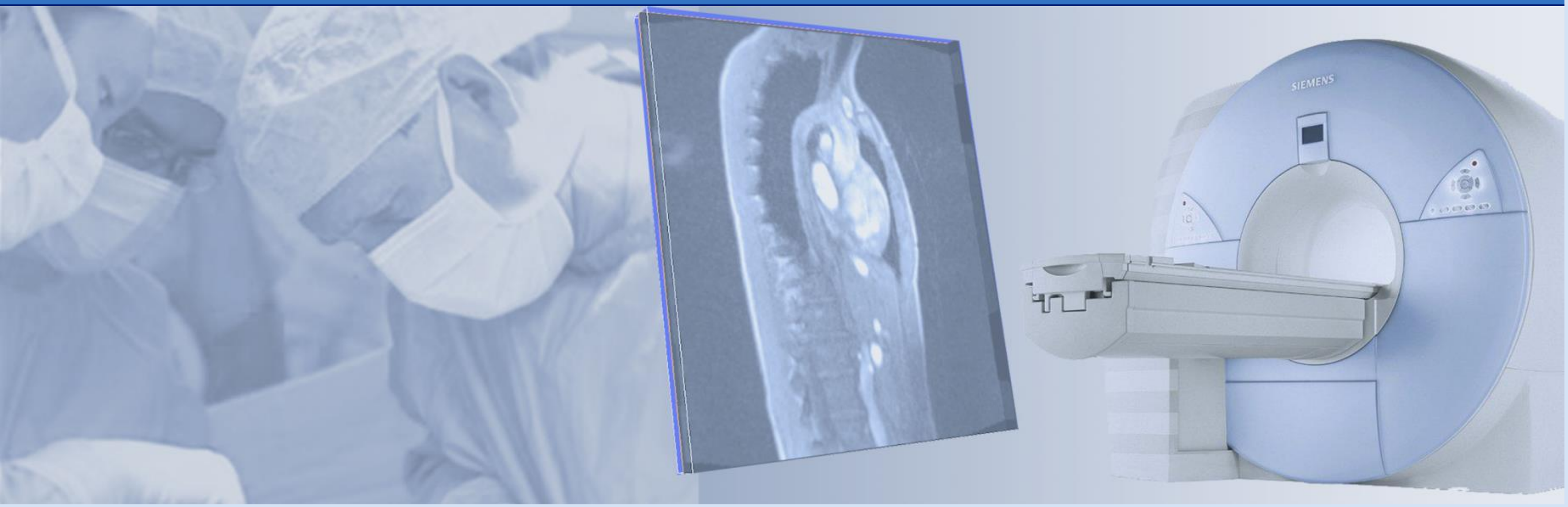
Computer- and robot-assisted Surgery



NCT
NATIONALES CENTRUM
FÜR TUMORERKRANKUNGEN
PARTNERSTANDORT DRESDEN
UNIVERSITÄTS KREBSCENTRUM UCC

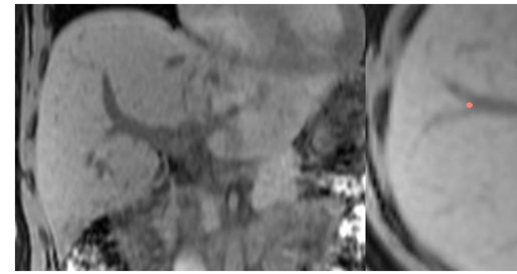
getragen von:
Deutsches Krebsforschungszentrum
Universitätsklinikum Carl Gustav Carus Dresden
Medizinische Fakultät Carl Gustav Carus, TU Dresden
Helmholtz-Zentrum Dresden-Rossendorf

Lecture 13

Robotics

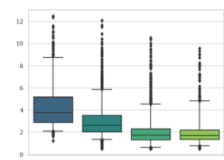
Master Thesis Topic: Analysis of a Human in-vivo Liver Deformation Dataset for Non-Rigid Registration

Supervisor: Bianca Güttner (bianca.guettner@nct-dresden.de)



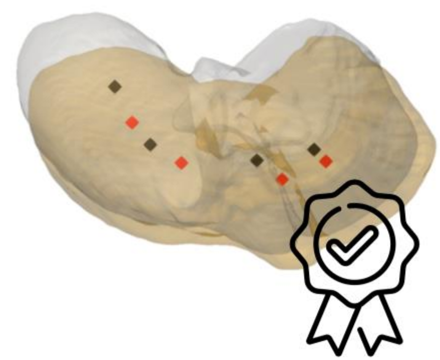
Clinical context

- Real-world data for evaluating preoperative-intraoperative landmark registration is sparse → collected a dataset (annotation ongoing)
- Idea: Evaluate quality and share the dataset to support non-rigid registration research



Tasks

- Meta-analysis of dataset quality reporting in the literature
- Derivation of quality criteria for registration datasets
- Application to the collected dataset
- Discussion of implications for the intended downstream task



Skills & Tools

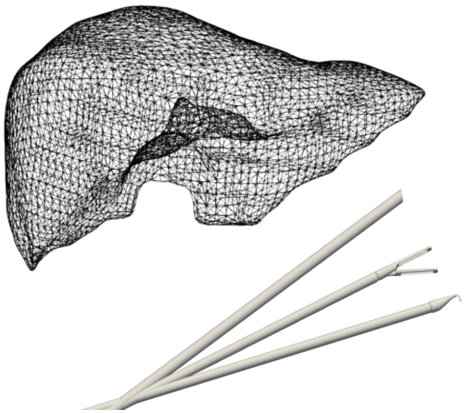
- Strong analytical skills, attention to detail
- Python (scripting, data handling, environment setup, version control)
- VTK, SimpleITK, pandas

Surface and landmark image: https://doi.org/10.1007/978-3-030-59719-1_70, Icons: Flaticon

Research Assistant Position or Research Project:

Designing an Experimental Test Bench for Synthetic Tissue Deformation and Cuts

Supervisor: Bianca Güttner (bianca.guettner@nct-dresden.de)

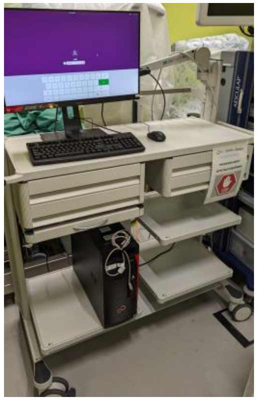


Clinical context

- Simulated data (tissue deformation) for training surgical navigation systems is being extended by manipulations
- To Do: Validation through experiments on a test bed including precisely defined movements and cuts

Tasks

- Conceptualization, construction and evaluation of a validation setup → actuators, sensors, deformation assessment
- Development of data collection and storage protocols → FAIR principles



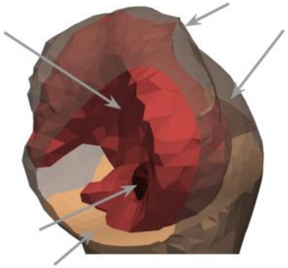
Skills & Tools

- Python, C++ (scripting, environment setup, version control)
- Interest in hardware elements (sensors, actuators) and their specifications
- Strong analytical skills, attention to detail
- Robot Operating System (ROS), microcontrollers

Liver: https://amos22.grand-challenge.org/

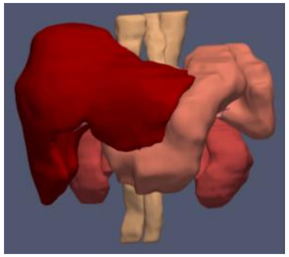
Research Assistant Position: Advancing Simulation Methods: Tissue Manipulation and Remeshing

Supervisor: Bianca Güttner (bianca.guettner@nct-dresden.de)



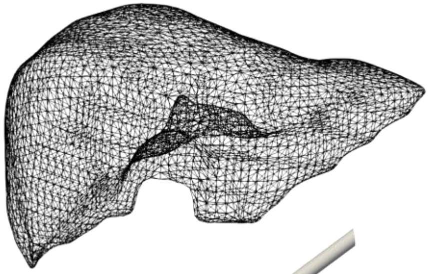
Clinical context

- Simulated data (tissue deformation) for training surgical navigation systems lacks manipulations → limited use
- Mitigation: Incorporate cuts into simulations



Tasks

- Familiarization with the in-house data generation pipeline
- Implementation of cut simulations (with and without known path)
- Implementation of communication between an external mesher and the simulation, incl. adaptation of the simulation to mesh changes between timesteps (matrix updates)



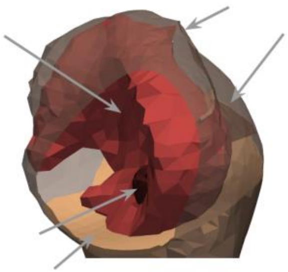
Skills & Tools

- Python, C++, Simulation Open Framework Architecture (SOFA)
- Quickly adapting to new frameworks, interfaces, and concepts
- Interest in FEM simulations and basic soft tissue mechanics

Organs: https://amos22.grand-challenge.org/

Research Assistant Position: Developing a Synthetic Data Generation Pipeline for Surgical Navigation

Supervisor: Bianca Güttner (bianca.guettner@nct-dresden.de)

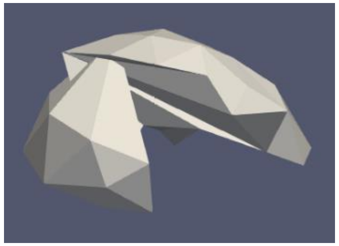


Clinical context

- Data for training surgical navigation algorithms is sparse → generate synthetic data of tissue deformation
- Contribution: Process large number of samples with automated data generation pipeline

Tasks

- Integration of methods into pipeline functionality, method development
- Failure case analysis and mitigation
- Preprocessing of real intraoperative data for validation
- Literature research on datasets and algorithms



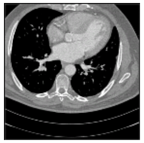
Skills & Tools

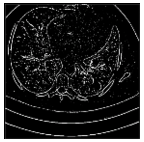
- Python (scripting, environment setup, version control)
- Quickly adapting to new frameworks, interfaces, and concepts
- Literature survey skills
- Visualization Toolkit (VTK), SimpleITK, video processing

Liver: https://amos22.grand-challenge.org/, Icons: Flaticon

Process chain computer-assisted surgery

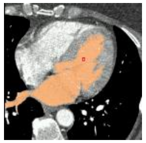
Image acquisition → Pre-processing





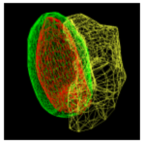
↓

Segmentation



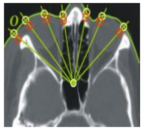
↓

Modeling



↓

Registration



↓

Visualization/
Assistance

NCT

Overview of the lecture

- Robotics
  - General definition
  - Components of a robotic system
  - Modelling a robotic system
    - Forward and inverse kinematics
  - Path Planning
  - History surgical robots

What is a robot?

- Term: Robota, Slavic for “forced labor
  - Coined by Karel Capek in 1920
- Context industry (VDI-guideline 2860, 1990)
  - A robot is a freely programmable, multi-functional manipulator with at least 3 independent axes that can move materials, parts, tools or devices along programmed, variable paths to perform a task.
- Context science (Christaller et al. 2001)
  - Robots are sensor-motoric machines for expanding human ability to act. They consist of mechatronic components, sensors and computer-based control functions. The complexity of a robot differs significantly from other machines in the greater number of degrees of freedom and the variety and scope of its behavior.

Robotics

- Interdisciplinary branch of engineering and science
- Includes among others
  - mechanical engineering
  - electronic engineering
  - information engineering
- Deals with the design, construction, operation, and use of robots, as well as computer systems for their control, sensory feedback, and information processing.

