Robotics 1

notes

Tomáš Sláma
30. 11. 2022

Contents

Preface

This website contains my lecture notes from a lecture by Lorenzo Masia from the academic year 2022/2023 (University of Heidelberg). If you find something incorrect/unclear, or would like to contribute either text or an image, feel free to submit a pull request (or let me know via email).

Note that since the course hasn't ended yet, the notes are not complete and won't be until the end of the semester.

Robotics Components

Definition (link/member) individual bodies making up a mechanism

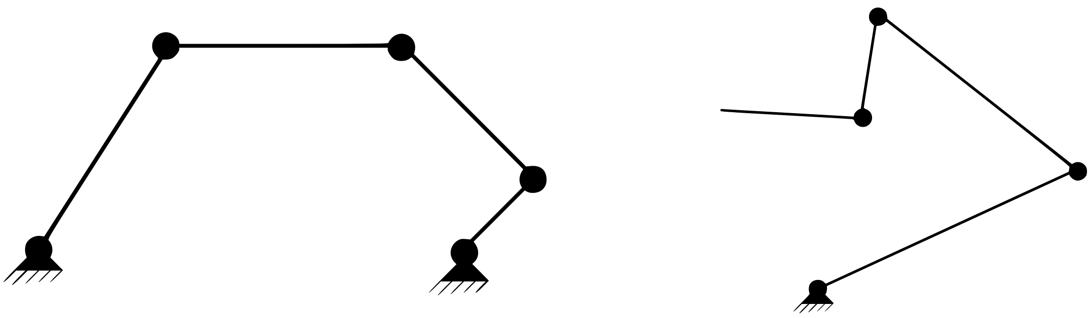
Definition (joint) connection between multiple links

Definition (kinematic pair) two links in contact such that it limits their relative

- **low-order** kinematic pairs: point of contact is a surface (eg. revolute/prismatic/spherical)
- **high-order** kinematic pairs: point of contact is a dot/line (eg. gears)

Definition (kinematic chain) assembly of links connected by joints

- is a **closed loop** (left image) if every link is connected to every other by at least two paths (including the ground – imagine it as a single link), else it's **open** (right image)



Degrees of Freedom

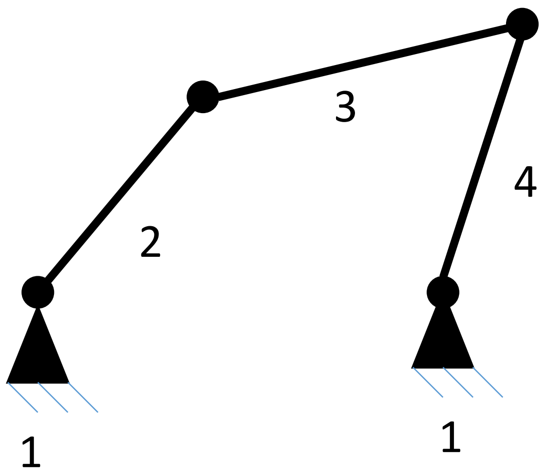
Definition (deegree of freedom) of a mechanical system is the **number of independent parameters** that define its configuration. It can be calculated by the **Grübler Formula**:

