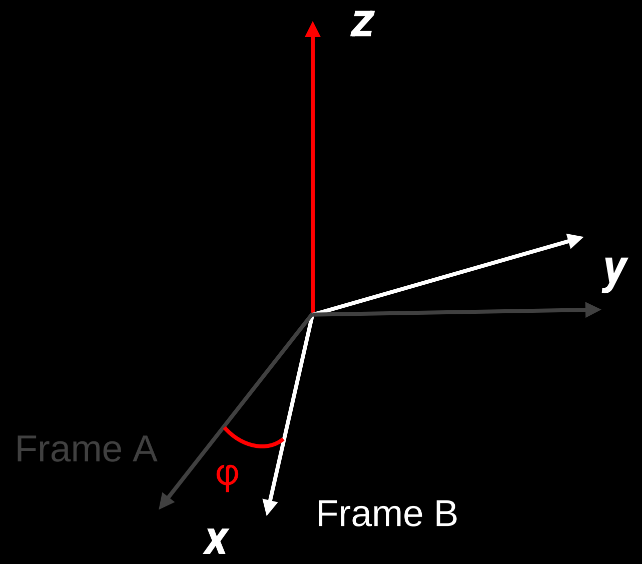
Rotation



Rotation







$$C_x(\varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\varphi & -sin\varphi \\ 0 & sin\varphi & cos\varphi \end{bmatrix}$$

$$C_y(\varphi) = \begin{bmatrix} cos\varphi & 0 & sin\varphi \\ 0 & 1 & 0 \\ -sin\varphi & 0 & cos\varphi \end{bmatrix}$$

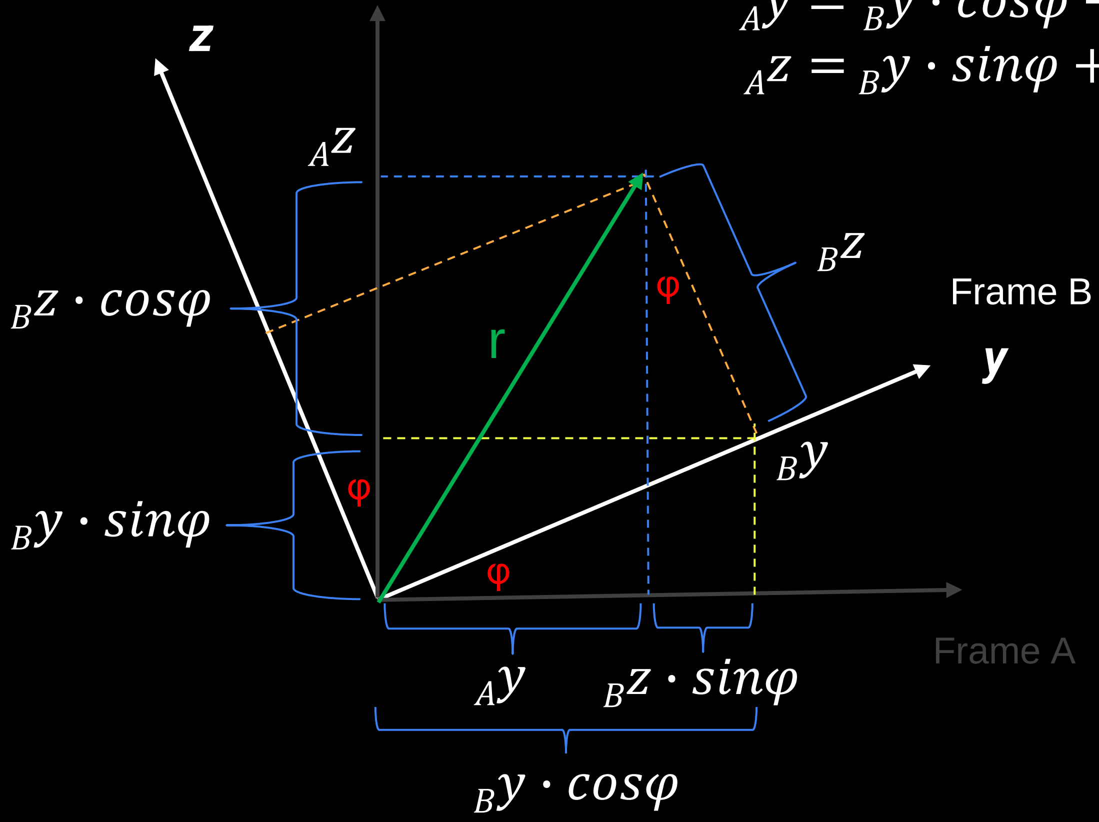
$$C_z(\varphi) = \begin{bmatrix} cos\varphi & -sin\varphi & 0 \\ sin\varphi & cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

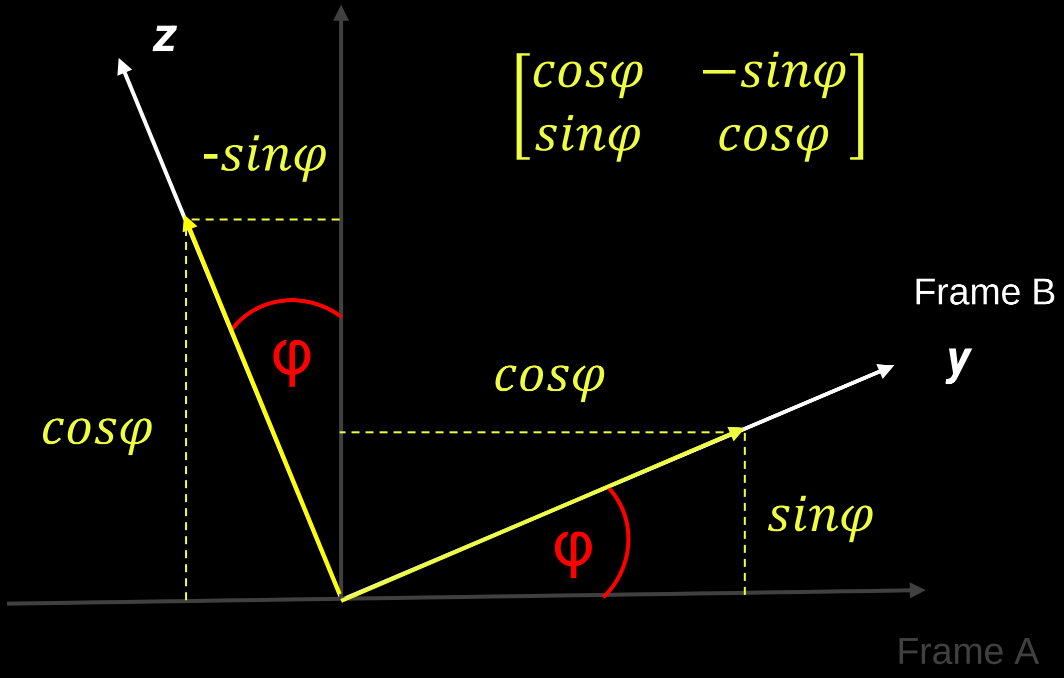
$$_A r = C_{AB} \cdot {_B r} \rightarrow \begin{pmatrix} _A x \\ _A y \\ _A z \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\varphi & -sin\varphi \\ 0 & sin\varphi & cos\varphi \end{bmatrix} \begin{pmatrix} _B x \\ _B y \\ _B z \end{pmatrix} = \begin{pmatrix} _B x \\ _B y \cdot cos\varphi - {_B z} \cdot sin\varphi \\ _B y \cdot sin\varphi + {_B z} \cdot cos\varphi \end{pmatrix}$$

Rotation

$$_A r = C_{AB} \cdot {_B r} \rightarrow \begin{pmatrix} _A x \\ _A y \\ _A z \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\varphi & -sin\varphi \\ 0 & sin\varphi & cos\varphi \end{bmatrix} \begin{pmatrix} _B x \\ _B y \\ _B z \end{pmatrix} = \begin{pmatrix} _B x \\ _B y \cdot cos\varphi - {_B z} \cdot sin\varphi \\ _B y \cdot sin\varphi + {_B z} \cdot cos\varphi \end{pmatrix}$$

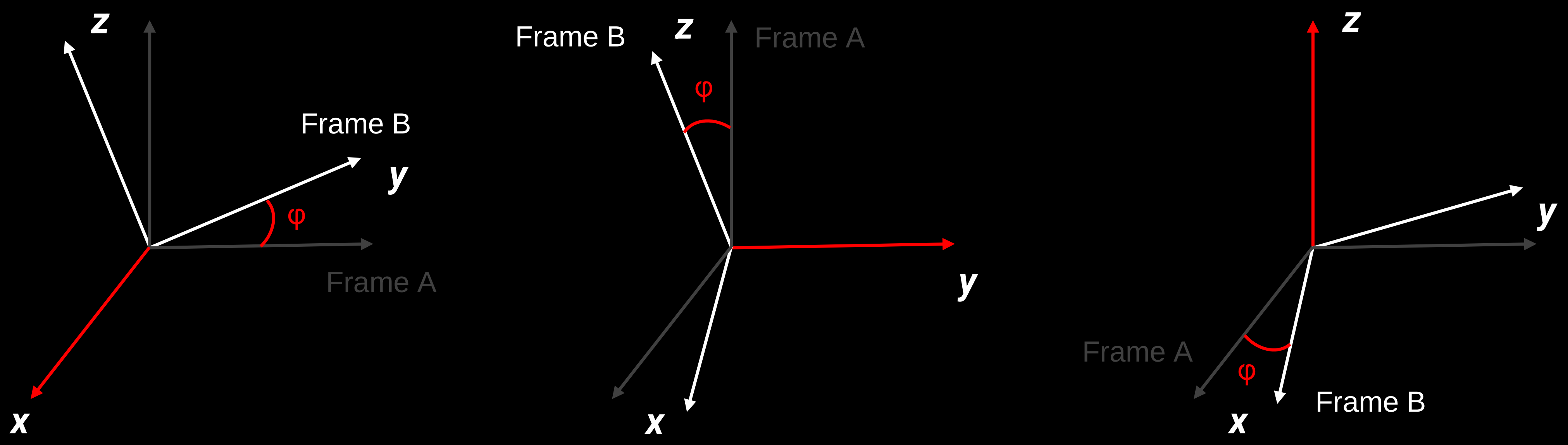
$$_A y = {_B y} \cdot cos\varphi - {_B z} \cdot sin\varphi$$
$$_A z = {_B y} \cdot sin\varphi + {_B z} \cdot cos\varphi$$