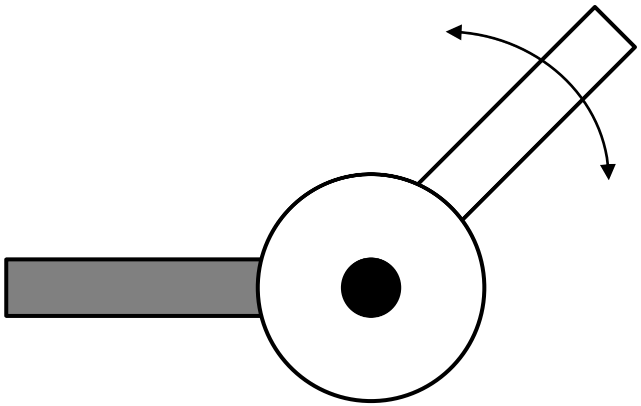
Examples

Components of a robotic system

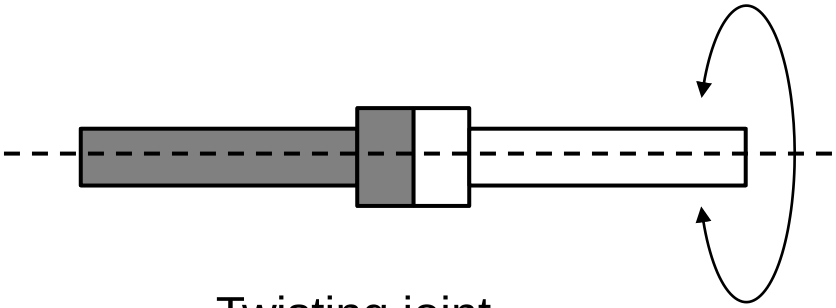
- Mechanical components
  - **Joint types**
  - **Workspace**
  - Wheel configurations
- Drive system
  - Fluid drive (e.g. hydraulic)
  - Muscular drive
  - Electric drive
- Gear drive

- Sensors
  - Internal sensors (Encoders, speedometer, temperature, forces, …)
  - External sensors (Cameras, microphone, laser, ultrasound, …)

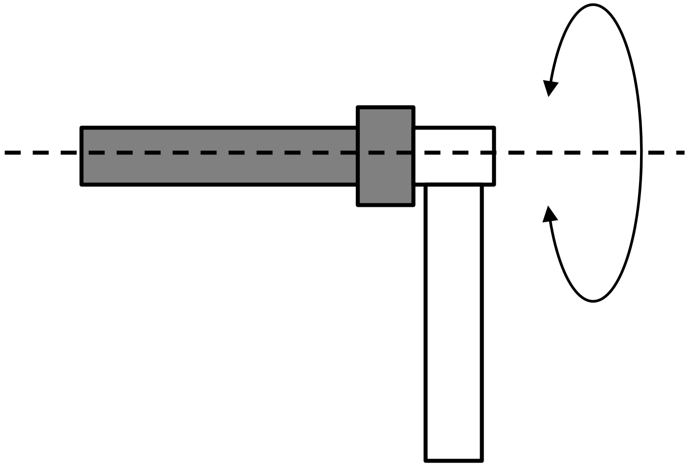
Components of a robotic system: Joint types



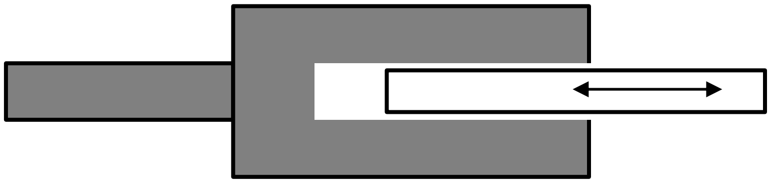
Rotational joint



Twisting joint



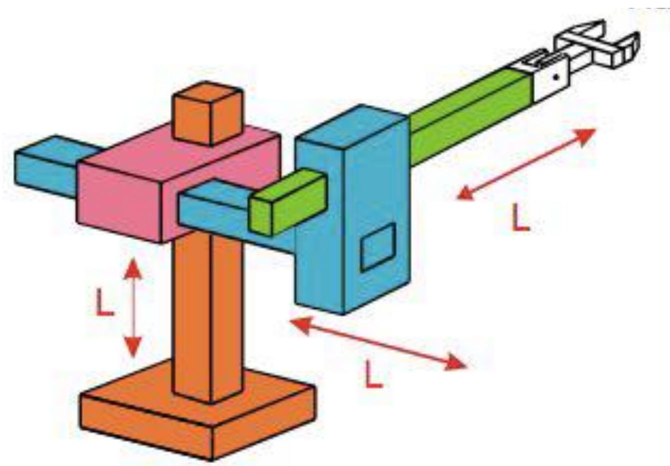
Revolving joint



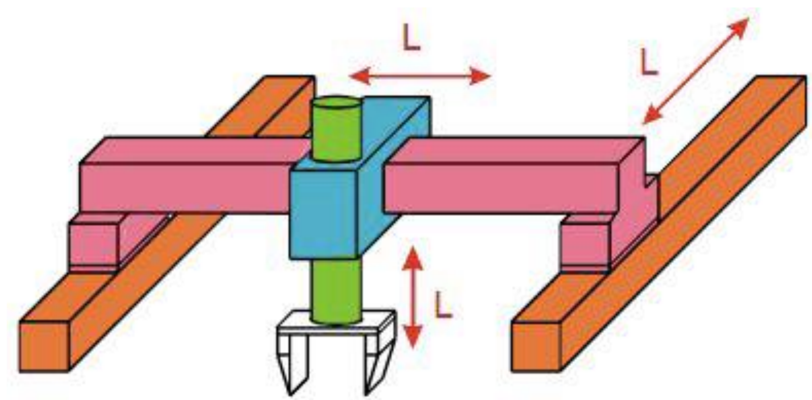
Linear joint

Components of a robotic system: Workspace

- The points in 3D space that can be reached with the end effector
- The basic form of the workspace results when collisions between robot arms and the angle range boundaries of the joints are not considered



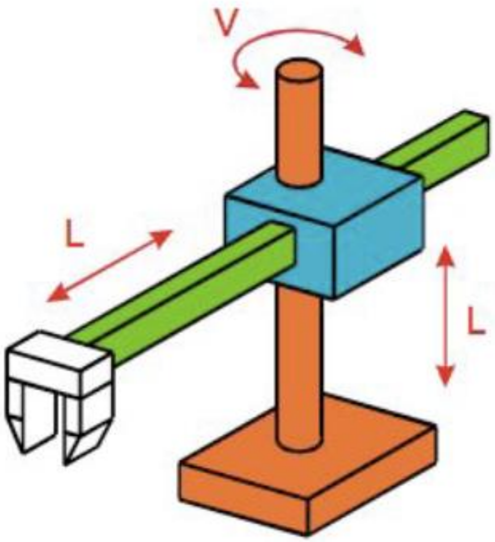
Workspace: Cuboid



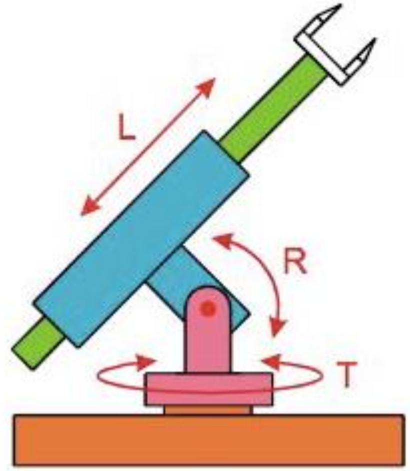
Workspace: Cuboid

Components of a robotic system: Workspace

- The points in 3D space that can be reached with the end effector
- The basic form of the workspace results when collisions between robot arms and the angle range boundaries of the joints are not considered

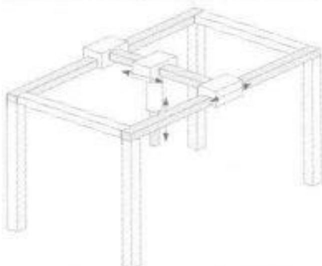
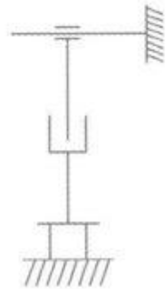
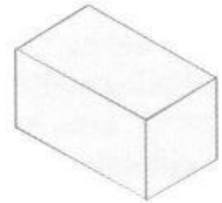
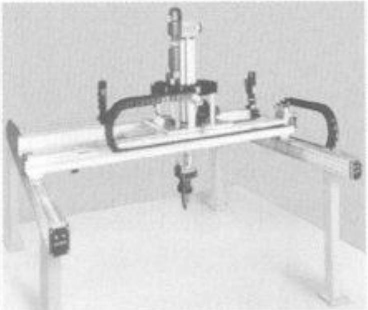
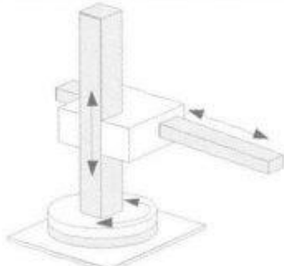
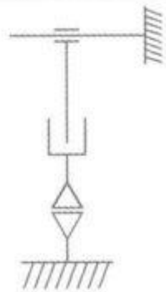
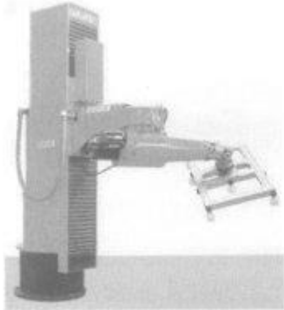
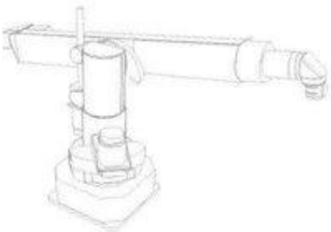
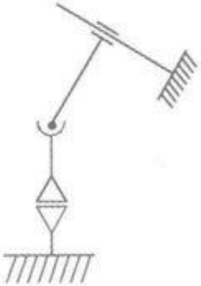
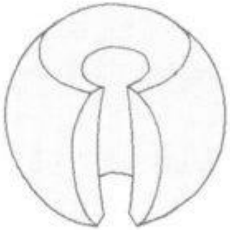
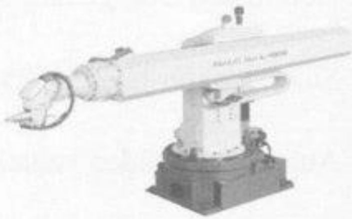


Workspace: Hollow cylinder

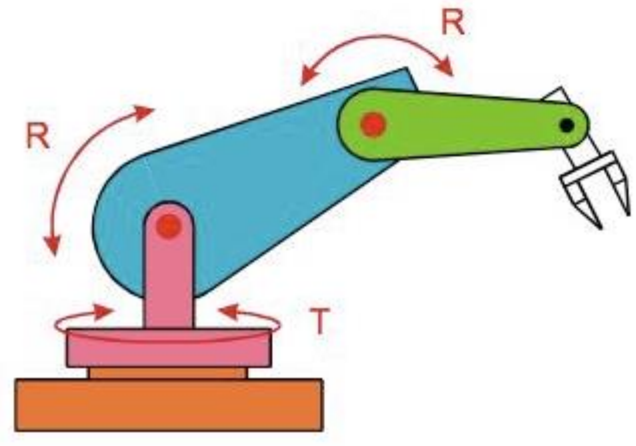


Workspace: Hollow sphere

Components of a robotic system: Workspace

Robot	Axes		Examples
Principle	Kinematic Structure	Workspace	Photo
 Cartesian Robot			
 Cylindrical Robot			
 Spherical Robot			

Workspace
What shape does the workspace of this robot have?



