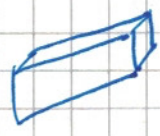
Chapter 2 : Configuration Space 4.11.21

- Degree of freedom (dof) of a rigid body
- dof of a mechanism
- freedoms & constraints of common robot joints
- Grubler's formula

Chapter 2.1: Dof of a Rigid Body

- Fundamental Question:
  - Where is it
- Answer: The <u>Configuration</u> is a specification of the position of all points of a Robot.



: Rigid Body (links)

: Joints connect links

- Typically only a few numbers are needed to describe the configuration on a Robot.

- C-Space (configuration Space) : Space of all configurations
- degrees of freedom (dof) : Dimension of the c-Space (the minimum # of <u>real</u> numbers needed to represent the configurati

real, continuous range

freedoms of bodies

$$\text{dof} = \sum(\text{freedoms of points}) - \text{number of independent constraints (equations)}$$

Degrees of freedom of rigid bodies:

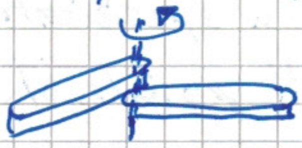
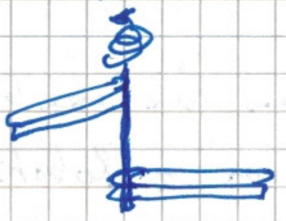
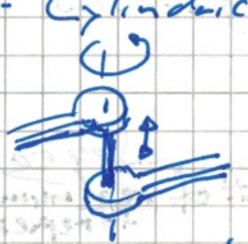
- 3D (spacial rigid body): 6 dof
- 2D (planar rigid body): 3 dof

Chapter 2.2 Dof of a Robot

21.11.21?

2.2.1 Robot Joints

Every Joint connects exactly <u>two</u> links.

Joints:	dof	constraints (c) 2D	 3D
R: Revolute, rotational, hinge 	1	2	5
P: Prismatic, translational, sliding 	1	2	5
H: Helical 	1	–	5
C: Cylindrical 	2	–	4
U: Universal 	2	–	4
S: Spherical 	3	–	3

29.11.21

Ch. 2.2.2. Grüblers Formula

N : number of bodies (links), including base

J : number of joints

m : dof of a single body { 6: spacial (3D) / 3: planar (2D)

c_i : joint constraint

f_i : number of freedoms of joint

$$dof = \underset{\text{rigid body freedoms}}{m(N-1)} - \underbrace{\sum_{i=1}^{J} c_i}_{\text{joint constraint}}$$

$$= m(N-1) - \sum_{i=1}^{J}(m-f_i)$$

Grüblers Formula
$$\underline{dof = m(N-1-J) + \sum_{i=1}^{J} f_i}$$

all constraints need to be independent

Grüblers formula provides a lower bound on the degrees of freedom of robots, i.e. if the mechanism has redundant constraints or singularities the dof can be higher than calculated by the formula.

11.12.21

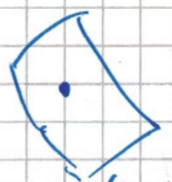
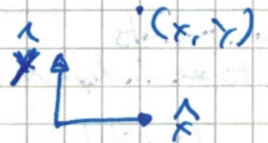
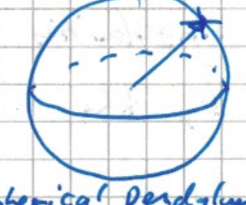
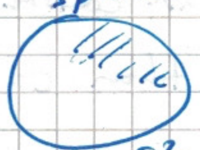
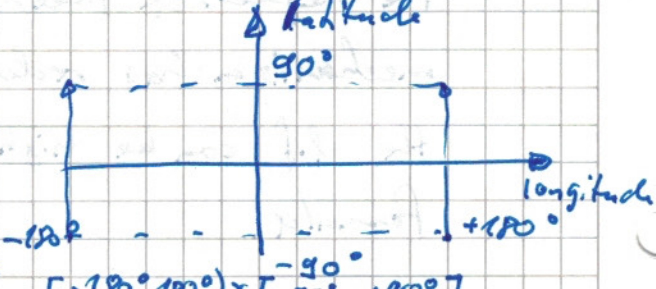
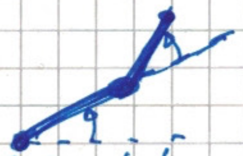
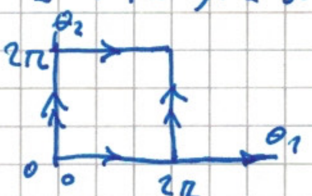
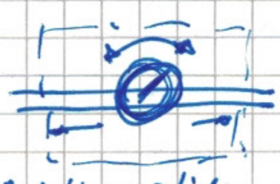
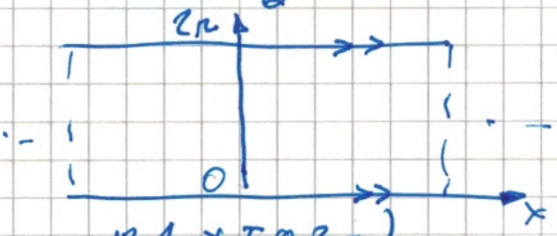
Chapter 2.3 Config Space Topology Representation