(only looking at y & z here)

Rotation



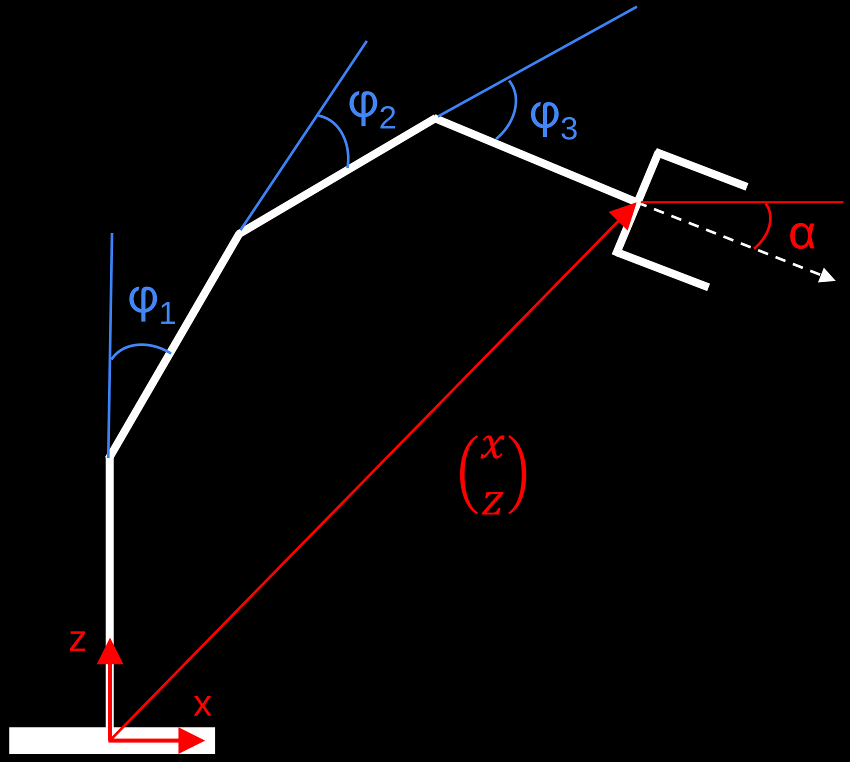
$$C_x(\varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\varphi & -sin\varphi \\ 0 & sin\varphi & cos\varphi \end{bmatrix} \quad C_y(\varphi) = \begin{bmatrix} cos\varphi & 0 & sin\varphi \\ 0 & 1 & 0 \\ -sin\varphi & 0 & cos\varphi \end{bmatrix} \quad C_z(\varphi) = \begin{bmatrix} cos\varphi & -sin\varphi & 0 \\ sin\varphi & cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

If first rotate about z axis for z_1 angle, then about y axis for y angle, and lastly about z axis again for z_2 angle:

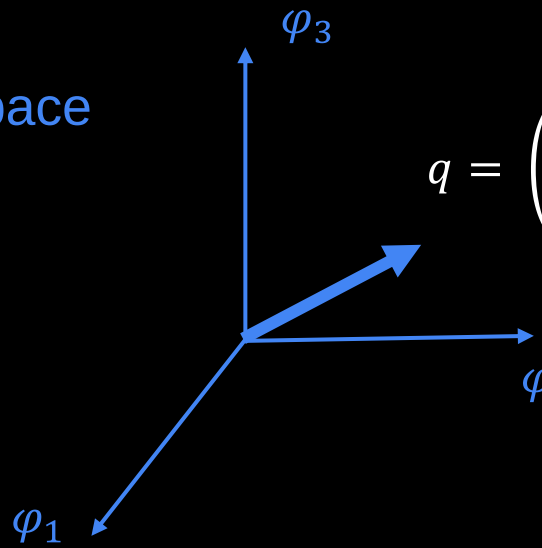
$$C_{AD} = C_{AB}(z_1)\, C_{BC}(y)\, C_{CD}(z_2) = \begin{bmatrix} cosz_1 & -sinz_1 & 0 \\ sinz_1 & cosz_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cosy & 0 & siny \\ 0 & 1 & 0 \\ -siny & 0 & cosy \end{bmatrix} \begin{bmatrix} cosz_2 & -sinz_2 & 0 \\ sinz_2 & cosz_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

↳ ${}_{A}r = C_{AD} \cdot {}_{D}r$

Joint Space and Task Space



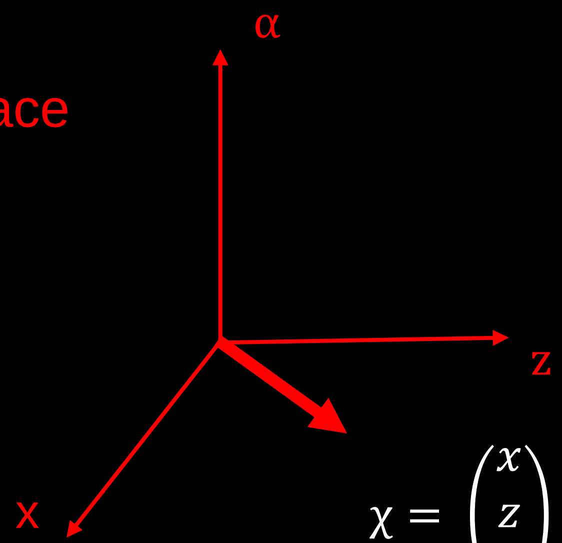
Joint Space



$$q = \begin{pmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \end{pmatrix}$$