- A: Cube
- B: Rectangle
- C: Hollow cylinder
- D: Hollow sphere

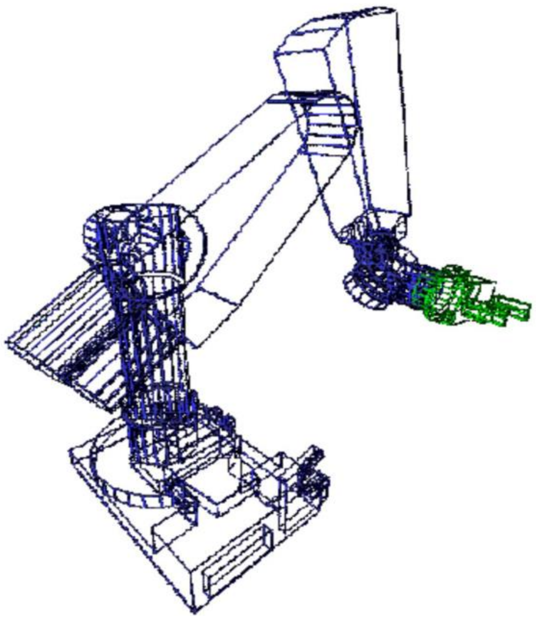
Modelling robotic systems

- Geometric model
  - Describing the shape of the robot
  - For measuring distances and collision detection
  - Basis for calculation of movements
  - Basis for determining forces and momentum
- Kinematic model
  - Kinematics: branch of mechanics that geometrically and analytically describes motion of a system
  - Describes the relationship between the space of joint angles and the space of end effector poses
- Dynamic model
  - Analyses of the movement of a body as a result of forces and momentum
  - Describes the relationship between forces, momentum and movements in a mechanical multi-body system

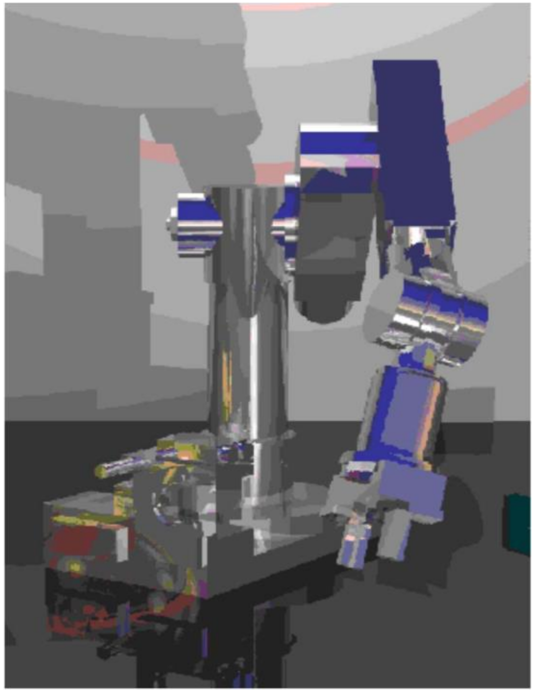
Modelling robotic systems: Geometric model

Classification

- By space
  - 2D model
  - 2.5D model
  - 3D model
- By geometric primitive
  - Wire frame model
  - Surface model
  - Volume model



Wire frame model



Volume model

Modelling robotic systems: Kinematic model

Two main problems

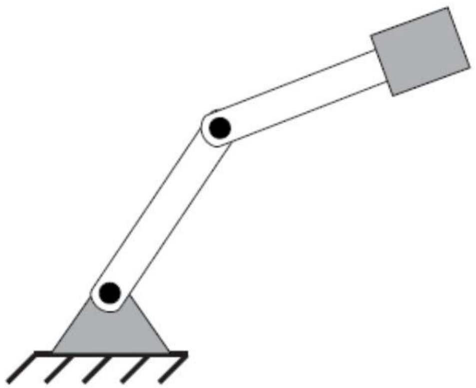
- Forward kinematics
  - Determining position of end effector from the joint angles of the robot

- Inverse kinematics
  - Determining joint angles for a specific position of the end effector

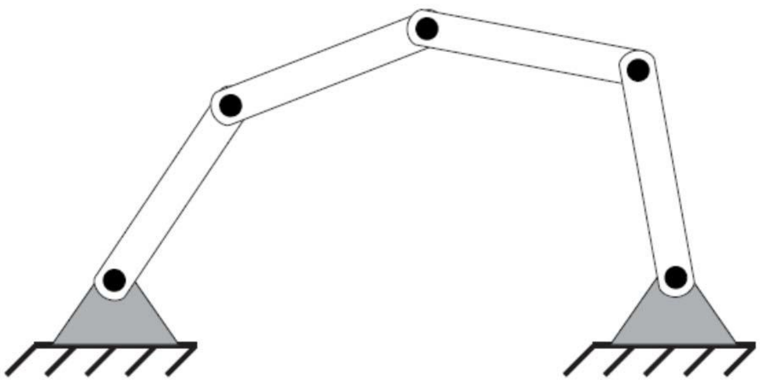
Kinematic model: Kinematic chain

- Definition: A kinematic chain is formed by multiple bodies that are kinematically connected via a joint, e.g. a robot arm

- Types



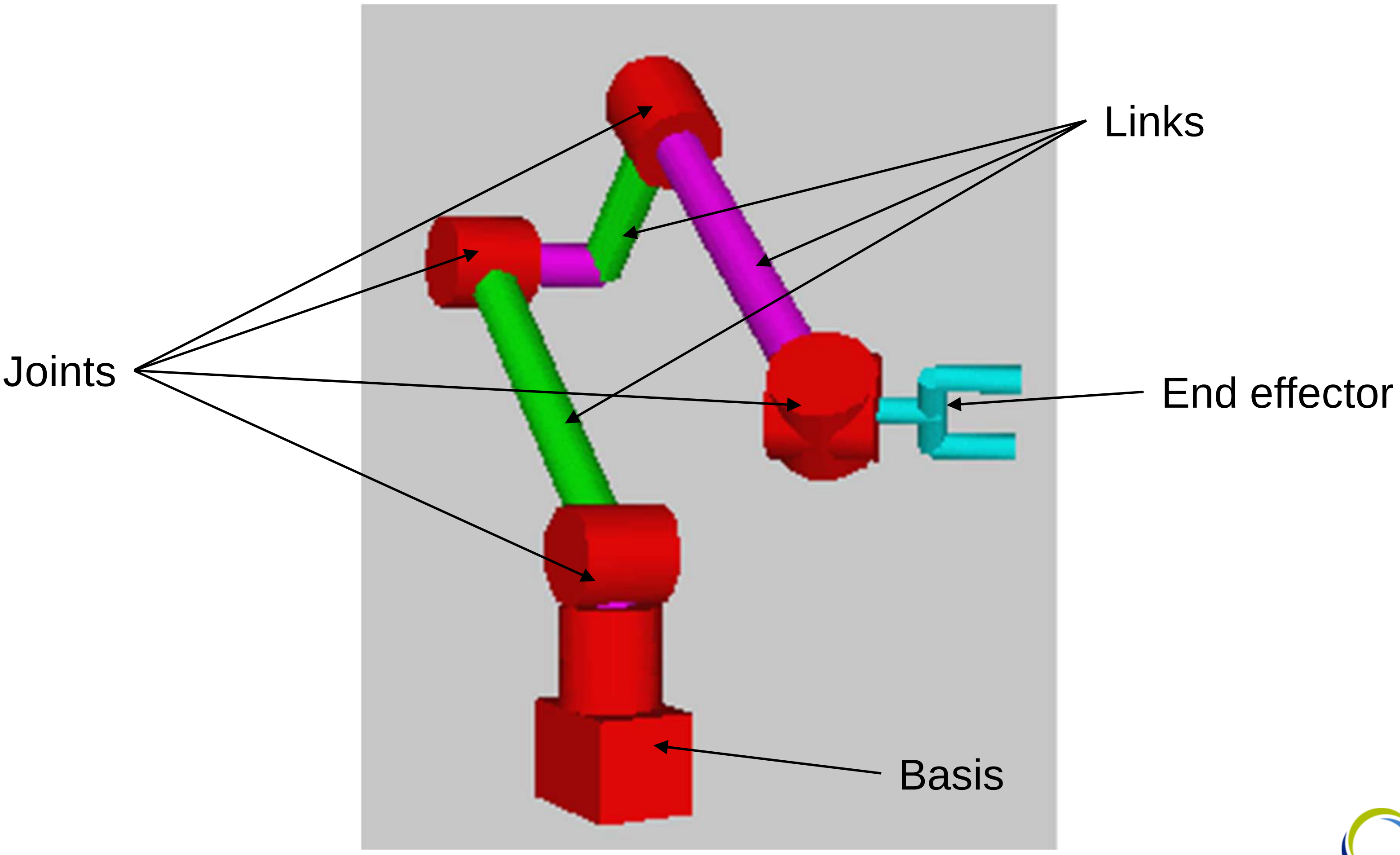
Open kinematic chain



Closed kinematic chain

Kinematic model: Kinematic chain

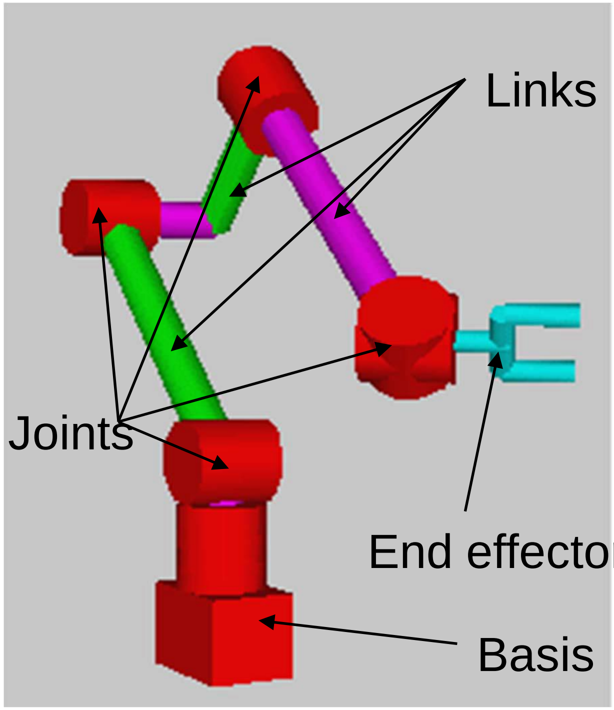
- Elements of a kinematic chain



Kinematic model: Kinematic chain

- Convention
  - Each link is a rigid body
  - Each link is connected to the next via a joint (rotational or translational)
  - Each joint only has one degree of freedom (rotational or translational)

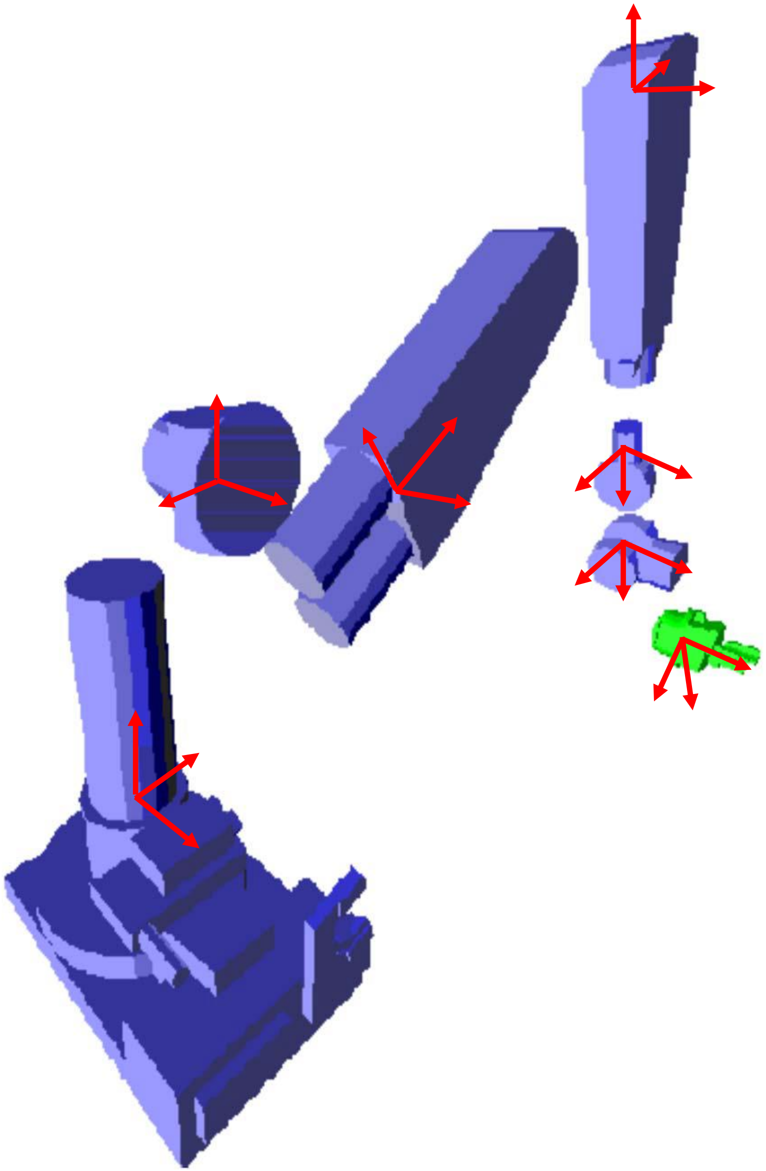
- Kinematic parameters = Joint & link parameters



