RMP-L1-Configuration Space

Key words	DOF, angle-axis representation, configuration space, euler angles, holonomic & non-holonomic, hull, rotation matrix, workspace
Status	note complete

Workspace:

- all **reachable points** of the end-effector.
- robot modelling: as joints, links.
- ▼ collision check of a path from A to B:
    - discretize the path into multiple small sub-paths: check collision of sub-paths
    - large obstacles: larger steps
    - small obstacles in confined space: smaller steps

    —> testing each step necessary, each test **computationally expensive**: for each sub-path, test all parts of robot structure for all obstacles.

    —> other representations of robot

    —> **transform workspace into configuration space**: easier test, robot represented as a single point.

Robot modelling

- ▼ **Robot modelling**
    - ▼ rigid body
        - ▼ types
            - one single rigid body: represented as **one point**

- ▼ several linked rigid bodies as **links**, each rigid body is connected with **joints**
    - **each rigid body** can be represented as **one point:** motion of one point represents this rigid body
    - motion parametrization: point representation — rotation as θ, translation as (x, y, z)

- ▼ **Degree of Freedom: minimum** number of parameters needed to fully describe a motion.
  - open chains / serial mechanism: DOF increase by 1 when adding 1 joint.
    - common joint: **revolute** — 1 DOF
  - closed chains / parallel mechanism: $DOF = N(k-1) - \Sigma_{i=1}^{n}(N - f_i)$
    - N: each movable link has N DOF — planar: N = 3, spatial: N = 6
    - k: number of **links** (including stationary ground link)
    - k-1: number of **movable links**
    - n: number of **joints**
    - f_i: DOF of each joint — revolute: 1,

Configuration Space

- ▼ **configuration space C**:
  - motivation: **abstraction** of motion. describe motion by the **cause**.
    - workspace: (x,y) position of the end effector
    - config-space: rotation of the links (θ_1, θ_2)
  - **all possible configurations** of the robot motions, **robot** represented **as a point**

- dimension: **minimum** number of **independent variables** in the representation of
  the configuration — **degrees of freedom (DOF)**
  - redundant dimension: not independent. (eg: represent screw with rotation θ and translation x — a line exists in config-space)
- topology: not Cartesian space, a toral space
- challenge: **projection of the obstacles** onto config-space. The shape of the obstacles depends on
  - shape of the obstacles
  - structure of robot

  —> no direct transformation

▼ examples: drawing of workspaces VS. config-spaces

Workspace can be parametrized differently, but Configuration space defines only the minimum number of parameters that fully describe the motion, independent of parametrization.

▼ rigid robot in 2D-workspace: #parameters aren't necessarily minimum(DOF)

- 3 parameters: (x,y) with orientation θ
- 4 parameters: (x,y) with (u= $cos\theta$,v= $sin\theta$)
- **3D config-space**: $q = (x, y, \theta)$, **independent of parametrization (DOF = 3)**.

▼ rigid robot in 3D-workspace

- 12 parameters: translation T = (x,y,z) with **rotation matrix** $R_{3\times3}$
- 6 parameters: translation T = (x,y,z) with **euler angles** (α, β, γ) — rotation around the x,y,z-axis
  - Problem with Euler angles: **gimbal lock**, lost of 1 DOF when eg: $\beta = 90^\circ$, changing α or γ has same effect.

    —> different rotation orders has different representations

- 7 parameters: translation T = (x,y,z) with **Axis-angle representation** (k_1, k_2, k_3, θ) for rotation: roation matrix R is represented by **Rodrigues formula**

  $R = I + (1 - cos\theta) \cdot K^2 + sin\theta \cdot K$

  - K: skew-symmetric matrix, rotation axis
  - θ: rotation angle
  - no gimbal lock, able to describe every motion.

- 7 parameters: translation T =(x,y,z) with orientation as **unit quadranions** (u_1, u_2, u_3, u_4)
  - compact, no sigularity(gimbal lock), reflect topology of orientation space.

- **6D config-space**: $q = (x, y, z, \alpha, \beta, \gamma)$ — position, orientation

▼ Paths

- a **continuous curve** connecting start configuration q and end configuration q'

  —> represents the **change of geometric structure to get to goal**.

  $\tau : s \in [0, 1] \rightarrow \tau(s) \in C,\ \ \tau[0] = q, \tau[1] = q'$

- **trajectory:** a path parametrized by **time**

  $\tau : t \in [0, T] \rightarrow \tau(t) \in C,\ \ \tau[0] = q, \tau[T] = q'$

- constraints: max/min. length, bounded curvature, smoothness, min/max. time

- **free path:** lies **entirely** in the **free space F.** Robot **doesn't slide along obstacle boundaries.**
  - free space F is open subset of C

- **semi-free path:** Robot **can slide along obstacle boundaries**
  - semi-free space is closed subset of C

▼ Obstacles

- **collsion-free, free configuration q:** robot doesn't collide with any obstacles.
- configuration space obstacles: **C-obstacles**
  - shape: combination of **obstacle shape** and **robot shape**
- ▼ construction of free space and C-obstales:
  - ▼ **collision test** (brute-force)

    Process:

    1. **discretization:** divide the space into grids. eg: 0.1° accuracy for $\theta_1 \in [0, 180^\circ], \theta_2 \in [0, 360^\circ]$ —> 1800 x 3600 space
    2. test for collision against each obstacles for each grid.
    3. if collision: grid = 1; if no collision: grid =0

       —> C-obstacles: grids with collision.

    —> number of test: 1800 x 3600 x n, (n: #obstacles), **computation expensive, offline test necessary.**
  - ▼ **robot padding:** influence of the robot shape on config-space
    - for **circular** or **regular shaped** robots
    - Process:
      1. schrink the robot into a point (cicular: radius r, rectangular: width and length)
      2. pad the shape onto the obstacles

      —> **faster and easier** than brute-force collision test
  - ▼ **collision test with hulls**:
    - Motivation:
      - accelerate brute-force collision test
      - most of the space is free
    - Process:
      1. surround the robot with a hull: sphere, rectangular, triangle, etc.

2. if the hull collides with obstacles: further testing

   if no collision: free space

   - by defining the size of hull: control the complexity of the test.

- Hulls:
  - sphere: r
  - rectangular: $x_{min}, x_{max}, y_{min}, y_{max}$
  - rectangular with PCA: find greatest variation and orthogal direction
  - triangle with PCA

holonomic & non-holonomic systems:

- holonomic: **all** parameters of DOF are **independent. —>** no koppling between motion parameters.
- non-holonomic: **not all** parameters are **independent.** —> additional constraints.
  - eg: car — a car can't move horizontally without changing orientation.