Joint Coordinates

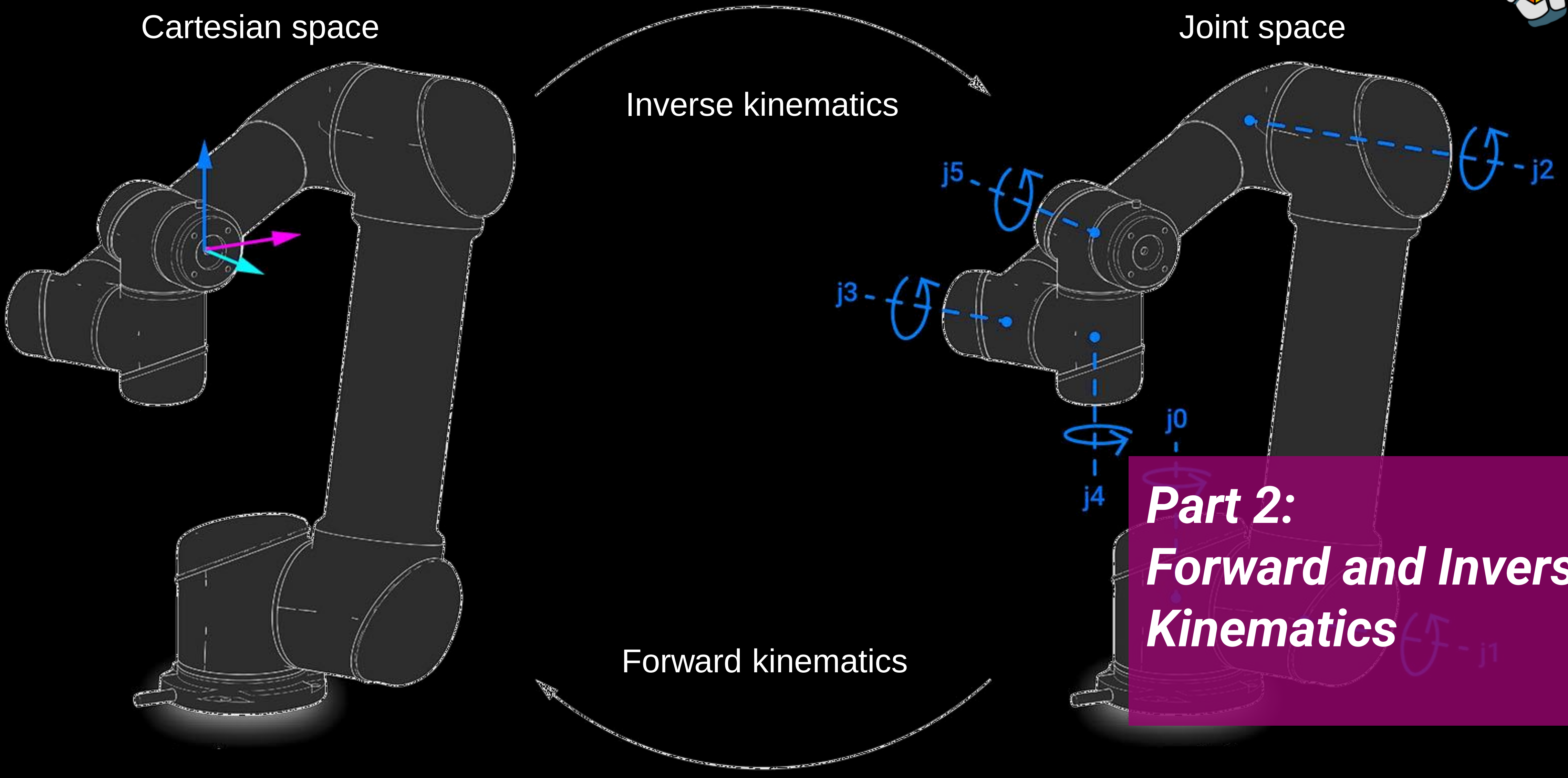
Task Space



$$\chi = \begin{pmatrix} x \\ z \\ \alpha \end{pmatrix}$$

Task Coordinates

ETH zürich

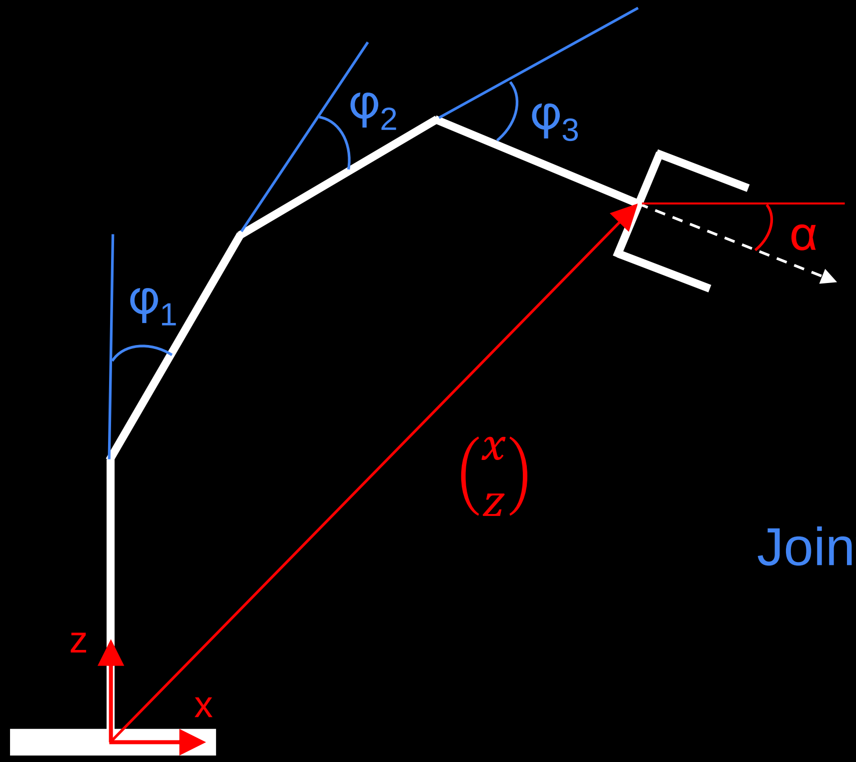


Part 2: Forward and Inverse Kinematics

compas_fab

SoftRobotics Laboratory

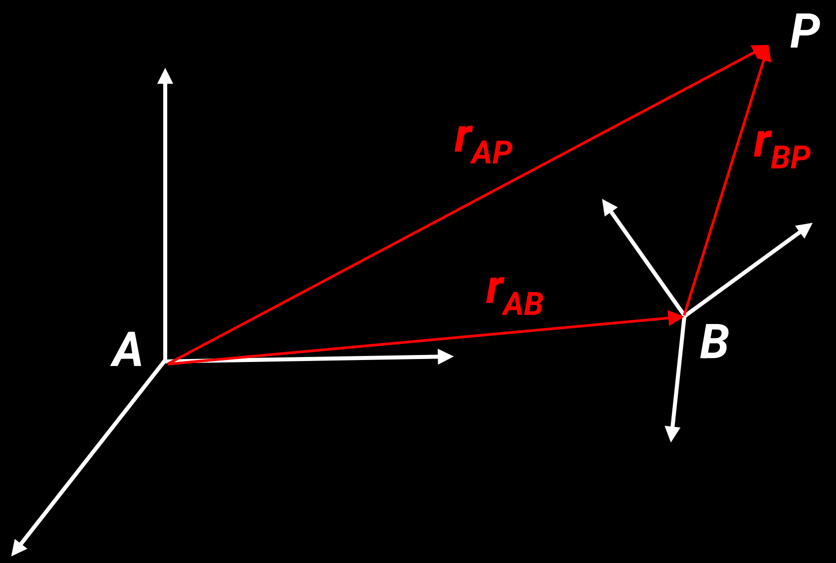
Forward and Inverse Kinematics



Joint Space —Forward Kinematics→ Task Space

Joint Space ←Inverse Kinematics— Task Space

Homogeneous Transformation Matrix



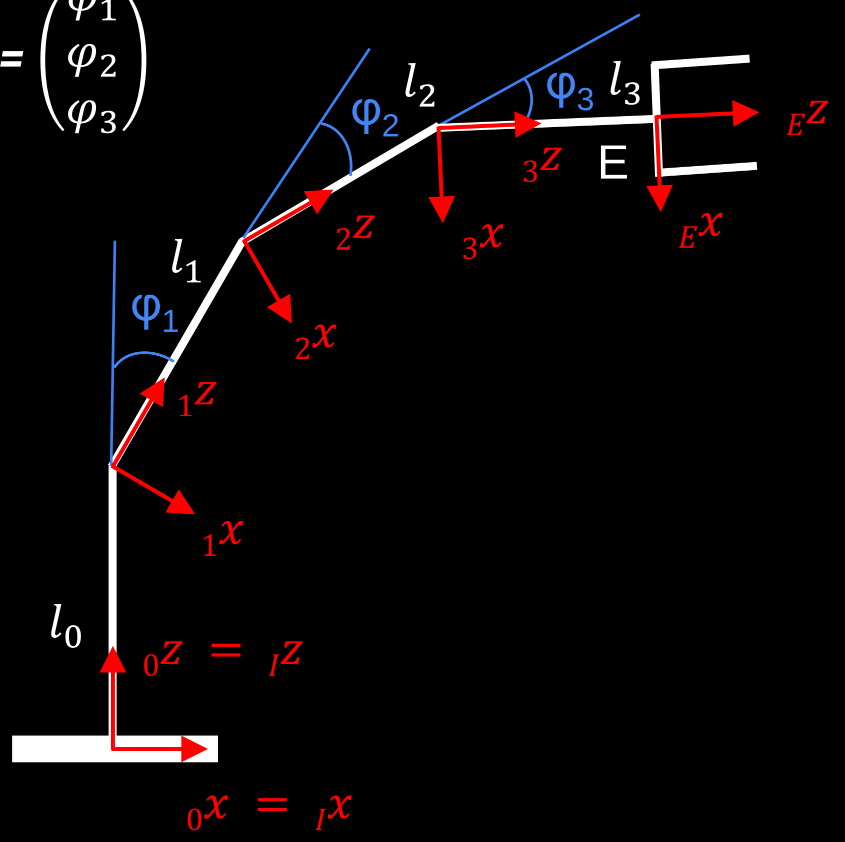
$$r_{AP} = r_{AB} + r_{BP}$$

$$_A r_{AP} = {}_A r_{AB} + {}_A r_{BP} = {}_A r_{AB} + C_{AB} \cdot {}_B r_{BP}$$

$$\begin{pmatrix} _A r_{AP} \\ 1 \end{pmatrix} = \underbrace{\begin{bmatrix} C_{AB} & _A r_{AB} \\ 0_{1x3} & 1 \end{bmatrix}}_{T_{AB}} \begin{pmatrix} _B r_{BP} \\ 1 \end{pmatrix}$$

Homogeneous Transformation Matrix

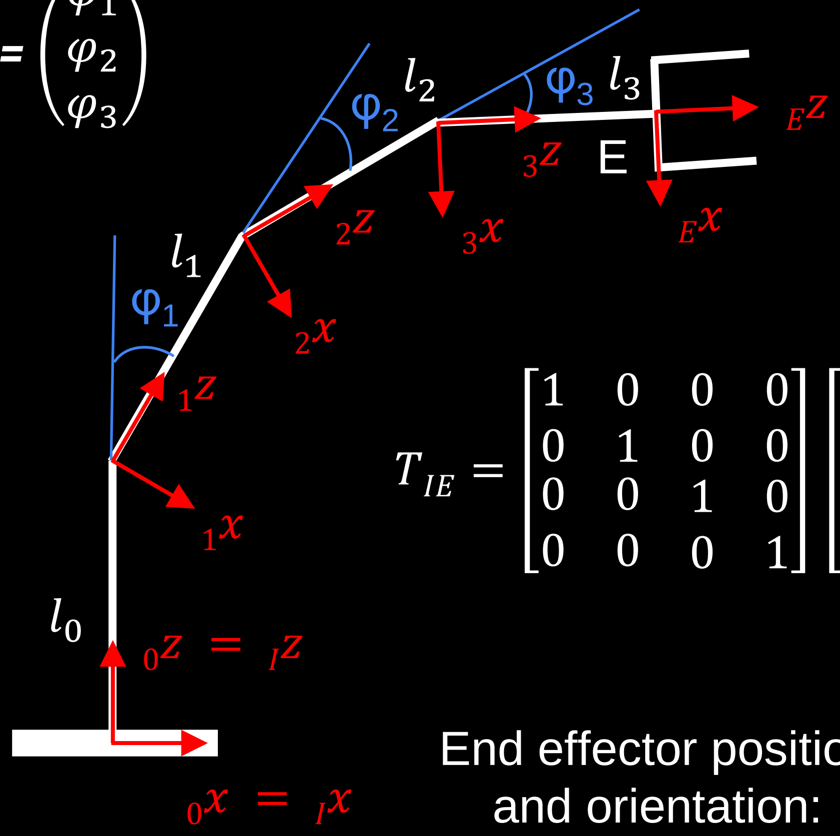
$\mathbf{q} = \begin{pmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \end{pmatrix}$



$$T_{IE} = TI_0 \cdot T_{01} \cdot T_{12} \cdot T_{23} \cdot T_{3E}$$

Homogeneous Transformation Matrix

$$\mathbf{q} = \begin{pmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \end{pmatrix}$$



$$T_{IE} = TI_0 \cdot T_{01} \cdot T_{12} \cdot T_{23} \cdot T_{3E}$$

$$T_{IE} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ 0 & 1 & 0 & 0 \\ -s_1 & 0 & c_1 & l_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_2 & 0 & s_2 & 0 \\ 0 & 1 & 0 & 0 \\ -s_2 & 0 & c_2 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_3 & 0 & s_3 & 0 \\ 0 & 1 & 0 & 0 \\ -s_3 & 0 & c_3 & l_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & l_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

End effector position and orientation:

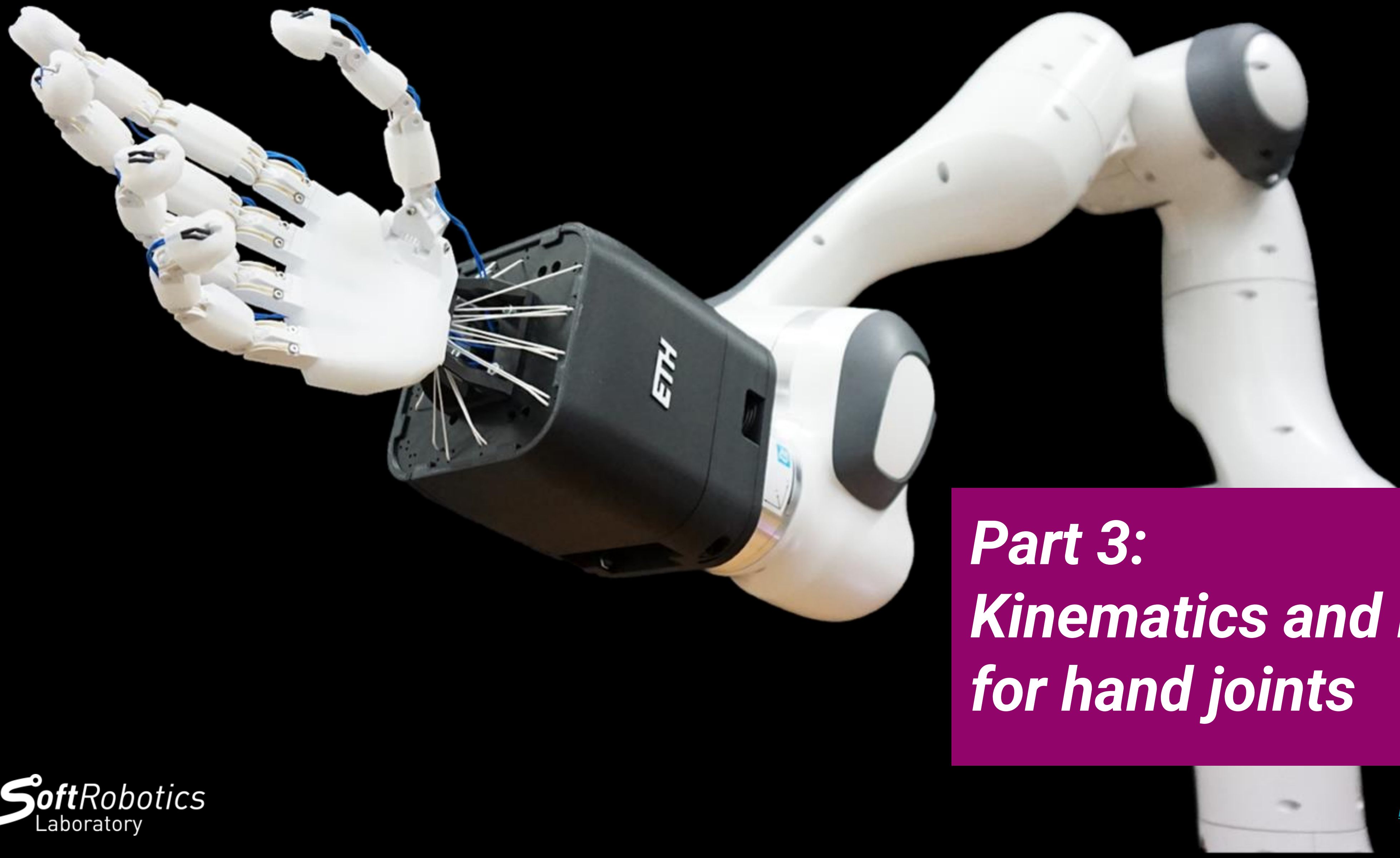
$$_I\mathbf{X}_E(\mathbf{q}) = \begin{pmatrix} l_1 \sin(\varphi_1) + l_2 \sin(\varphi_1 + \varphi_2) + l_3 \sin(\varphi_1 + \varphi_2 + \varphi_3) \\ l_0 + l_1 \cos(\varphi_1) + l_2 \cos(\varphi_1 + \varphi_2) + l_3 \cos(\varphi_1 + \varphi_2 + \varphi_3) \\ \varphi_1 + \varphi_2 + \varphi_3 \end{pmatrix}$$

Forward Differential Kinematics and Jacobian

$$\delta X_E \approx \frac{\delta X_E(q)}{\delta q}\delta q = J_{EA}(q)\delta q \qquad \text{with } J_{EA} = \frac{\delta X_E}{\delta q} = \begin{bmatrix} \frac{\delta X_1}{\delta q_1} & \cdots & \frac{\delta X_1}{\delta q_n} \\ \vdots & \ddots & \vdots \\ \frac{\delta X_m}{\delta q_1} & \cdots & \frac{\delta X_m}{\delta q_n} \end{bmatrix}$$

$$\dot{X}_E = J_{EA}(q)\dot{q} \qquad \text{with } J_{EA}(q) \in \mathbb{R}^{m\times n}$$