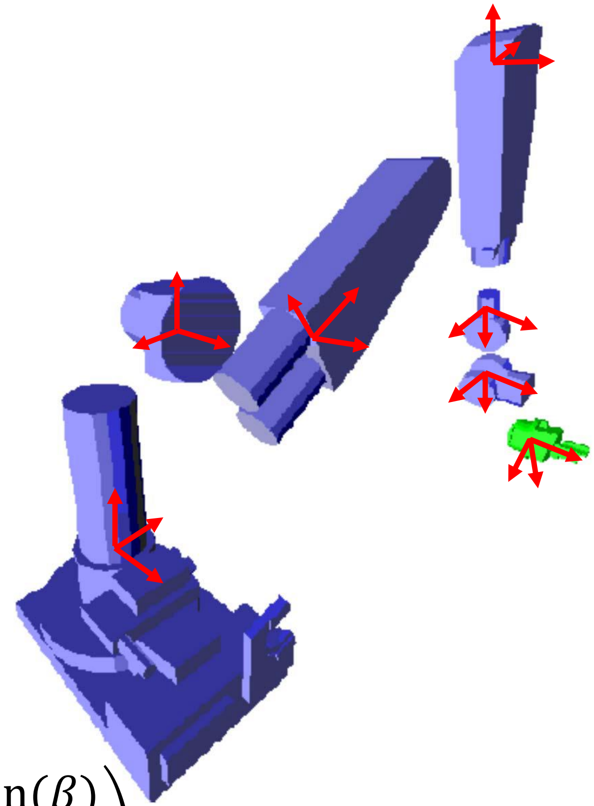
Kinematic model: Kinematic chain

Description of a kinematic chain

- Requires the pose of each link in relation to a basis coordinate system
- Each link has a fixed local coordinate system (Object coordinate system, OCS)
  - Robot basis OCS_{Basis}
  - Link 1 OCS_{Link1}
  - Link 2 OCS_{Link2}
  - … …
  - Link 6 OCS_{Link6}
- Origin of each OCS lies in the joint that moves the current link
- Each link requires a transformation from the OCS into the reference CS
- Transformation via, e.g., homogenous matrices



Kinematic model: Kinematic chain



- For each link, a transformation matrix to the reference coordinate system is required:
  - 3 rotational parameters
  - 3 translation parameters
  - $\Rightarrow$ 6 parameters per link
- E.g. via
  - Rotation matrix $\boldsymbol{R}$

$$R = R_z(\gamma)R_y(\beta)R_x(\alpha)$$

$$R_x(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{pmatrix} \quad R_y(\beta) = \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{pmatrix}$$

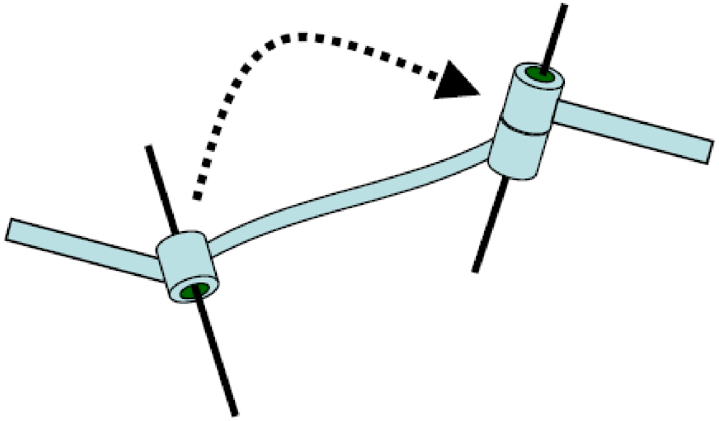
$$R_z(\gamma) = \begin{pmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

  - Translation vector $\boldsymbol{t}$ $t = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}$

Kinematic model: Denavit-Hartenberg convention

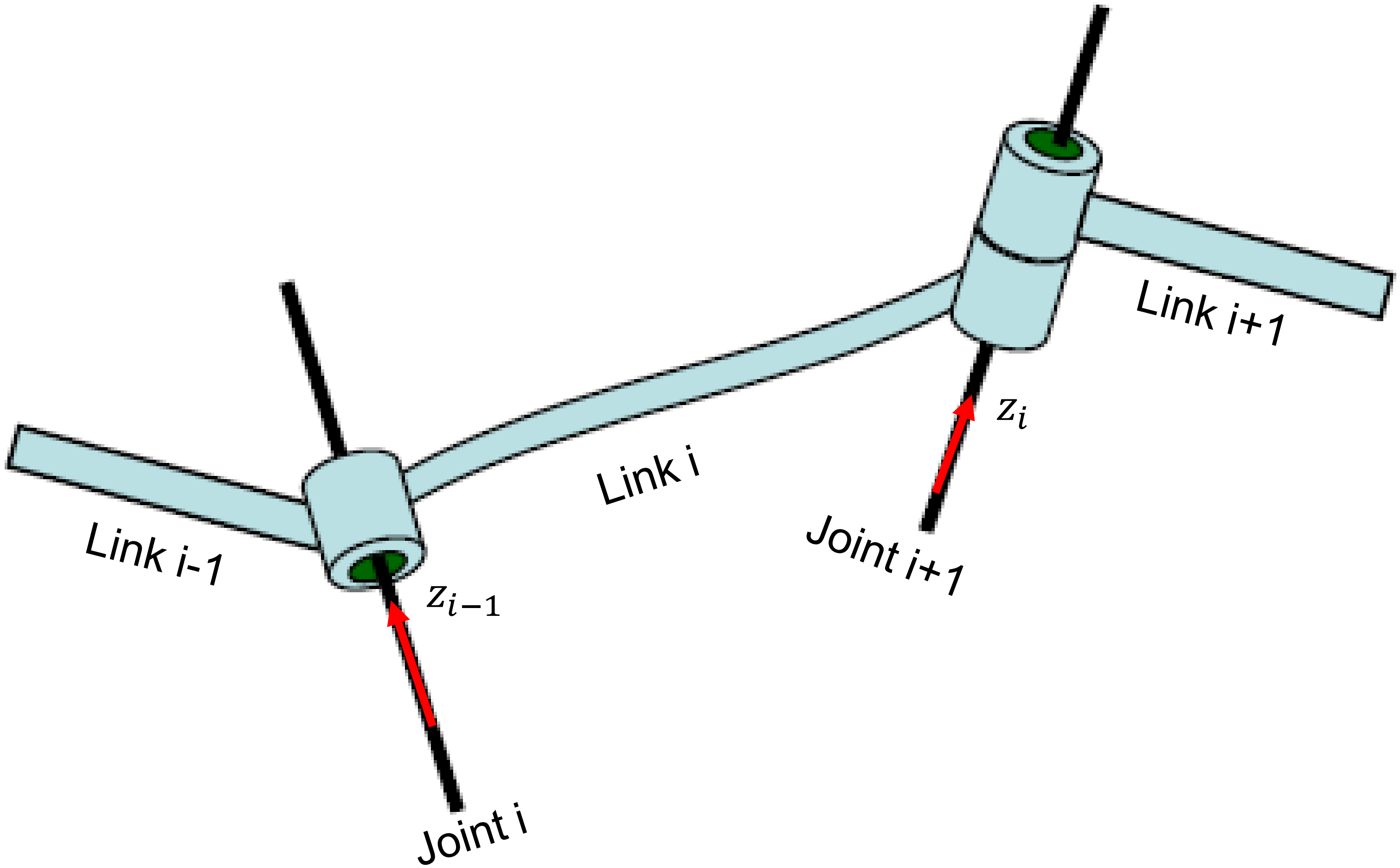
- Goal: reduction of parameters for describing a link with a joint

- Properties: Systemically describes the relationship (translation and rotation) between **neighboring** joints
- Reduces the number of parameters from 6 to 4



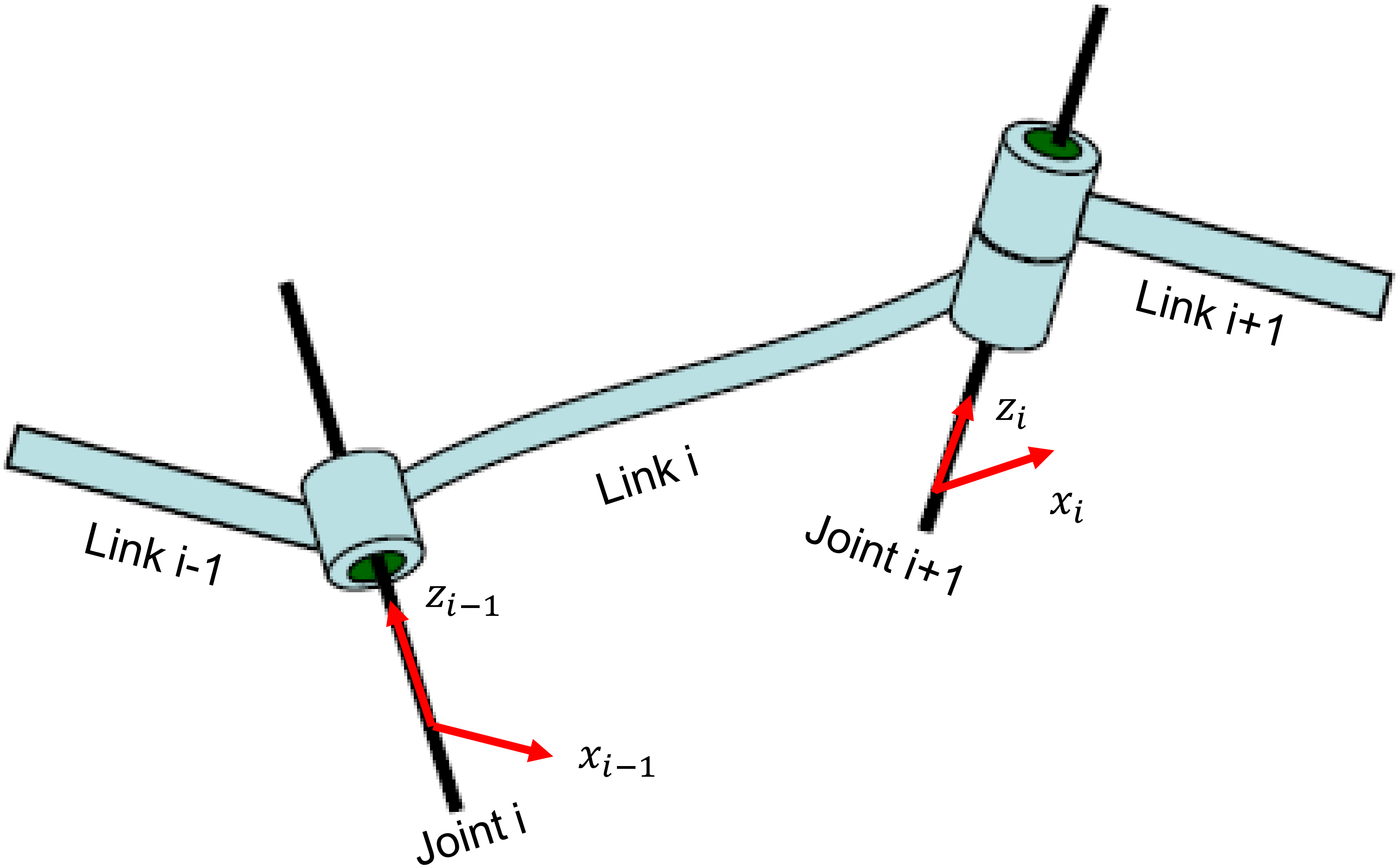
Kinematic model: Denavit-Hartenberg convention