$$\mathrm{DOF} = \lambda(n-1) - \sum_{i=1}^{j} c_i$$

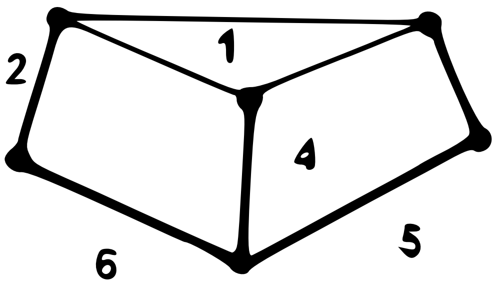
where

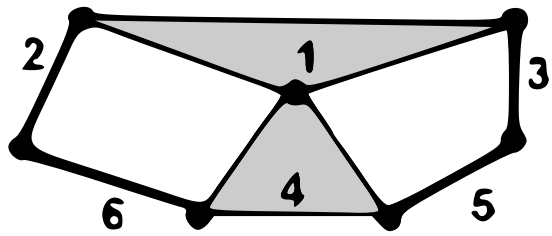
- $\lambda \ldots$ DOF of the operating space (3 for 2D, 6 for 3D)
  - in 2D it's 2 for orientation and 1 for position
  - in 3D it's 3 for orientation and 3 for position
- $n \ldots$ number of links
- $j \ldots$ number of kinematic pairs
- $c_j \ldots$ degree of constraint of the i-th kinematic pair
  - eg. a revolute joint in 2D takes away 2 DOF (we can only rotate)

For example, the following has $3(4-1) - 2 \cdot 4 = 1$ DOF:



Definition (coincident joints) when there are more than two kinematic pairs in the same joint