ETH zürich

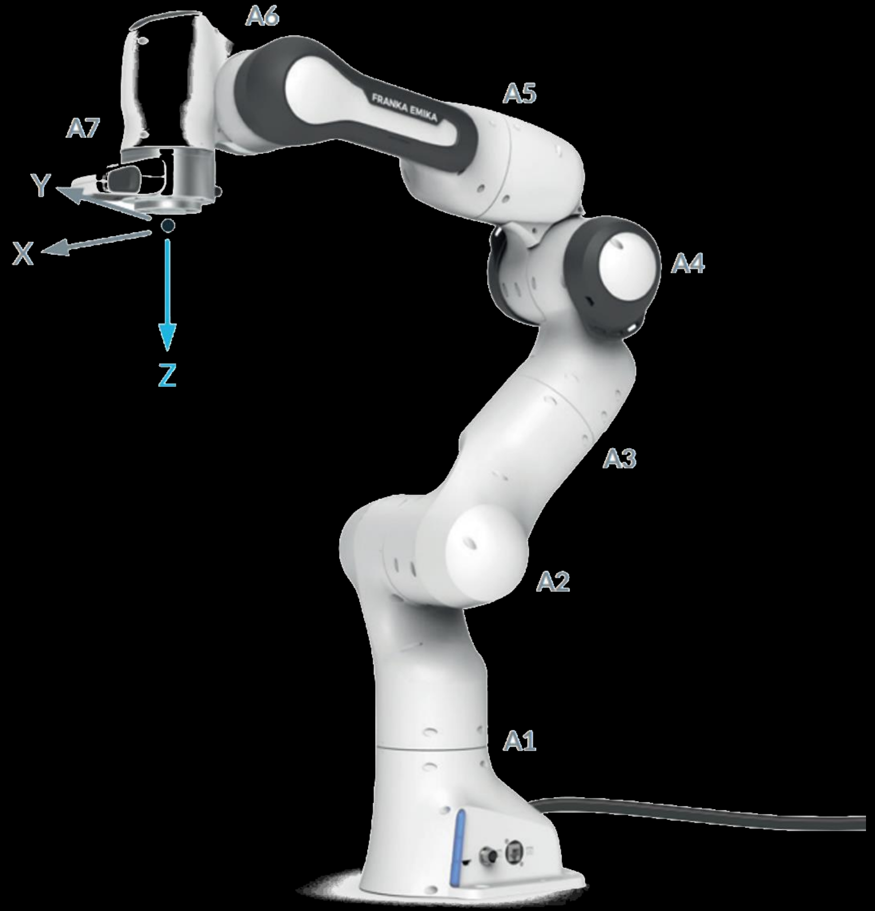


Part 3: Kinematics and Dynamics for hand joints

Faive Robotics

SoftRobotics Laboratory

Difference Between Conventional Robots and Robotic Hands

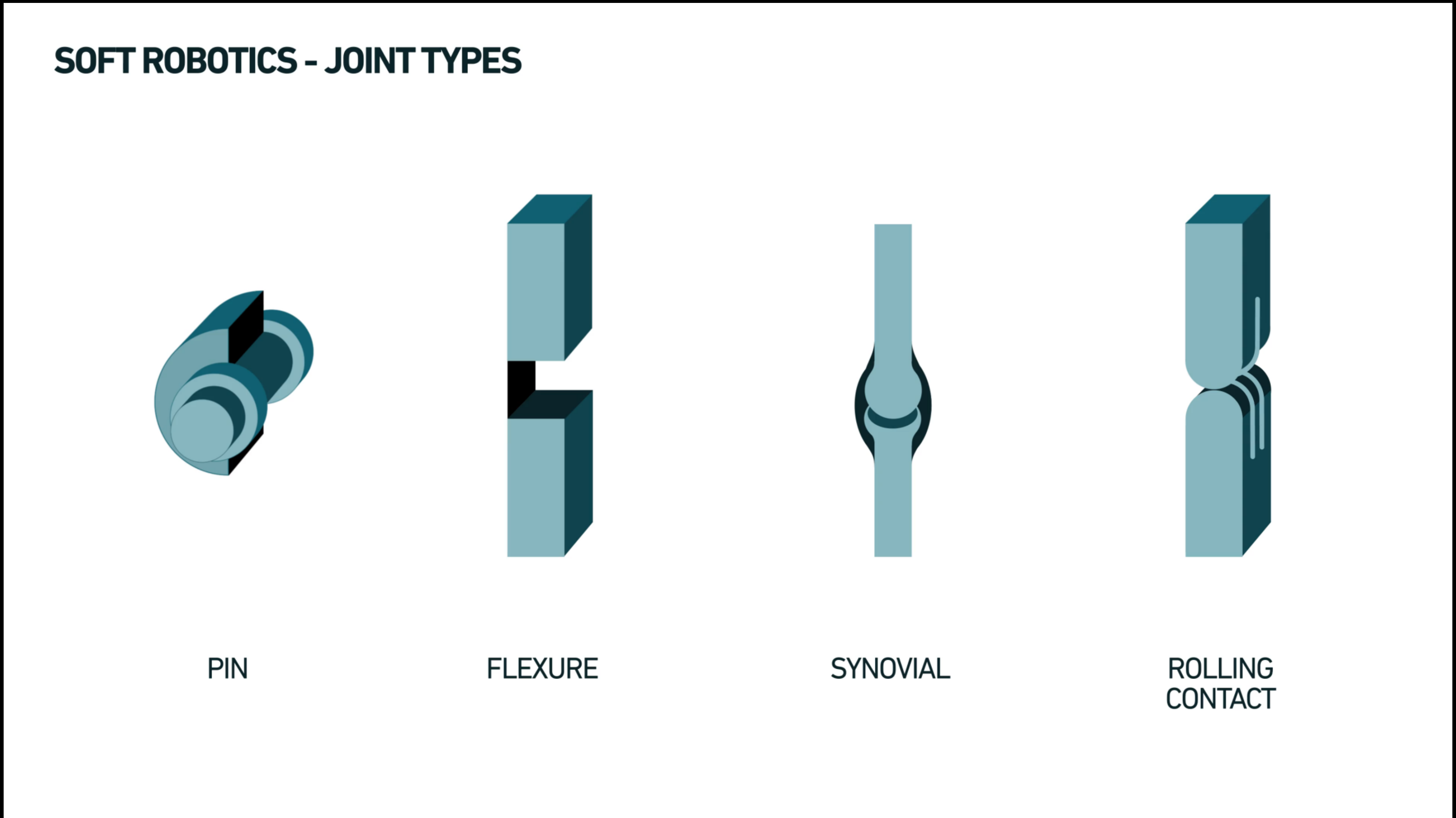


franka.de

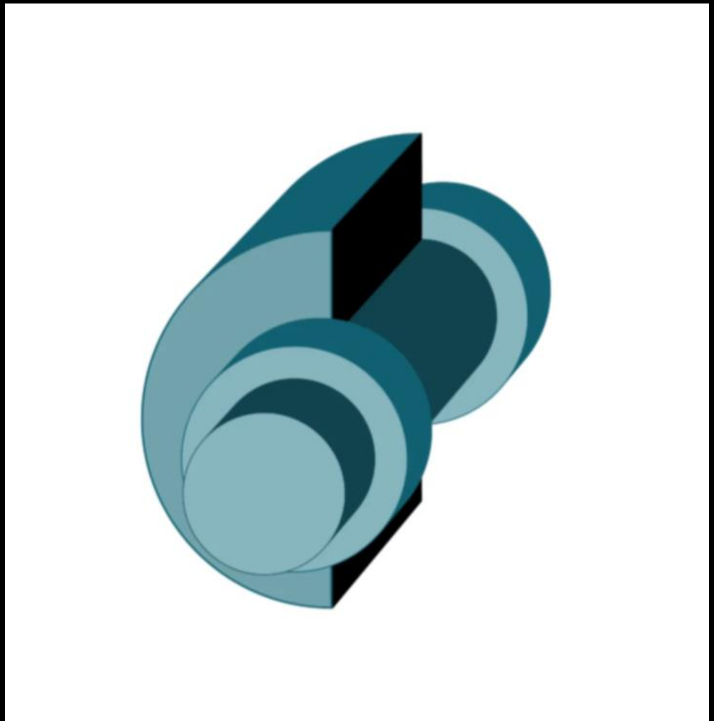
abb.com

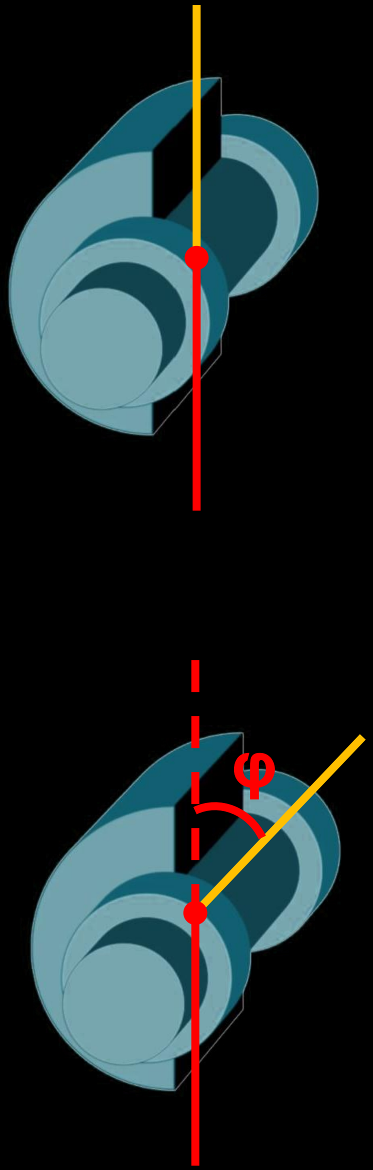
ETH zürich SoftRobotics Laboratory

Recap: Different Types of Joints

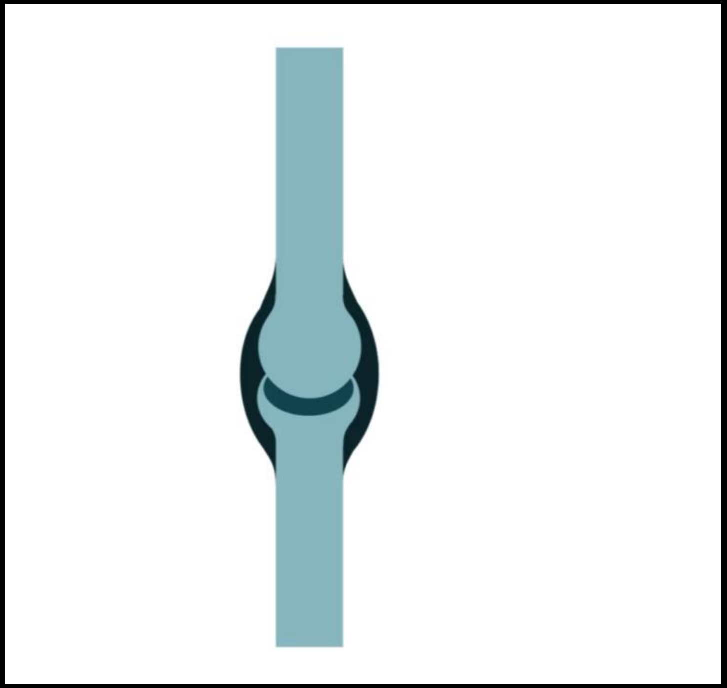


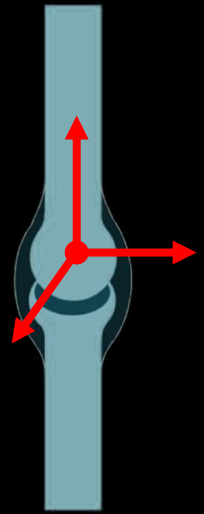
Pin Joint

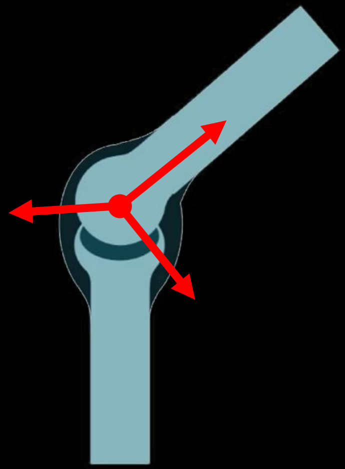




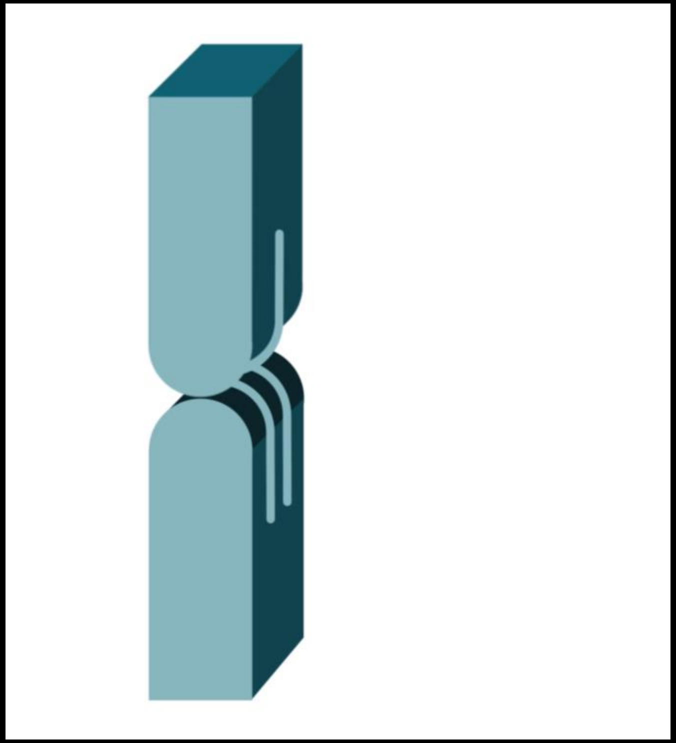
Synovial Joint

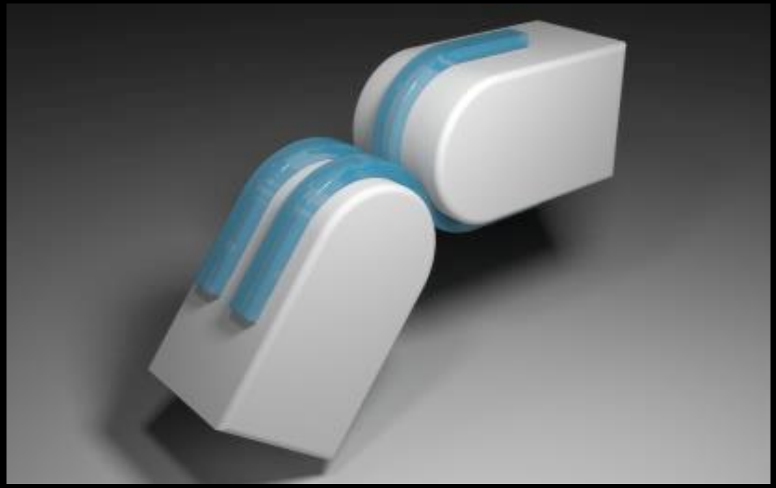