- Axis z_{i-1} is in the direction of the joint axis



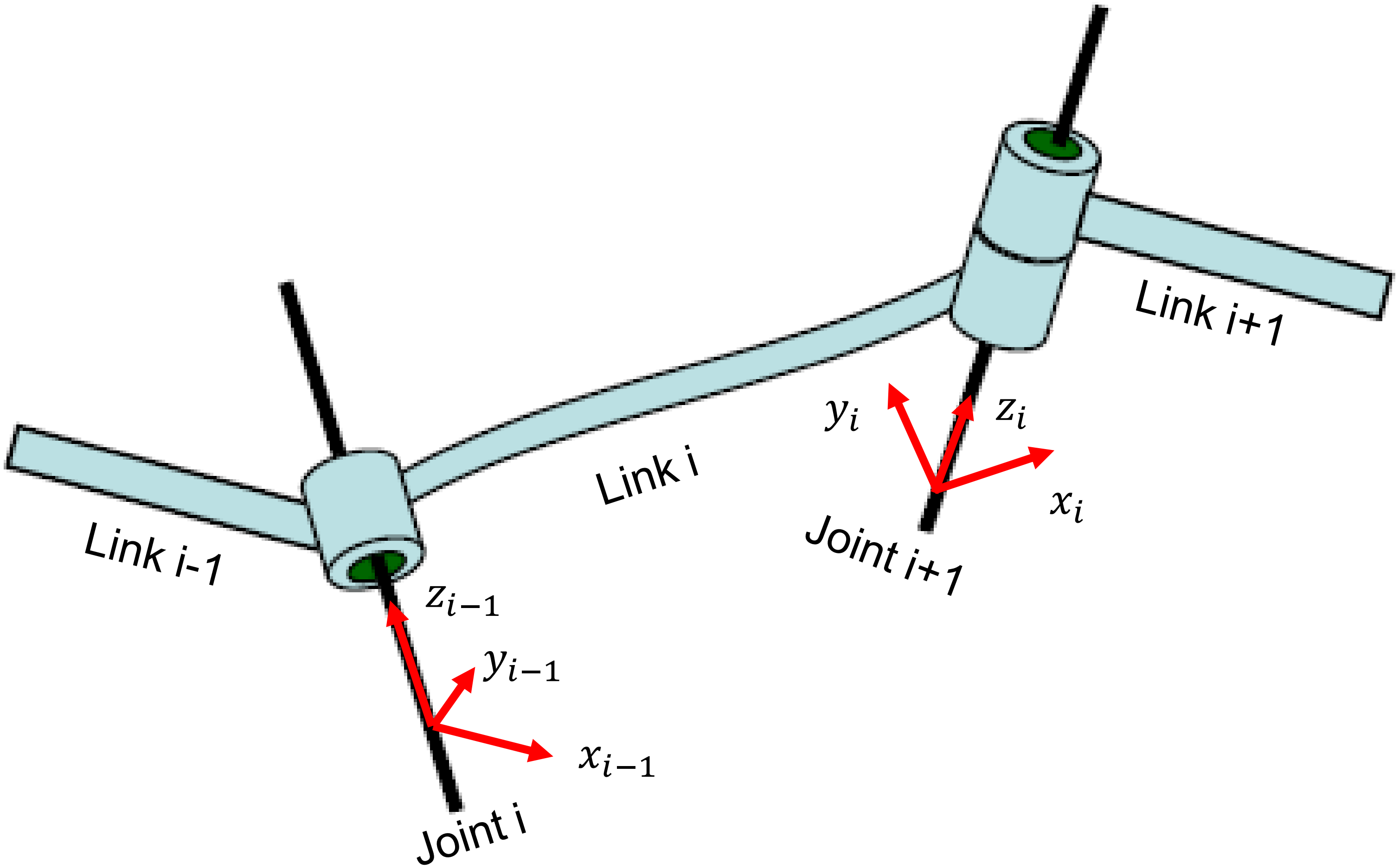
Kinematic model: Denavit-Hartenberg convention

- Axis x_i is orthogonal to z_i and z_{i-1}: $x_i = z_i \times z_{i-1}$
  - i.e. x_i is orthogonal to rotation axis and points along the link



Kinematic model: Denavit-Hartenberg convention

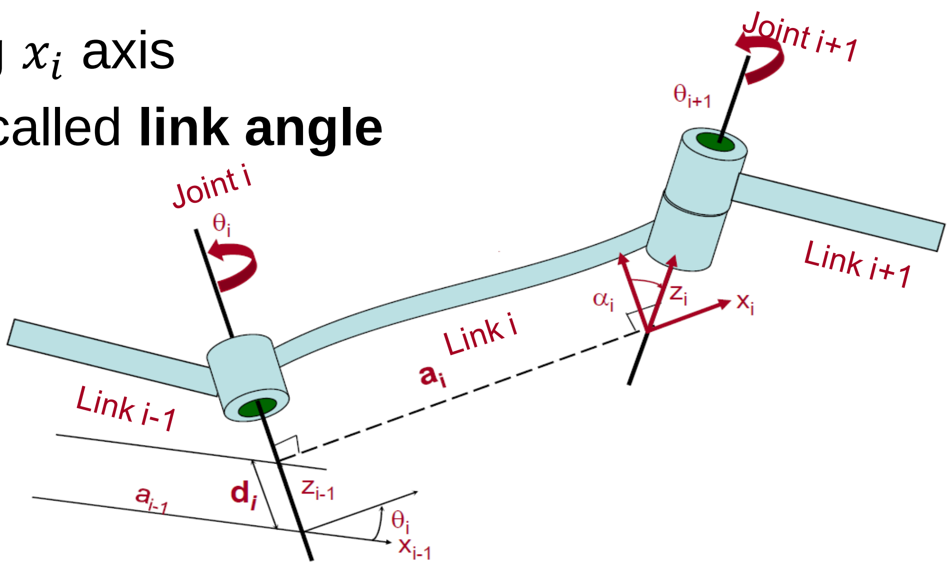
- Axis y_i is orthogonal to x_i and z_i: $y_i = x_i \times z_i$ (right-handed coordinate system)



Kinematic model: Denavit-Hartenberg convention

Parameters of the link

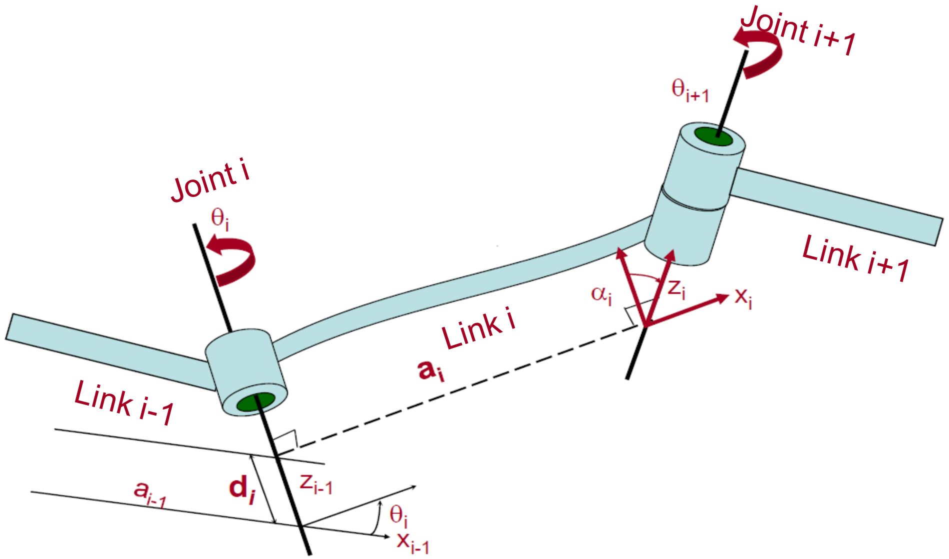
- Each link is connected via 2 neighboring joints i and i+1
- Let g_i and g_{i+1} be the axes of the joints (skewed to one another)
- Shared normal (base point distance) with the shortest distance is called **link length $\boldsymbol{a_i}$**
- Basis point of a_i with axis $g_{i+1}$is origin of the CS (x_i, y_i, z_i)
  - x_i axis: extension a_i to joint i+1
  - z_i axis: is in direction of g_{i+1} axis
  - y_i axis: completed according to right-hand rule
- Link length a_i is translation along x_i axis
- Angle $\boldsymbol{\alpha_i}$ between z_{i-1} and z_i is called **link angle**



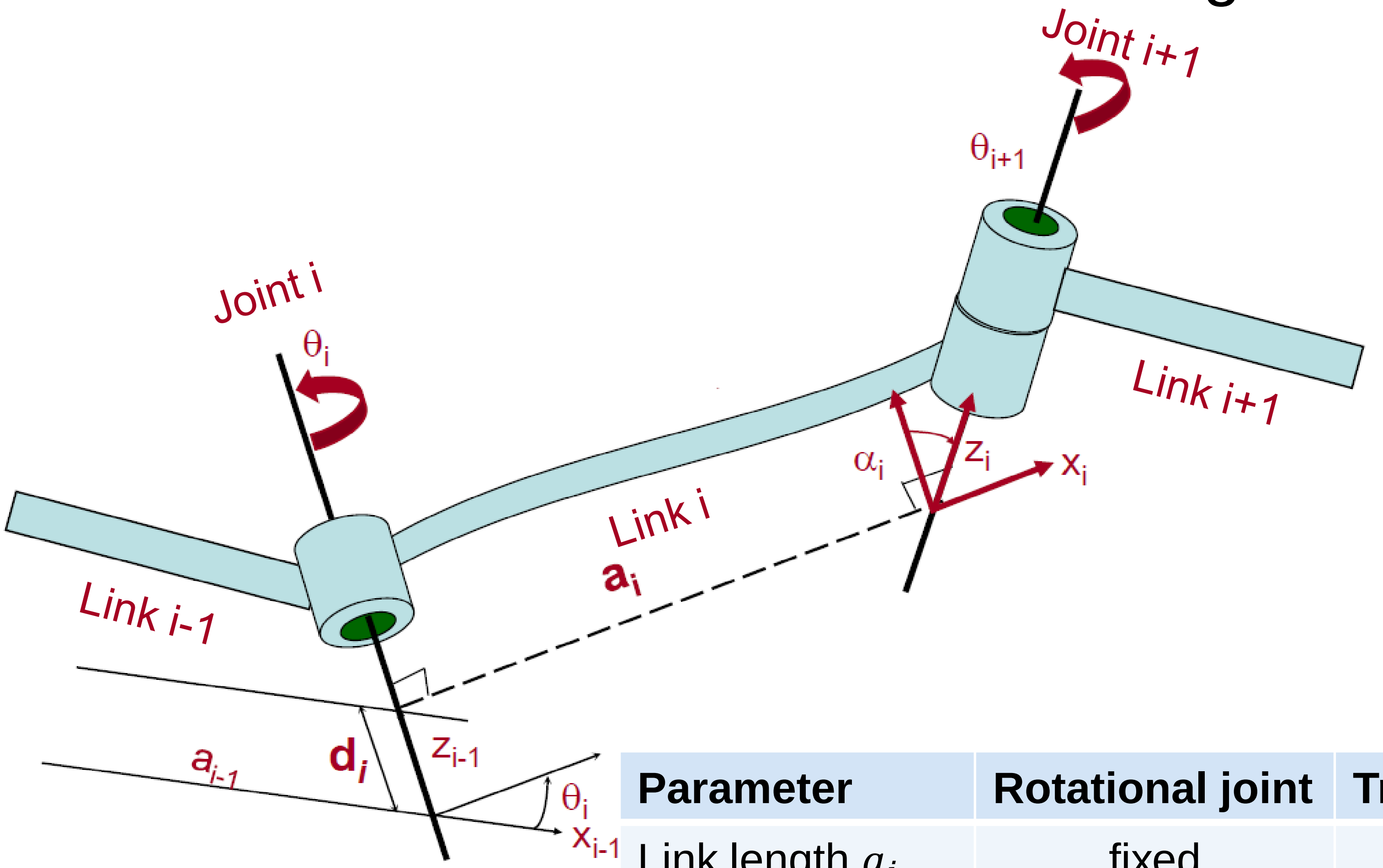
Kinematic model: Denavit-Hartenberg convention