2.3.1 Config Space Topology ("shape" of c-space)

Two spaces are topologically equivalent if one can be smoothly deformed to the other without cutting and gluing.

The topology of a space is a fundamental property and is not affected on how to represent the space (i.e. with coordinates)

Examples:

System	Topology (of c-space)	sample representation of a config.
 Point on a plane	plane  $\mathbb{E}^2$ (2D euclidian space)	 $\mathbb{R}^2$ (2 real numbers
 spherical pendulum	sphere  S^2 (S^2 is the 2-D surface of a sphere in a 3-D space)	 $[-180°, 180°) \times [-90°, +90°]$
 2R robot arm	torus  $T^2 = S^1 \times S^1$	 $[0, 2\pi) \times [0, 2\pi)$
 Rotating sliding knob	Cylinder  $\mathbb{E}^1 \times S^1$	 $\mathbb{R}^1 \times [0, 2\pi)$

C-spaces of the same dimension can have different topologies!

$\mathbb{E}^n$: Euclidian (flat) space in n dimensions (usually represented as $\mathbb{R}^n$)
S^n: n-dimensional sphere
T^n: n-dimensional torus $\underbrace{S^1 \times S^1 \times \ldots \times S^1}_{n \text{ times}} = T^n \neq S^n$ (e.g. S^2: Sphere (in 3D) $\neq$ T^2: Torus)

Chapter 2.3.2 Configuration Space Representation

To represent a configuration space for calculations/computations with real numbers, some choices need to be made (i.e. vector (x, y) coordinates on a plane, with given origin)

The topology of the space <u>is independent</u> of the representation of the space!

Two possibilities to represent a space

explicit parametrization
- min. number of coordinates
  - n numbers for n dimensions
- e.g. latitude, longitude
- advantages: less complex
- disadvantages: singularities, ~~disc~~ discontinuities and other similar effects

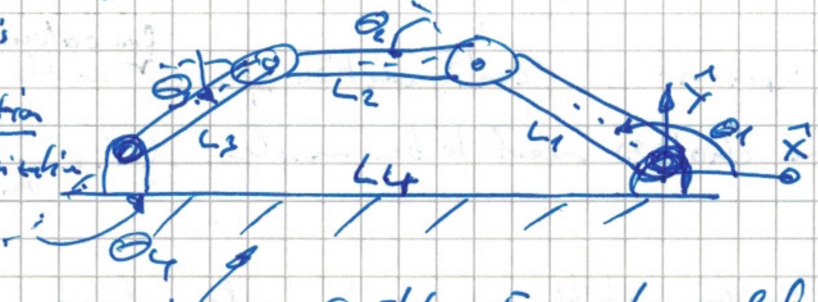
implicit representation
- a "surface embedded in higher-dimensional space
  - add constraints to reduce dof
- e.g. (x, y, z) such that
  $$x^2 + y^2 + z^2 = 1$$
  - 3 coordinates and 1 constraint
    → 2-D c-space (2 dof)
- disadvantages:
  - more complex
  - Velocities are <u>not</u> time-~~deri~~ rate of change of coordinates
- advantage: no singularities, discontinuities...

↳ Uses in this course/book (in particular: Rotation Matrix)

Rotation Matrix:
- 9 numbers
- 6 constraints
- ↳ 3 orientation freedoms of rigid body in space

Chapter 2.4 Configuration and Velocity Constraints

For closed loops it is often easier to find an <u>implicit representation</u> than an explicit parametrization



Four-bar: Grüblers formula → dof = 1

We can see the C-Space as 1D space embedded in the 4D space of <u>joint angles</u> defined by the <u>loop-closure equations</u>:

the final position and orientation must be equal to the initial position and orientation, when going around the loop.