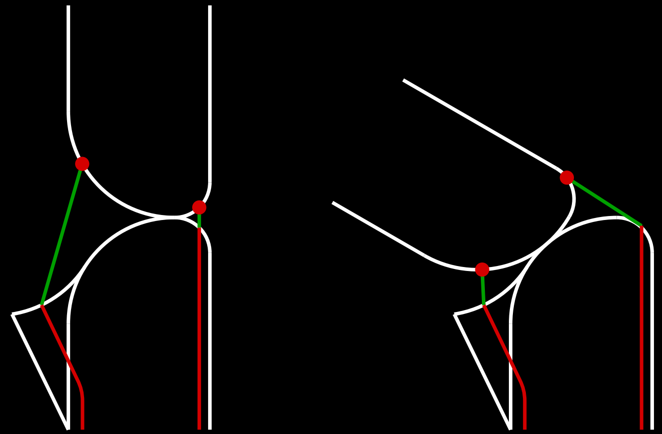




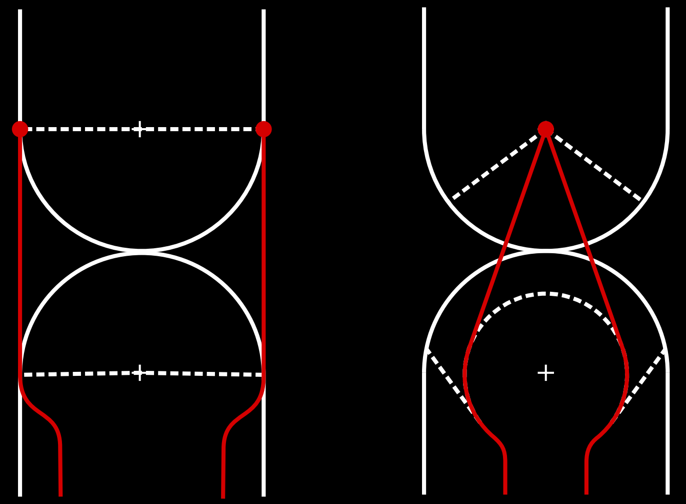
Rolling Contact Joint — Joints Used on Faive Hand

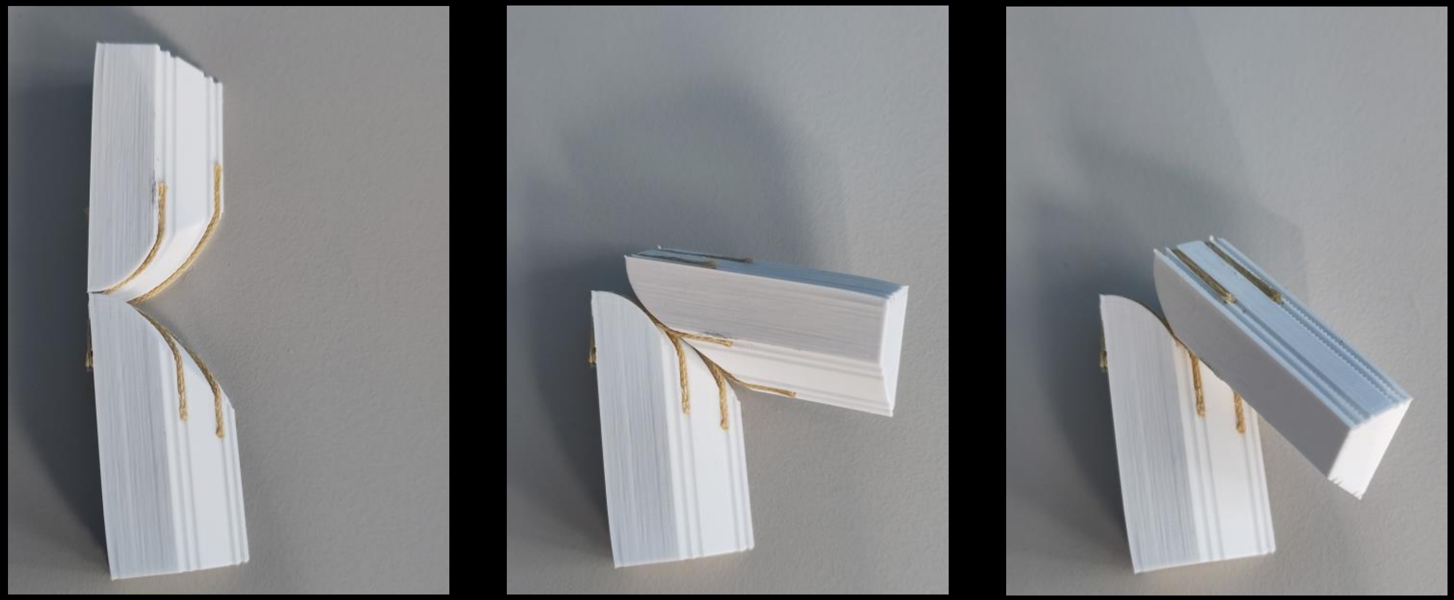




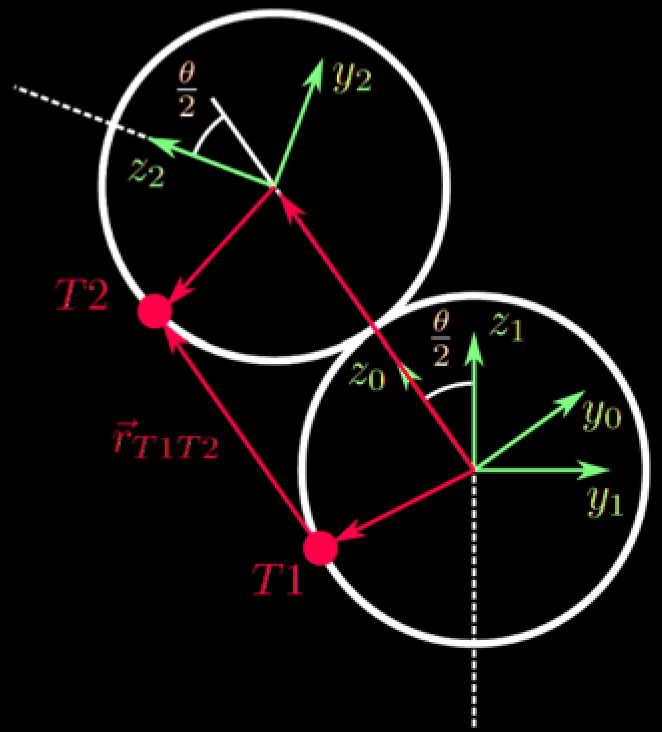


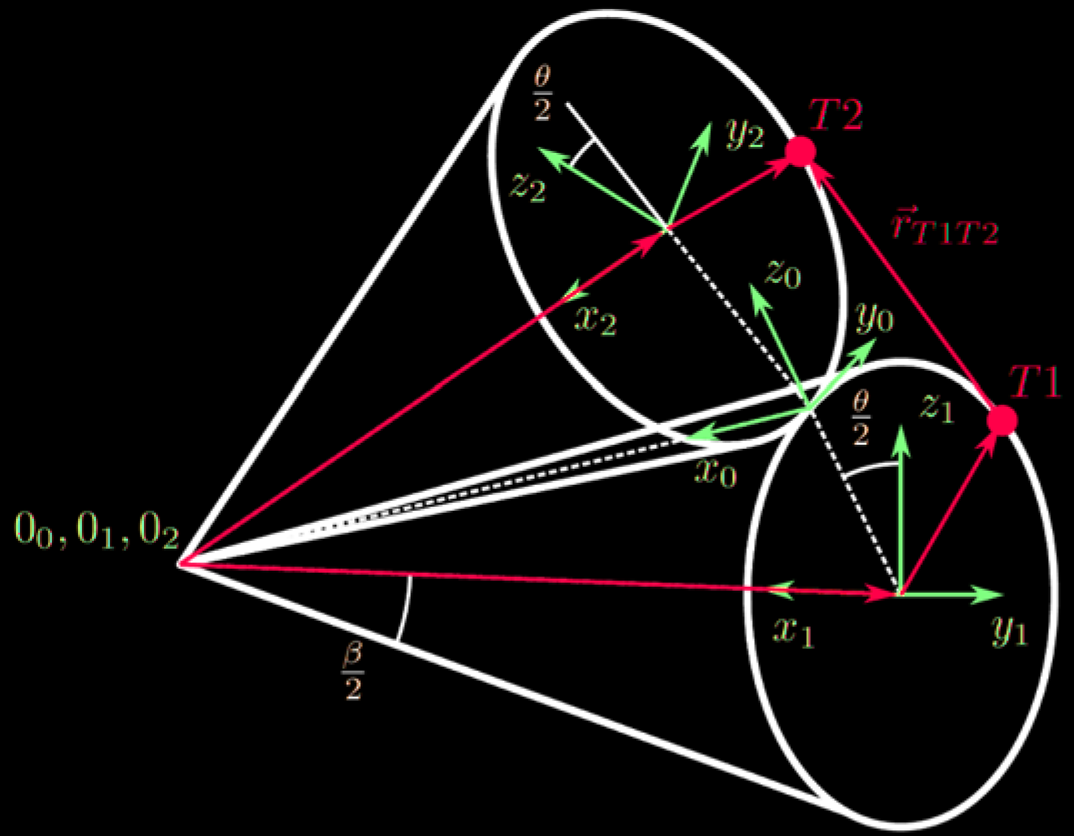
Kinematics for Rolling Contact Joint



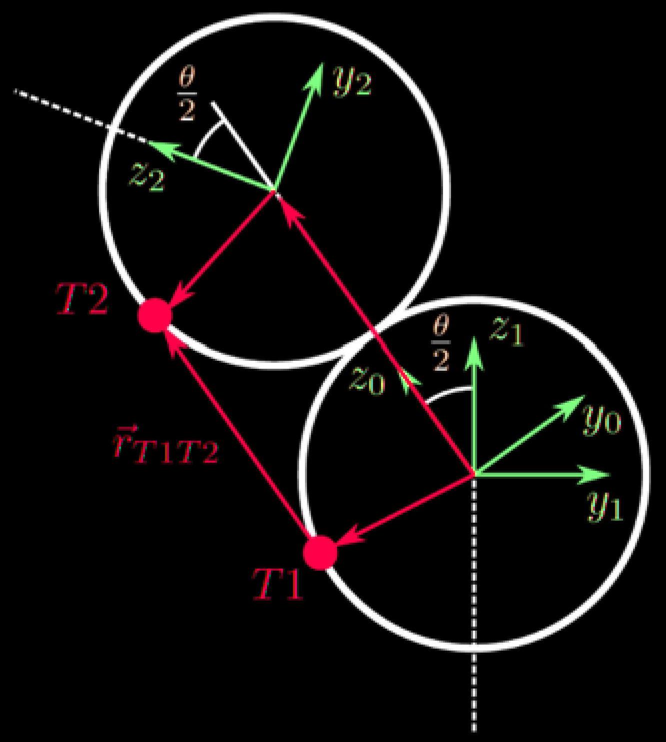


Kinematics for Rolling Contact Joint





Kinematics for Rolling Contact Joint



$$\vec{r}_{T1T2} = \begin{pmatrix} -X_1 \\ R_1 sin\alpha_1 \\ -R_1 cos\alpha_1 \end{pmatrix} + C_{10}\begin{pmatrix} 0 \\ 0 \\ 2R \end{pmatrix} + C_{12}\begin{pmatrix} X_2 \\ -R_2 sin\alpha_2 \\ R_2 cos\alpha_2 \end{pmatrix}$$