Parameters of the joint

- Distance from origin i to origin i-1 along z_{i-1} is called **joint distance $\boldsymbol{d_i}$**
- d_i is translation along z_{i-1} axis so the x axes cut each other
- Rotation of link i around joint i with **joint angle $\boldsymbol{\theta_i}$** makes x_{i-1} and x_i parallel to one another



Kinematic model: Denavit-Hartenberg convention



Parameter	Rotational joint	Translational joint
Link length a_i	fixed	fixed
Link angle α_i	fixed	fixed
Joint distance d_i	fixed	variable
Joint angle θ_i	variable	fixed

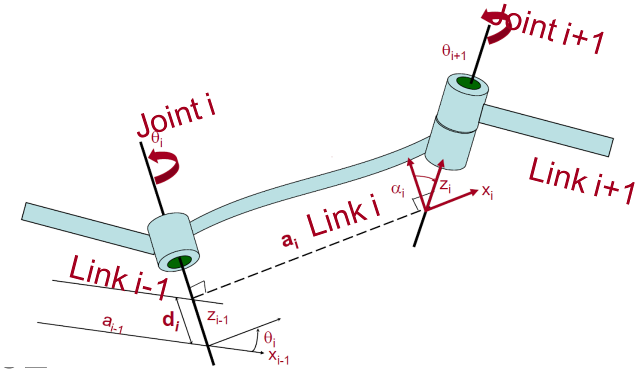
Kinematic model: Denavit-Hartenberg convention

Transformation from OCS_i to OCS_{i-1}

1. A rotation of θ_i around z_{i-1} axis, so x_{i-1} axis is parallel to x_i axis

$$R_{z_{i-1}}(\theta_i) = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2. A translation of d_i along z_{i-1} axis, so z_{i-1} axis and x_i axis cut each other



$$T_{z_{i-1}}(d_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

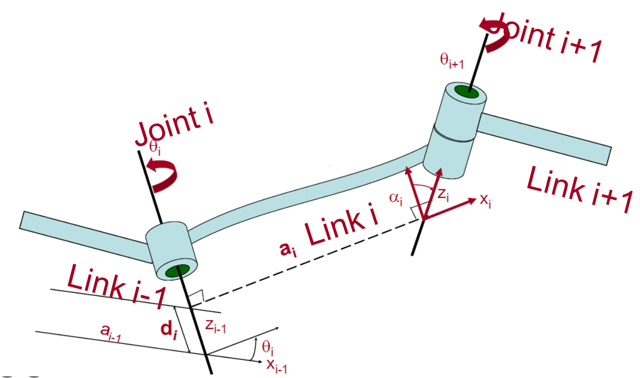
Kinematic model: Denavit-Hartenberg convention

Transformation from OCS_i to OCS_{i-1}

3. A translation of a_i along x_i axis so the origins of the coordinate systems match

$$T_{x_i}(a_i) = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

4. A rotation of α_i along x_i axis, so z_{i-1} axis and z_i axis match



$$R_{x_i}(\alpha_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Kinematic model: Denavit-Hartenberg convention

Transformation from OCS$_i$ to OCS$_{i-1}$

$$A_{i-1,i} = R_{z_{i-1}}(\theta_i) \cdot T_{z_{i-1}}(d_i) \cdot T_{x_i}(a_i) \cdot R_{x_i}(\alpha_i)$$

Forward kinematic:

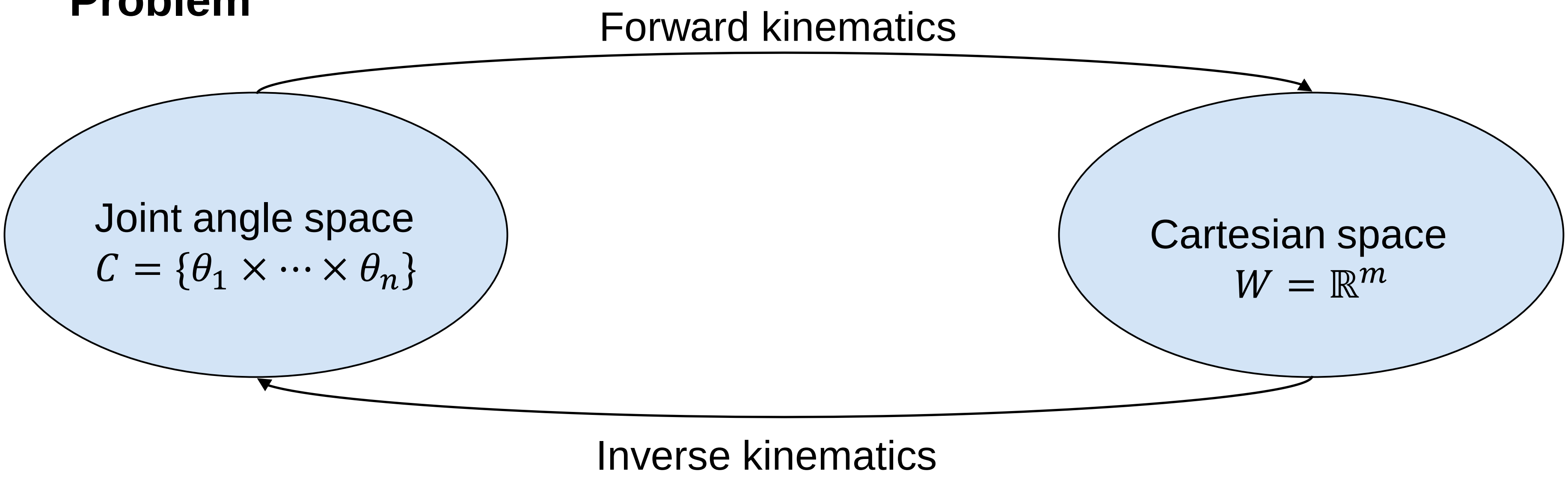
Transformation from OCS$_{basis}$ (i=0) to OCS$_{end\ effector}$ (i=n)

$$A_{0,n}(\theta) = A_{0,1}(\theta_1) \cdot A_{1,2}(\theta_2) \cdot \dots \cdot A_{n-1,n}(\theta_n) = \prod_{i=1}^{n} A_{i-1,i}\,(\theta_i)$$

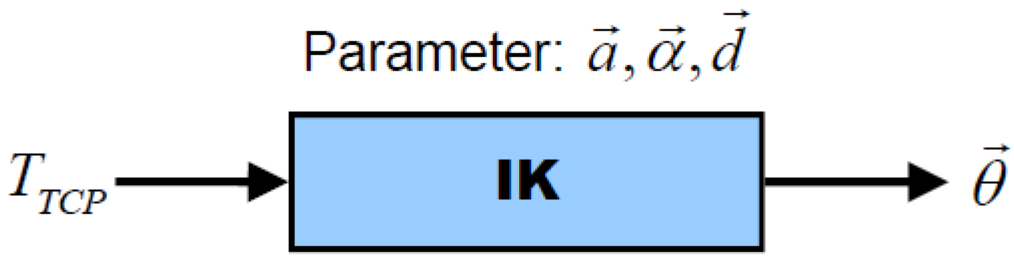
θ_i: Variable joint parameter (joint angle)

Kinematic model: Inverse kinematics

Problem



n: DOF movement, i.e. number of joints
m: DOF



Kinematic model: Inverse kinematics

Approach

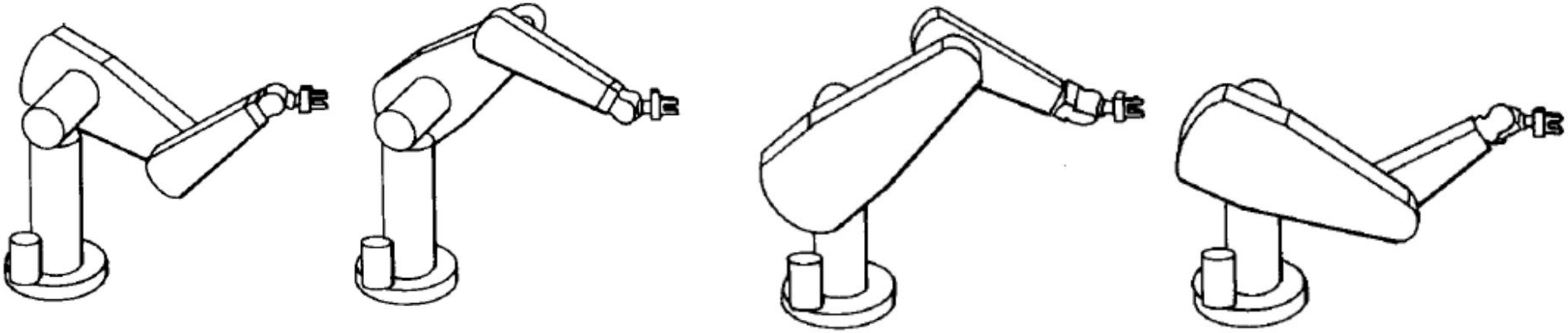
- Pose of end effector (T_{TCP}) is given
- Kinematic model

$$T_{TCP} = A_{0,n}(\theta) = \prod_{i=1}^{n} A_{i-1,i}\,(\theta_i)$$

- Solve for θ
- Non-linear problem

Kinematic model: Inverse kinematics

- Unreachable pose: Outside of the arm radius of the robot
- Invalid pose: In principle reachable, but not possible due to physical boundaries
  - Construction-based restraints (workspace)
  - Collisions with obstacles in workspace
  - Collision of handled objects or end effector with obstacles or the robot itself
- Uniqueness of configurations
  - In a plane, a system with $n \geq 3$ DOF in movement has multiple possibilities for reaching a given end effector pose
  - In 3D, the same is true for a robot with $n \geq 6$ DOF in movement