$$\begin{cases} L_1 \cos\theta_1 + L_2 \cos(\theta_1+\theta_2) + \dots + L_4 \cos(\theta_1+\dots+\theta_4) = 0 \\ L_1 \sin\theta_1 + L_2 \sin(\theta_1+\theta_2) + \dots + L_4 \sin(\theta_1+\dots+\theta_4) = 0 \\ \theta_1+\theta_2+\theta_3+\theta_4-2\pi = 0 \end{cases}$$

The C-space can be viewed as a surface of dimension n-k embedded in $\mathbb{R}^n$

Loop-closure equation in vector form:

θ: configurations $= [\theta_1, \dots, \theta_n]^T \in \mathbb{R}^n$

$$g(\theta) = \begin{bmatrix} g_1(\theta_1, \dots, \theta_n) \\ \vdots \\ g_k(\theta_1, \dots, \theta_n) \end{bmatrix} = 0$$

with k <u>holonomic constraints</u> (reduce the dimension of the C-space)

$k \leq n$

$$\boxed{\begin{array}{l} \text{if } \theta \in \mathbb{R}^n \\ \text{and } g(\theta) \in \mathbb{R}^k \\ \text{then dof} = n-k \end{array}}$$

⇒ n variables needed to define robots configuration

⇒ k independent holonomic constraints

⇒ dimension of C-space (= dof)

Velocity constraints: as $g(\theta) = 0$ the time rate of change of g must also be zero

$$\hookrightarrow \begin{bmatrix} \frac{\partial g_1}{\partial \theta_1}(\theta)\dot{\theta}_1 + \dots + \frac{\partial g_1}{\partial \theta_n}(\theta)\dot{\theta}_n \\ \vdots \\ \frac{\partial g_k}{\partial \theta_1}(\theta)\dot{\theta}_1 + \dots + \frac{\partial g_k}{\partial \theta_n}(\theta)\dot{\theta}_n \end{bmatrix} = 0$$

holonomic constraints: "integrable" constraints as they are the integral of velocity constraints.

$A(\theta) = \frac{\partial g(\theta)}{\partial \theta}$

$$\hookrightarrow \underbrace{\begin{bmatrix} \frac{\partial g_1}{\partial \theta_1}(\theta) + \dots + \frac{\partial g_1}{\partial \theta_n}(\theta) \\ \vdots \\ \frac{\partial g_k}{\partial \theta_1}(\theta) + \dots + \frac{\partial g_k}{\partial \theta_n}(\theta) \end{bmatrix}}_{A(\theta)} \underbrace{\begin{bmatrix} \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_n \end{bmatrix}}_{\dot{\theta}} = 0$$

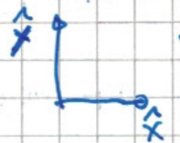
$A(\theta)$ → k rows, n columns: $A_{k \times n}$

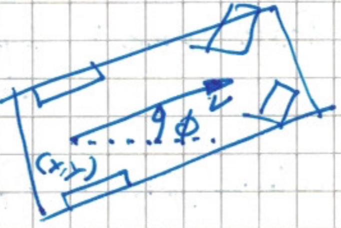
$$\boxed{\text{Pfaffian constraints: } A(\theta)\dot{\theta} = 0}$$

In some cases a set of ~~v~~ (Pfaffian) velocity constraints cannot be integrated to equivalent configuration constraints.

↳ <u>nonholonomic constraints</u>

car:





nonholonomic constraints are not integrable

configuration of car: $q = (\phi, x, y)$

velocities:

$$\left.\begin{array}{l}\dot{x} = v\cos\phi \\ \dot{y} = v\sin\phi\end{array}\right\} \dot{x}\sin\phi - \dot{y}\cos\phi = 0$$

$$A(q)\dot{q} = A(q)\begin{bmatrix}\dot{\phi}\\ \dot{x}\\ \dot{y}\end{bmatrix} = 0 \quad \text{as Pfaffian constraint}$$

$$A(q) = [\,0 \quad \sin\phi \quad -\cos\phi\,] \in \mathbb{R}^{1\times 3}$$

A nonholonomic constraint reduces the space of possible velocities of the car (car can not slide sidewards)

It does not reduce the space of configurations!

Nonholonomic constraints arise in robot systems subject to conservation of momentum or rolling without slipping.

Summary:

Holonomic constraints: Constraints on configuration

Nonholonomic constraints: Constraints on velocity

Pfaffian constraints: $A(\theta)\dot{\theta} = 0$

↳ can be holonomic or nonholonomic

2.5 Task space and Workspace

C-space : space of all possible configs of a robot

Task space : Space where the robots task is naturally expressed
Task space is defined by task not by the robot

Work space : Specification of readable configs of a robots end-effector. Independent of a particular task. Defined by the robot.