R: radius of the rolling cylinders

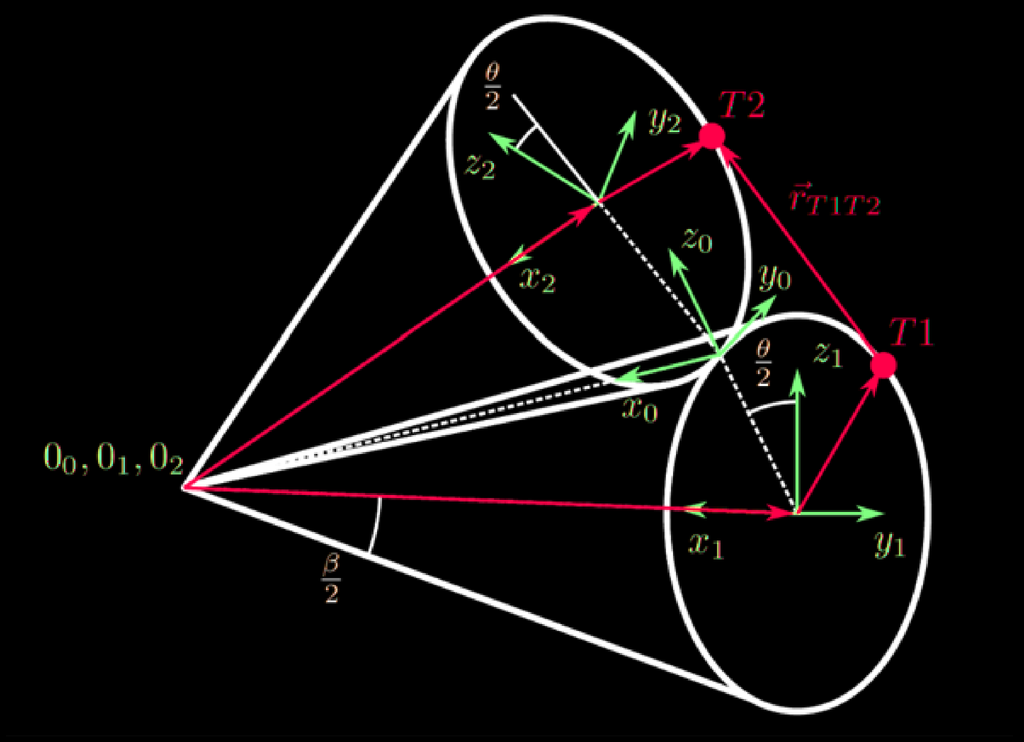
R_1, α_1 and X_1: cylinder coordinates of the point T1 in system 1

R_2, α_2 and X_2: cylinder coordinates of the point T2 in system 2

$$C_{10} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\frac{\theta}{2} & -sin\frac{\theta}{2} \\ 0 & sin\frac{\theta}{2} & cos\frac{\theta}{2} \end{bmatrix}, \qquad C_{12} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\theta & -sin\theta \\ 0 & sin\theta & cos\theta \end{bmatrix}$$

ETH zürich SoftRobotics Laboratory

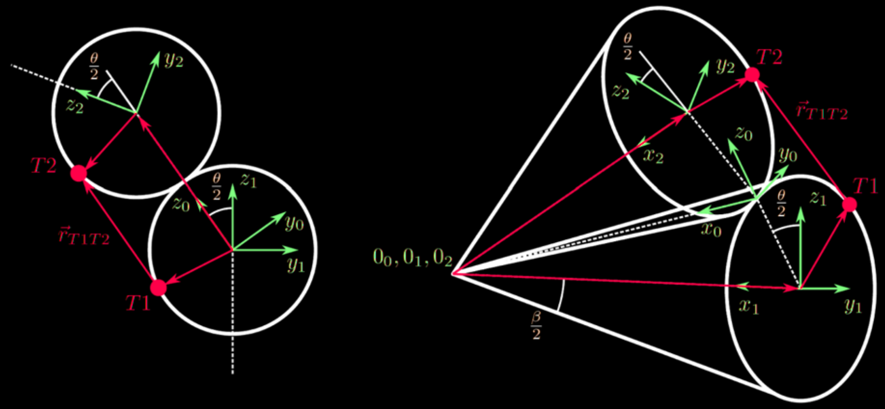
Kinematics for Rolling Contact Joint



$$\vec{r}_{T1T2} = \begin{pmatrix} -X_1 \\ R_1 sin\alpha_1 \\ -R_1 \cos\alpha_1 \end{pmatrix} + C_{12} \begin{pmatrix} X_2 \\ -R_2 sin\alpha_2 \\ R_2 cos\alpha_2 \end{pmatrix}$$

$$C_{12} = \begin{bmatrix} cos\beta & sin\frac{\theta}{2}sin\beta & cos\frac{\theta}{2}sin\beta \\ sin\frac{\theta}{2}sin\beta & cos\left(\frac{\theta}{2}\right)^2 - \sin\left(\frac{\theta}{2}\right)^2 cos\beta & -cos\frac{\theta}{2}sin\frac{\theta}{2}(1+cos\beta) \\ -cos\frac{\theta}{2}sin\beta & cos\frac{\theta}{2}sin\frac{\theta}{2}(1+cos\beta) & cos\left(\frac{\theta}{2}\right)^2 cos\beta - \sin\left(\frac{\theta}{2}\right)^2 \end{bmatrix}$$

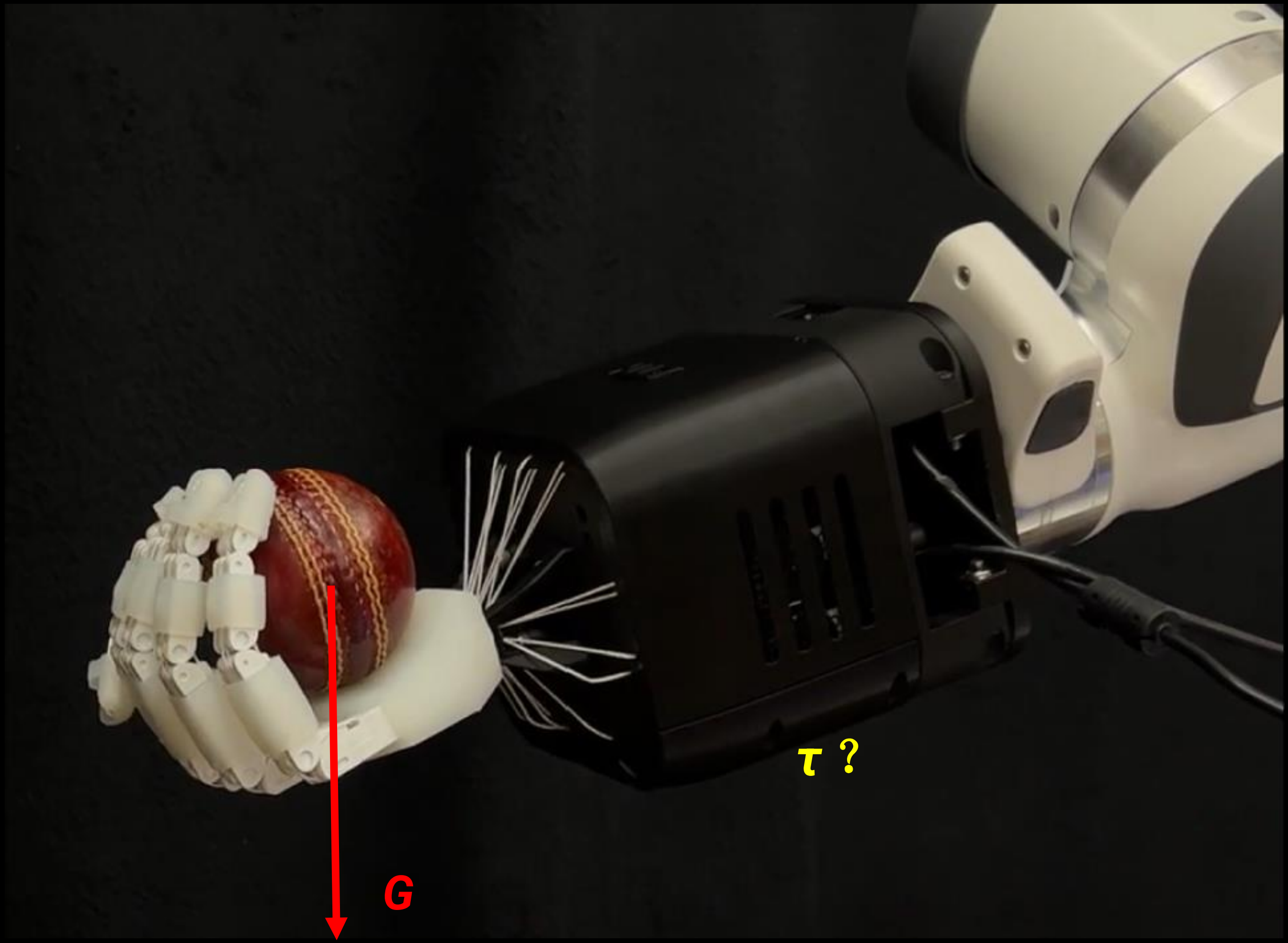
Kinematics for Rolling Contact Joint



$$\vec{r}_{T1T2} = \begin{pmatrix} -X_1 \\ R_1 sin\alpha_1 \\ -R_1 cos\alpha_1 \end{pmatrix} + C_{10}\begin{pmatrix} 0 \\ 0 \\ 2R \end{pmatrix} + C_{12}\begin{pmatrix} X_2 \\ -R_2 sin\alpha_2 \\ R_2 cos\alpha_2 \end{pmatrix}$$

$$C_{10} = \begin{bmatrix} cos\frac{\beta}{2} & 0 & sin\frac{\beta}{2} \\ sin\frac{\beta}{2}sin\frac{\theta}{2} & cos\frac{\theta}{2} & -cos\frac{\beta}{2}sin\frac{\theta}{2} \\ -sin\frac{\beta}{2}cos\frac{\theta}{2} & sin\frac{\theta}{2} & cos\frac{\beta}{2}cos\frac{\theta}{2} \end{bmatrix}$$