Same end effector pose with different configurations

Kinematic model: Inverse kinematics

Properties inverse kinematics

- No general applicable method
- Has to be fast

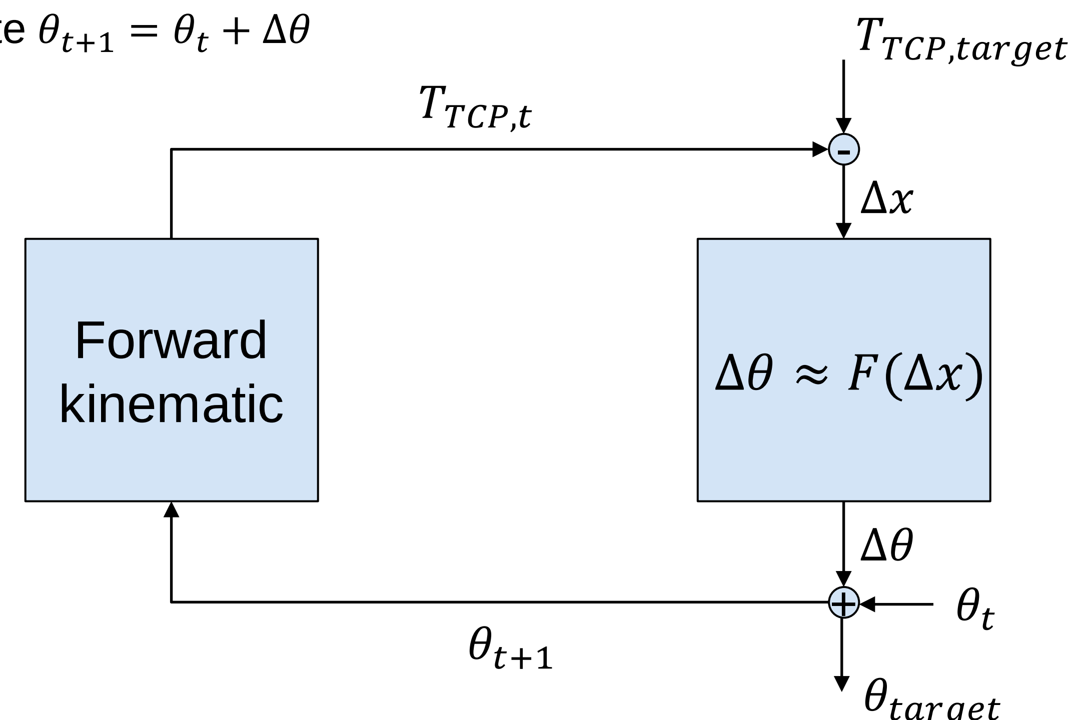
Solving

- Solution in closed form
  - Geometric methods
    - Trigonometric functions
    - Difficult to solve due to non-linear terms
  - Analytical methods
    - Solve analytically
    - 16 equations, 12 non-trivial
    - Often not enough to solve for all parameters
- **Numerical methods**

Kinematic model: Inverse kinematics

Numerical methods

- Iterative approach
  - Compute $T_{TCP,t}$ in iteration *t* from current joint angles θ_t (θ_0= actual position robot)
  - Compute error Δx $T_{TCP,t}$ and $T_{TCP,target}$
  - Use approximated inverse kinematic model *F* to compute joint angle error $\Delta\theta$
  - Compute $\theta_{t+1} = \theta_t + \Delta\theta$



Kinematic model: Inverse kinematics

Numerical methods: Approach Jacobian

1. Forward kinematics as function

$$x(t) = f(\theta(t))$$

$x(t)$: Pose end effector
$\theta(t)$: Joint angles

2. Derivative with respect to time

$$\frac{dx(t)}{t} = \dot{x}(t) = J(\theta)\dot{\theta}(t)$$

$\dot{x}(t)$: Velocity end effector
$\dot{\theta}(t)$: Joint angle velocities
$J(\theta)$: Jacobian matrix

3. Difference quotient

$$\Delta x \approx J(\theta)\Delta\theta$$

Δx: Error in end effector
$\Delta\theta$: Error in joint angles

4. Inversion

$$\Delta\theta \approx J^{-1}(\theta)\Delta x$$

Dynamic model

- Definition: Relationship between forces, momentum and motion that occur in a system
- Purpose
  - Analysis of dynamics
  - Synthesis of mechanical structures
  - Modelling of elastic structures
  - Controller design

$$Q = M(q) \cdot \ddot{q} + n(\dot{q}, q) + g(q) + R \cdot \dot{q}$$

- Q: Vector of forces and momentum
- $M(q)$: Mass inertia matrix
- $n(\dot{q}, q)$: Vector of centrifugal and Coriolis components
- $g(q)$: Gravitational components
- R: Diagonal matrix describing friction forces
- q: Angular position of the robot arm

Dynamic model

$$Q = M(q) \cdot \ddot{q} + n(\dot{q}, q) + g(q) + R \cdot \dot{q}$$

- Forward dynamic problem
  - Given external forces and momentums as well as initial state of the robot, compute movement changes

  => Solve for $q(\mathrm{t}), \dot{q}(t), \ddot{q}(t)$

- Inverse dynamic problem
  - Given target movement parameters, compute required forces and momentums

  => Compute *Q*

Path planning

- Definition: Methods for planning collision-free movements

- Movements of the robot are seen as state changes over time relative to a stationary coordinate system

- Given:
  - S_{Start}: State at initial position
  - S_{target}: State at end position
- Goal
  - S_i: intermediate states for a smooth and continuous movement

Path planning: Terms

- A **configuration q** describes the state of the robot
  - In Euclidean space through its pose
  - In joint angle space through the values of the joints
- The **configuration space C** is the space of all possible configurations of the robot
- A **path** is a continuous map from the configuration q_{start} to configuration q_{target}

$$\tau: [0,1] \rightarrow C$$

With $\tau(0) = q_{start}$, $\tau(1) = q_{target}$

- A **workspace obstacle *H_i*** is the space taken up by an object in workspace
- **Configuration space obstacle *CH_i*** is the set of all configurations that result into points in H_i
- **Obstacle space** is the set of all configuration space obstacles

$$C_{obst} = \bigcup CH_i$$

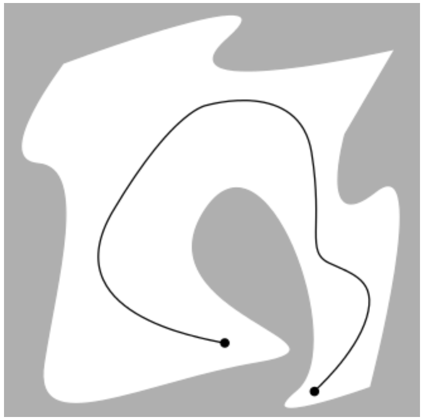
Path planning: Terms

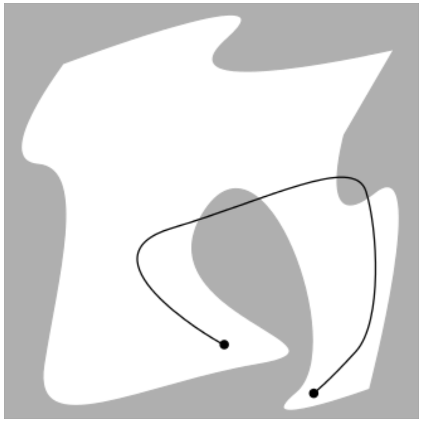
- **Free space** is the set of all points from *C* that are not in obstacle space

$$C_{free} = \{q \in C | q \notin C_{obst}\} = C \setminus C_{obst}$$

- Cost of computing free space: $O(m^n)$
  - With n: Movement DOF of the robot (number of joints)

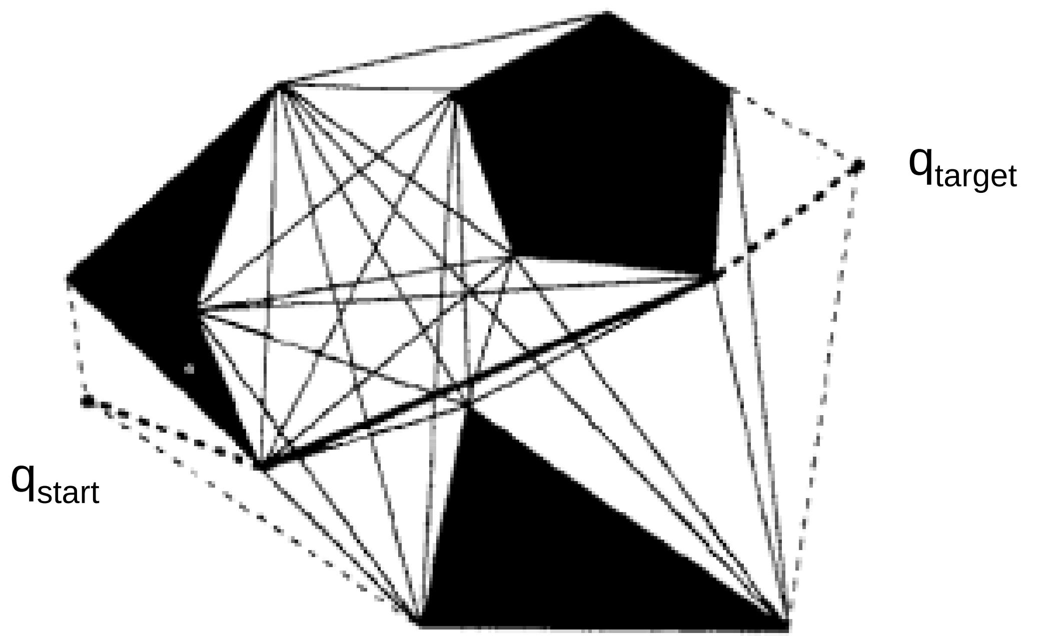
    m: Number of obstacles
  - Costly for complex (n>3) systems => Approximate free space
    - Voronoi diagrams
    - Visibility graphs
    - Quad- or Octtrees
    - Rapidly-exploring Random Trees



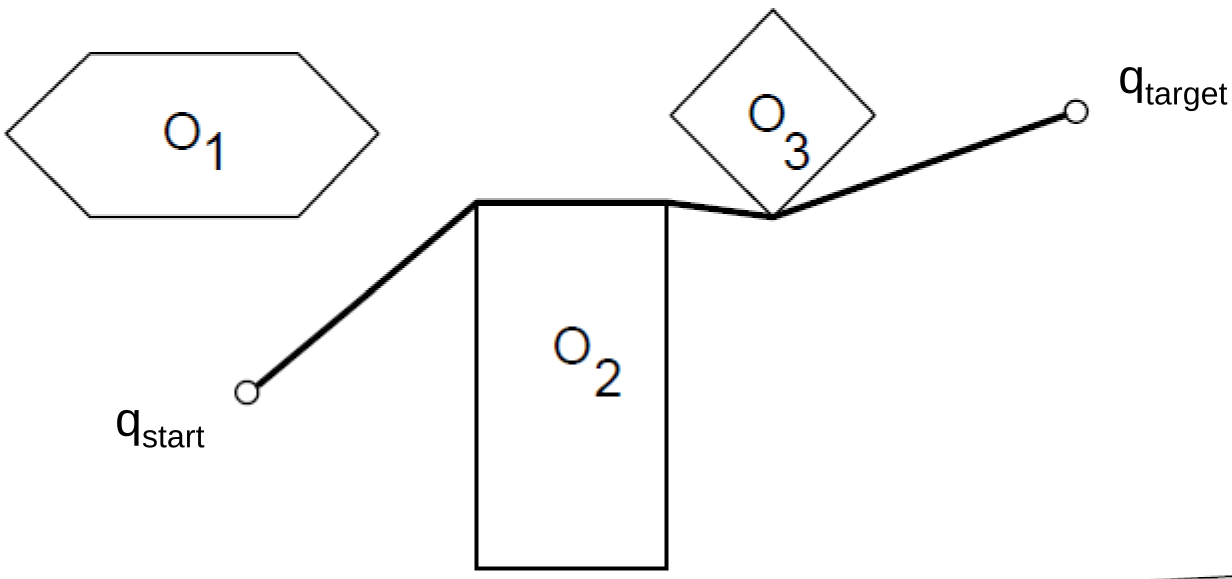


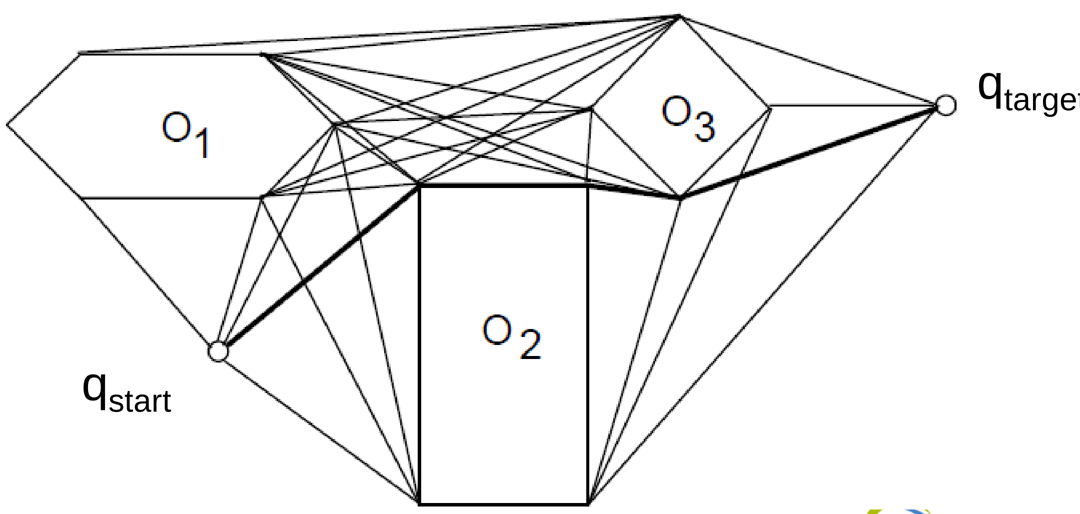
Path planning: Visibility graph

- **Construction**
  - Connect each pair of corner points on the edge of C_{free} if the line segment does not cross an obstacle
  - Connect q_{start} and q_{target}