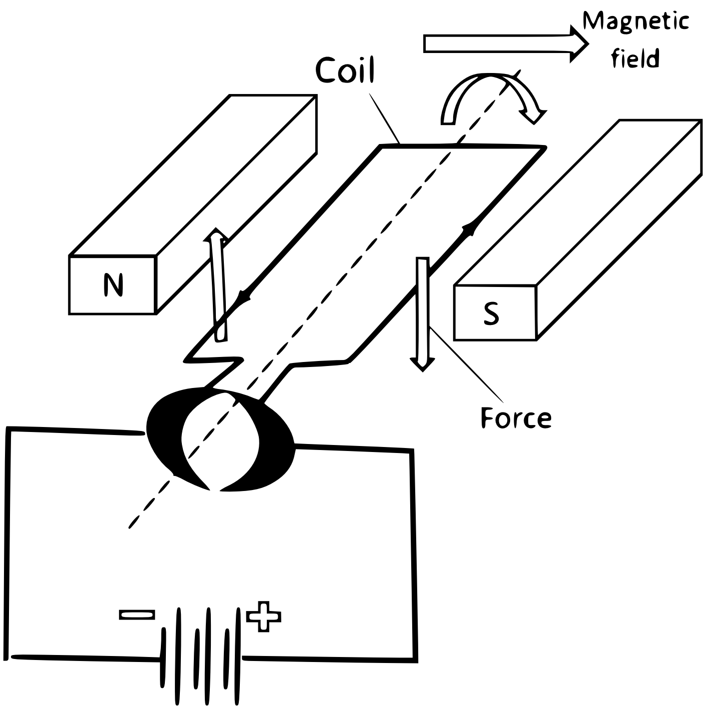




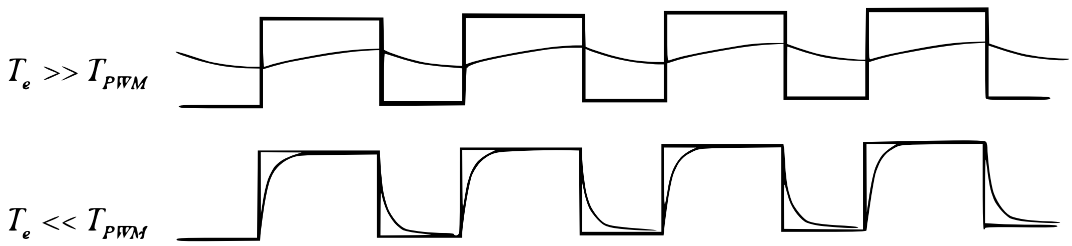
Actuators

Definition (actuator) a mechanical device for moving or controlling something

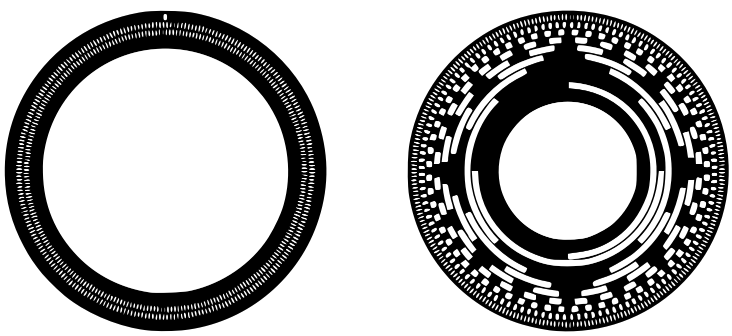
- **DC brushed motor**: based on Lorentz' force law (electromagnetic fields)
  - **brushed** because the metal brush powers the magnets (they're turning)
  - **brushless** uses the position of the motor to turn on/off currents for specific windings
  - usually contains **gears reductions** to trade torque for speed and **sensors** to measure the position of the motor (see further)



- since the voltage controls the motor but setting it to a specific value is impractical, **pulse width modulation (PWM)** is used:

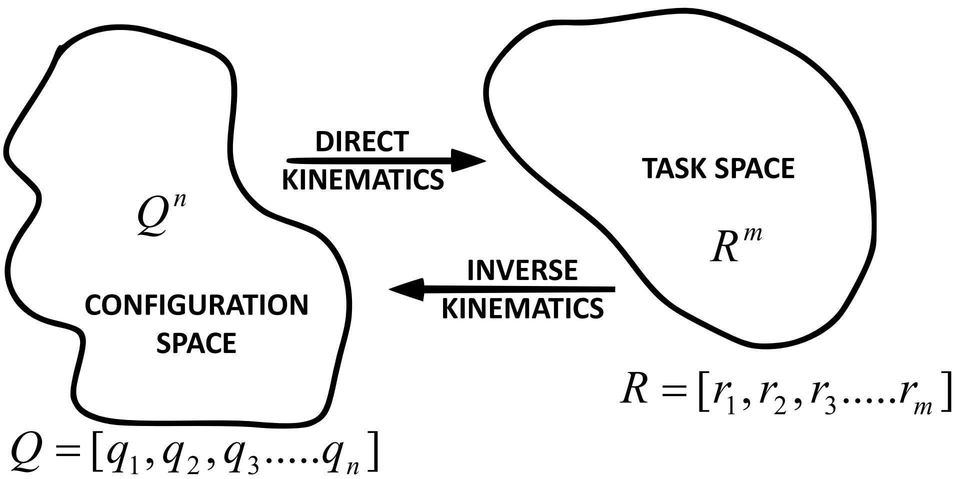


- to measure the position of the motor, **optical shaft encoders** (incremental/absolute) are used:



Kinematics

- establishment of various coordinate systems to represent the positions and orientations of rigid objects and with transformations among these coordinate systems



- the dimension of the **configuration space** (n) must be **larger or equal** to the dimension of the **task space** (m) to ensure the existence of kinematic solutions

Forward (direct) Kinematics

Definition (forward/direct kinematics) the process of finding the position/orientation of the **end-effector** $(r_1, \ldots, r_m)$ given a set of joint parameters $(q_1, \ldots, q_n)$.

$$(r_1, \ldots, r_m) = F(q_1, \ldots, q_n)$$

Body Pose

The **pose/frame** of a rigid body can be described by its **position** and **orientation** (wrt. a reference frame).

- the **position** is a vector $P \in \mathbb{R}^3$
- the **rotation** is an **orthonormal matrix** $R \in \mathbb{R}^{3\times 3}$ with $\det(R) = 1$ (a determinant of -1 would flip the object, we only want rotation)
  - due to orthogonality: $R^T = R^{-1}$

$$R = \begin{bmatrix} x' & y' & z' \end{bmatrix} = \begin{bmatrix} x'_x & y'_x & z'_x \\ x'_y & y'_y & z'_y \\ x'_z & y'_z & z'_z \end{bmatrix}$$

The elementary rotations about each of the axes are the following:

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$
$$R_y(\alpha) = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix}$$
$$R_z(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- important to remember for calculating robot kinematics (or just think about what the rotation about an axis is doing to the other two axes, it's not too hard to remember)
- positive values for the rotation are always **counter-clockwise**

When discussing multiple frames, we use the following notation:

$${}^{\text{to}}_{\text{from}}R = {}^{\text{to}}R_{\text{from}} = R^{\text{to}}_{\text{from}}$$

E.g. if we have the same point p_0, p_1, p_2 in three different frames, we know that

$$\begin{aligned} p_1 &= R_2^1 \cdot p_2 \\ p_0 &= R_1^0 \cdot p_1 \\ p_0 &= R_2^0 \cdot p_2 \end{aligned}$$

For a minimal representation, we will use **Euler's ZYZ angles**, which does three rotations:

$$\begin{aligned} R &= R_z(\varphi) R_{y'}(\vartheta) R_{z''}(\psi) \\ &= \begin{bmatrix} c_\varphi c_\vartheta c_\psi - s_\varphi s_\psi & -c_\varphi c_\vartheta s_\psi - s_\varphi c_\psi & c_\varphi s_\vartheta \\ s_\varphi c_\vartheta c_\psi + c_\varphi s_\psi & -s_\varphi c_\vartheta s_\psi + c_\varphi c_\psi & s_\varphi s_\vartheta \\ -s_\vartheta c_\psi & s_\vartheta s_\psi & c_\vartheta \end{bmatrix} \end{aligned}$$

For the inverse problem (calculating angles from a matrix of numbers), we can do

$$\begin{aligned} \varphi &= \operatorname{atan2}(r_{2,3}, r_{1,3}) \\ \vartheta &= \operatorname{atan2}\left(\sqrt{r_{1,3}^2 + r_{2,3}^2}, r_{3,3}\right) \qquad \vartheta \in (0, \pi) \text{ since we took + sign} \\ \psi &= \operatorname{atan2}\left(r_{3,2}, -r_{3,1}\right) \end{aligned}$$

- if we divide by zero somewhere we get degenerate solutions where we can only get the sum of the angles (one of the problems with Euler angles)
- alternative is **RPY angles**, which are also three rotations (roll, pitch, yaw) and are just as bad

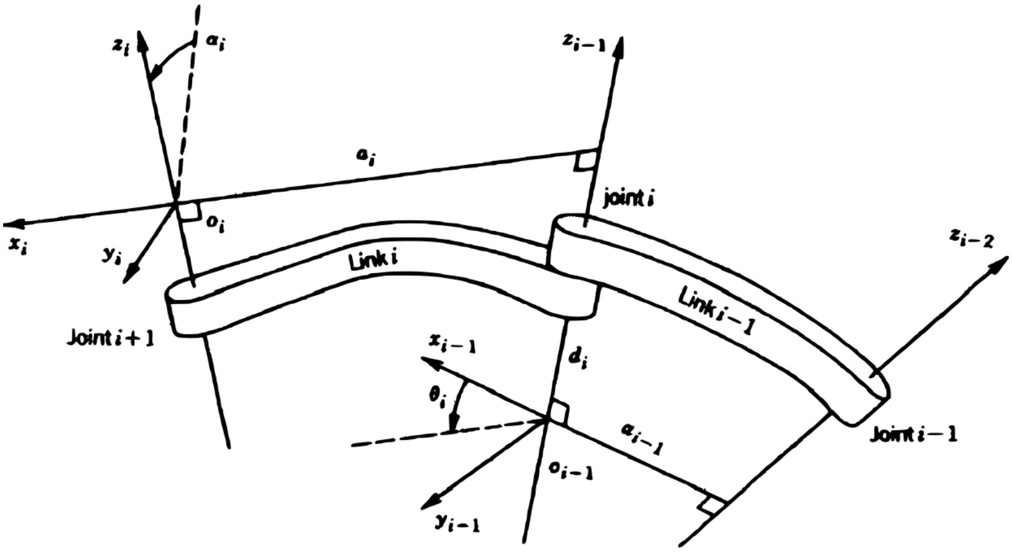
Denavit & Hartenberg Notation

For relating the base and the end effector, we need to both **rotate** and **translate**, so we'll use **homogeneus coordinates** (matrix is 4×4, encoding both rotation and translation):