Dynamics



Dynamics

Tendon Lengths

$$p = g(l) = g\big(f(q)\big) = F(q)$$

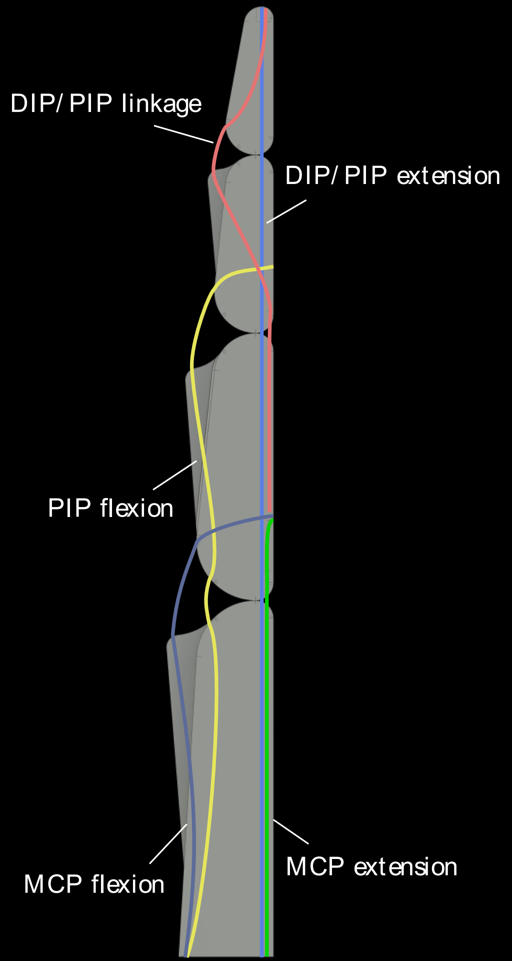
Motor Positions

Joint Angles

ETH zürich SoftRobotics Laboratory

Dynamics

$$J_m = \begin{bmatrix} \frac{\partial p_1}{\partial q_1} & \frac{\partial p_1}{\partial q_2} \\ \frac{\partial p_2}{\partial q_1} & \frac{\partial p_2}{\partial q_2} \end{bmatrix}$$



ETH zürich SoftRobotics Laboratory

Dynamics

Velocity of the finger joints

$$\dot{p} = J_m \cdot \dot{q}$$

Velocity of the motors

$$\tau^T \cdot \dot{q} = T^T \cdot \dot{p}$$

Conservation of Power

$$\tau^T \cdot \dot{q} = T^T \cdot J_m \cdot \dot{q} \quad \rightarrow \quad \tau = {J_m}^T \cdot T$$

Dynamics

Previous slide: $\tau = {J_m}^T \cdot T$

$$\dot{X}_{fingertip} = Jfing_{ertip} \cdot \dot{q}$$

$$\tau^T \cdot \dot{q} = {F_{fingertip}}^T \cdot \dot{X}_{fingertip}$$

$$\tau^T \cdot \dot{q} = {F_{fingertip}}^T \cdot J_{fingertip} \cdot \dot{q}$$

$$\tau = {J_{fingertip}}^T \cdot F_{fingertip}$$

Dynamics

$$\tau = {J_m}^T \cdot T$$

$$\tau = {J_{fingertip}}^T \cdot F_{fingertip}$$

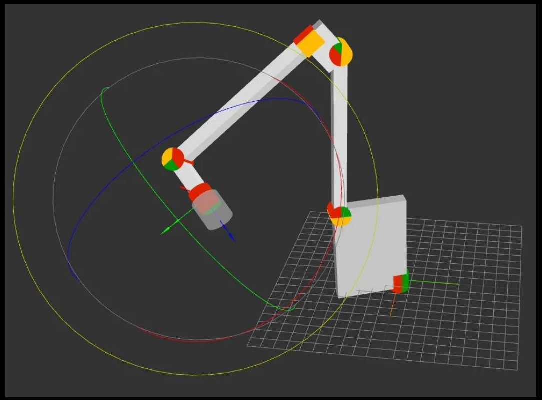
$$T = \left({J_m}^T\right)^{-1} \cdot {J_{fingertip}}^T \cdot F_{fingertip}$$

Summary

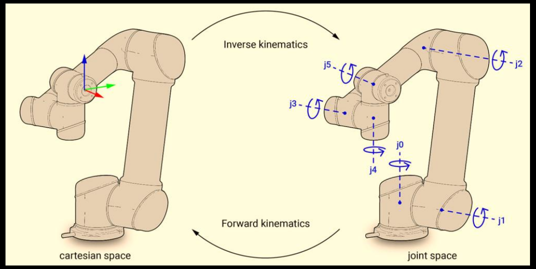
- *Intro to Robot Kinematics and Dynamics*
  - *Representing points and lines in different coordinates and frames*
  - *Rotational matrix*
  - *Joint space and task space*

- *Forward and Inverse Kinematics*
  - *Homogeneous transformation matrix*
  - *Forward differential kinematics and Jacobian*
  - *Inverse kinematics*

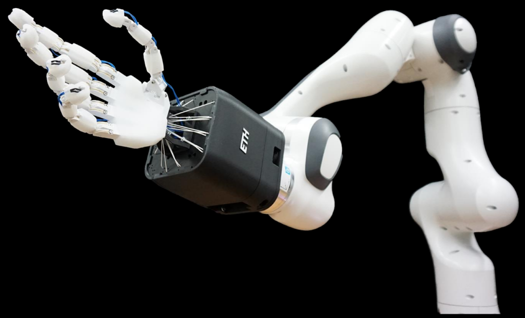
- *Kinematics and Dynamics for hand joints*
  - *Hand Joints*
  - *Kinematics for rolling joints*
  - *Dynamics for rolling joints*



Marginally Clever Robots



compas_fab

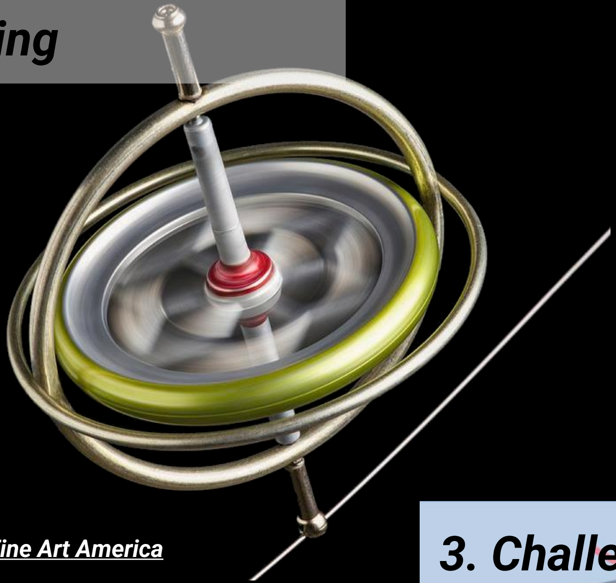


Faive Robotics

ETH zürich
SoftRobotics Laboratory

Next Tutorial? Implementing Control Strategies for Manipulation!

1. Sensing

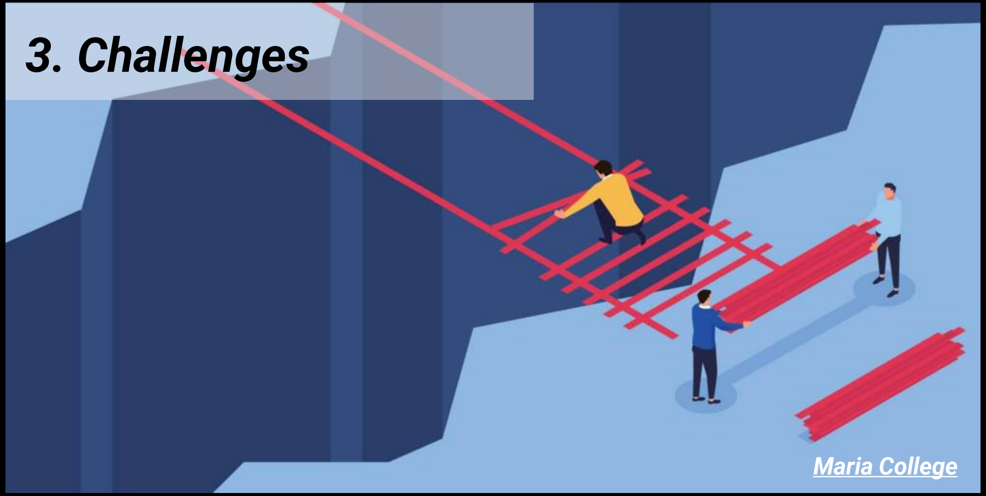


Fine Art America

2. Control

Wikimedia

3. Challenges



Maria College

ETH zürich

SoftRobotics Laboratory