Path planning: Visibility graph



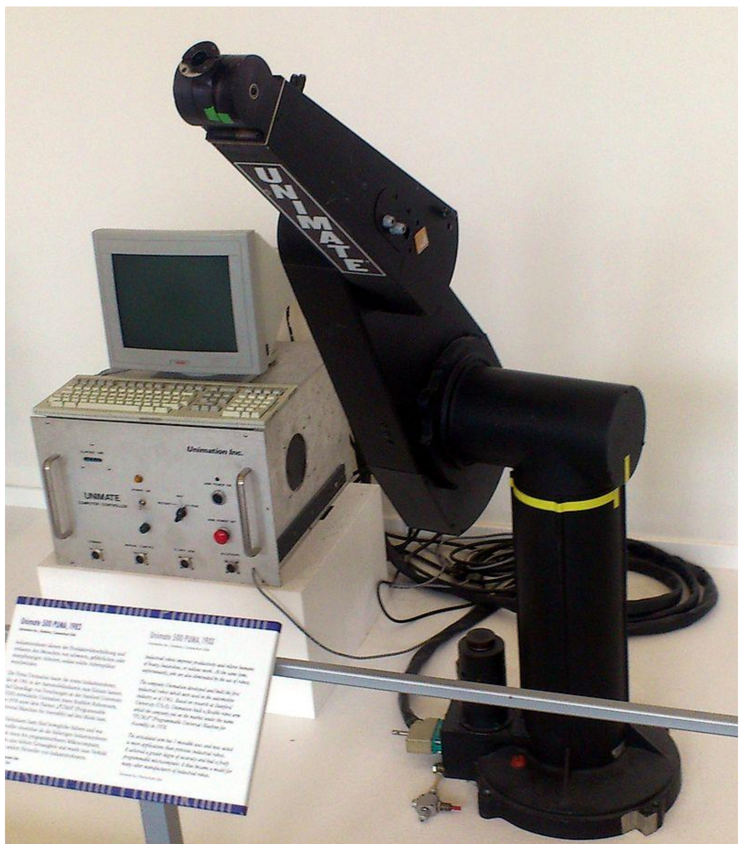


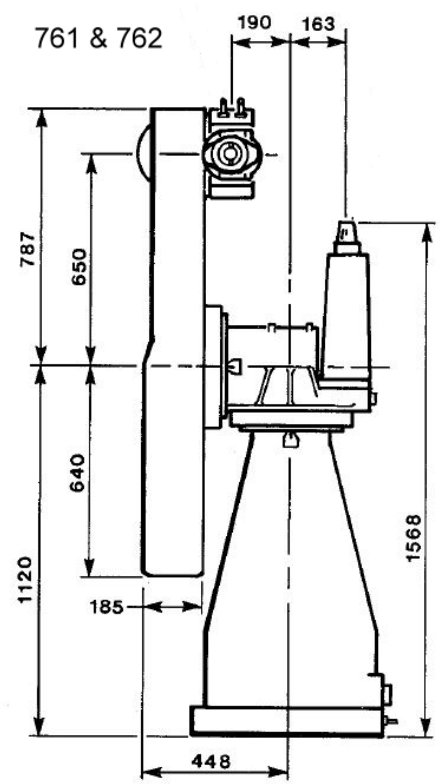
History of surgical robots: Arthobot (1985)

- From University of British Columbia
- Assistant for orthopedic surgery
- Changes leg position via voice command

History of surgical robots: Unimation Puma 200(1985)

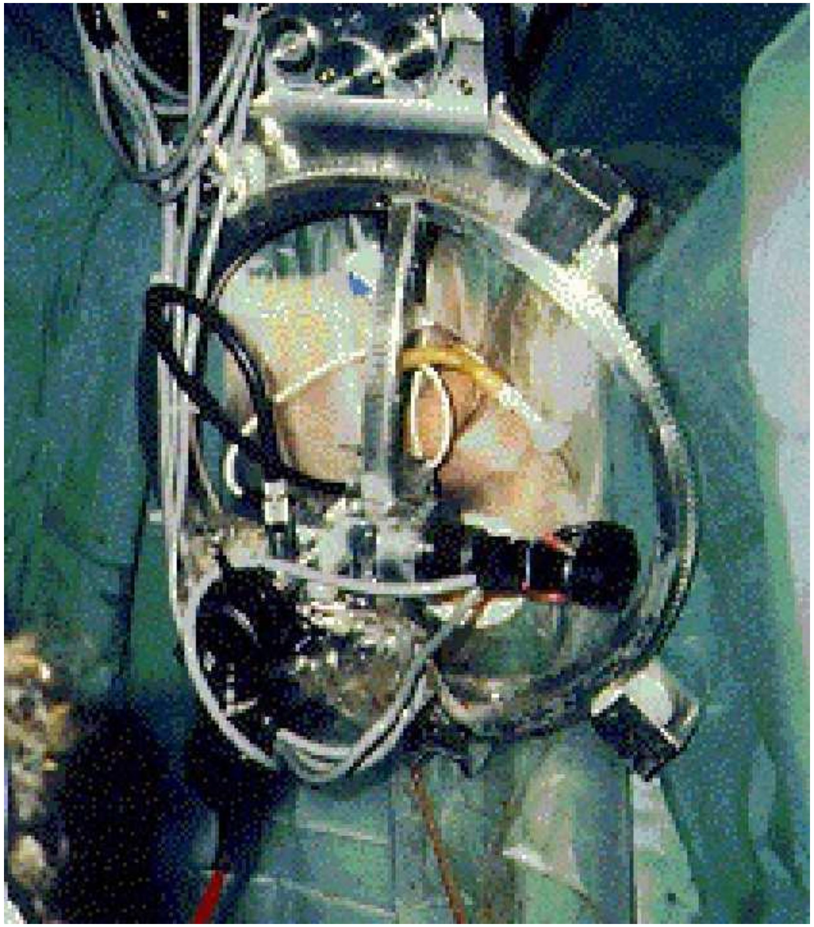
- Industrial robot
- Needle orientation for brain biopsy with CT guidance





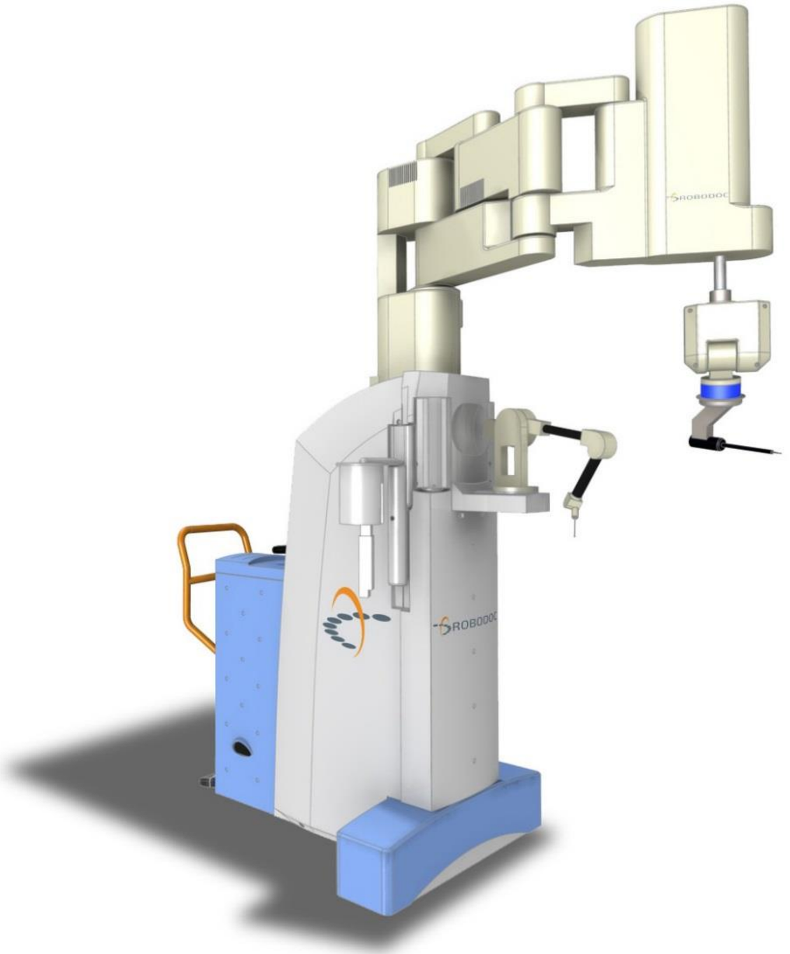
History of surgical robots: PROBOT (late 1980s)

- Robot for prostate resection
- Surgeon can specify within the prostate to be automatically cut



History of surgical robots: ROBODOC (1992)

- Robot for computer-guided milling of bone during hip replacment
- Needle orientation for brain biopsy with CT guidance



History of surgical robots: AESOP (1994)

- Automated Endoscopy System
- Robot for moving endoscope
- Voice activated

NCT

History of surgical robots: ZEUS (1998)

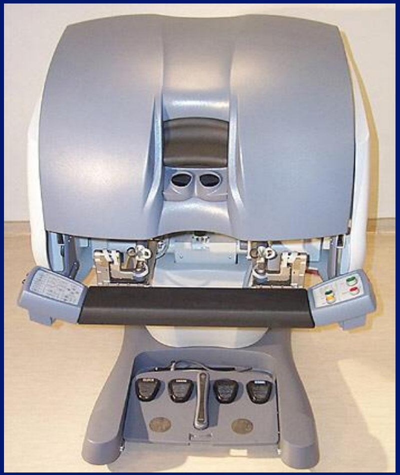
- Combines AESOP with two arms for holding instruments
- Tele-robot
- 3D vision

Zeus Robot Arms

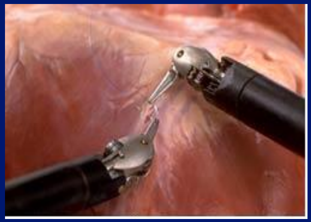
Zeus Console

History of surgical robots: da Vinci (2000)

Konsole



Manipulator



Quelle: Intuitive Surgical

DaVinci

Current Systems for minimal invasive surgery

- Versius Surgical Robotic System by CMR Surgical

- Senhance Surgical System by Asensus Surgical

NCT

Current Systems for minimal invasive surgery

- Hugo Robotic Assisted Surgery System by Medtronic