$$ {}^{\mathrm{A}}_{\mathrm{EE}}A = \begin{bmatrix} {}^{\mathrm{A}}_{\mathrm{EE}}R & {}^{\mathrm{A}}_{\mathrm{EE}}P \\ 000 & 1 \end{bmatrix} $$

To create the homogeneous coordinates in a standardized way, we use the Denavit & Hartenberg (DH) notation which systematically relates the frames of two consecutive links. We have **4 parameters**, each of which relates frame i to frame $i-1$:

parameter	meaning
link length a_i	distance between z_i and z_{i-1} along x_i
link offset d_i	distance between x_i and x_{i-1} along z_i
link twist α_i	angle between z_i and z_{i-1} around x_i
joint angle θ_i	angle between x_i and x_{i-1} around z_i



Example: anthropomorphic arm (3 revolute joins):

	a_i	d_i	α_i	θ_i
1	0	$\pi/2$	0	θ_1^*
2	a_2	0	0	θ_2^*
3	a_3	0	0	θ_3^*

We then get the following transformations:

$$A_1^0(\theta_1) = \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad R = R_z(\theta_1) \cdot R_x(\pi/2)$$
$$A_i^{i-1}(\theta_1) = \begin{bmatrix} c_i & -s_i & 0 & a_i c_i \\ s_i & c_i & 0 & a_i s_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad i = 2, 3$$

To get the final transformations, we can multiply the matrices and get $T_3^0(\mathbf{q}) = A_1^0 A_2^1 A_3^2$

Inverse Kinematics

Definition (inverse kinematics) the process of finding a set of joint parameters $(q_1, \ldots, q_n)$ given the position/orientation of the **end-effector** $(r_1, \ldots, r_m)$.

$$(q_1, \ldots, q_n) = G(r_1, \ldots, r_m)$$

Definition (primary workspace) set WS_1 of all positions p that can be reached with *at least one* orientation R

Definition (secondary workspace) set WS_2 of all positions p that can be reached with *any* orientation R

Analytical solution (closed form)

- preferred (if it can be found)
- use geometric inspection, solve system of equations

Example: spherical wrist (3 revolute joins):

$$T_6^3(\mathbf{q}) = \begin{bmatrix} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - s_4c_6 & c_4s_5 & c_4s_5d_6 \\ s_4c_5c_6 + c_4s_6 & -s_4c_5s_6 + c_4c_6 & s_4s_5 & s_4s_5d_6 \\ -s_5c_6 & s_5s_6 & c_5 & c_5d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The matrix R is a ZYZ Euler rotation matrix and can be solved as such (see Body pose).

- we use indexes 3 to 6, because the wrist is usually at an end of another manipulator

Definition (decoupling) dividing inverse kinematics problem for two simpler problems, **inverse position kinematics** and **inverse orientation kinematics**

- is applicable for manipulators with *at least 6 joints* where the *last 3 intersect at a point*

The general approach is the following:

1. calculate the orientations and position where the wrist needs to be
2. calculate the orientations of the rest of the robot

Numerical solution (iterative form)

TBA

- Gradient method derivation (might be on the exam)

```
# will be typeset shortly
H(q) = 1/2 ||r_d f_r(q)||^2
dH / dq = 1/2  2 (-1) [r_d - f_r(q)]^T J_r(q)
dH / dq = (-1) [r_d - f_r(q)]^T J_r(q)  // nabla transposes
nabla q H(q) = (-1) [r_d - f_r(q)] J_r(q)^T  // nabla transposes
```