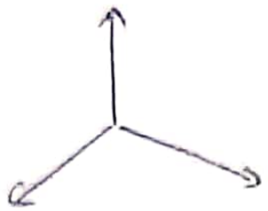
examples

- C → configuration space
- n → dimension of C

- a point robot in $\mathbb{R}^3$



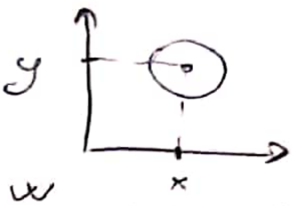
(figure 1)

$$q = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$n = 3$

$C = \mathbb{R}^3$

- a disk robot in $\mathbb{R}^2$

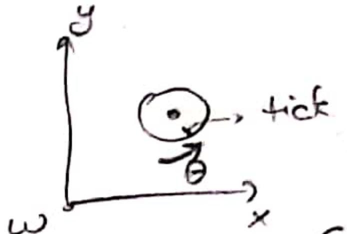


(figure 2)

we choose representative position of the center and determine its x and y coordinates

$q = \begin{bmatrix} x \\ y \end{bmatrix}$ $n = 2$ $C = \mathbb{R}^2$

- a disk robot in $\mathbb{R}^2$ with a tick



(figure 3)

θ represents the orientation of the tick. x and y gives us the centre of disk robot.

Note: without θ, we can't describe where the tick is.

$q = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$ $n = 3.$

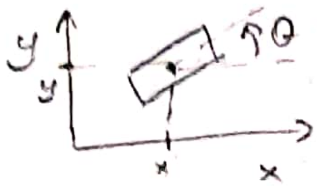
$$C = \mathbb{R}^2 \times SO(2) \quad (1)$$

$\mathbb{R}^2$ → involves x, y coordinates

$SO(2)$ → involves the information about the θ.

To sum up, configuration space is cartesian product of C and SO(2)

- a polygonal robot in $\mathbb{R}^2$



θ → we get θ as an orientation wrt x axis.

$$q = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \qquad C = \underbrace{\mathbb{R}^2 \times SO(2)}_{SE(2)} \;\;\S$$

(figure 4) $\quad n = 3$.

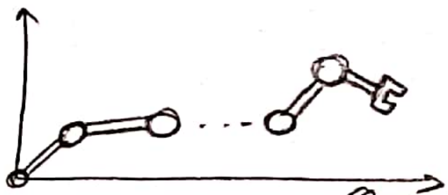
$SE(2) \rightarrow$ Special Euclidean in 2D space

- a polyhedral robot in $\mathbb{R}^3$

$$q = \begin{pmatrix} x \\ y \\ z \\ \alpha \\ \beta \\ \gamma \end{pmatrix} \quad n = 6 \qquad C = \underbrace{\mathbb{R}^3 \times SO(3)}_{SE(3)}$$

Note : Examples till now were single body robots. After that we'll analyze multiple-body robots.

- a planar manipulator with n_j revolute joints (an nR robot)



(figure 5)

Representation in space :
by taking
→ coordinates of each body ⇒ $3 \cdot n_j$
→ orientation of each body

Is this a minimal set? → No!

How can we manage to have minimal set?

→ each joint introduces 2 constraints:

~~End~~ For instance the end of the first link is the beginning of the second link in terms of cartesian coordinates.

So we can say that our set will have

$3n_j - 2n_j = n_j$ (3)

$$q = \begin{pmatrix} q_1 \\ \vdots \\ q_{n_j} \end{pmatrix} \qquad n_j \to \text{dimension of conf. space}$$

Configuration space will be:

$$C = \underbrace{SO(2) \times SO(2) \times \dots \times SO(2)}_{n_j \text{ times}} = (SO(2))^{n_j} \quad (4)$$

Because each joint in figure (5) is represented by ~~by~~ planar rotation (SO_2).

Note: $SO(2) \times SO(2) \times \text{———} \times SO(2) \neq SO(3)$.

• a spatial manipulator with n_j revolute joints :
will be completed number of coordinates with the equation (5) to represent our robot .

We get (5) by taking :

→ center of each body
→ orientation of each joint $\Big\} \Rightarrow 6n_j$

$6n_j$ is not minimal set of coordinates we need to represent this robot .

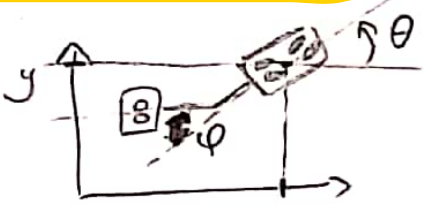
$$6n_j - 5n_j = n_j \quad (5)$$

Configuration space :

$$C = (SO(2))^{n_j} \quad (6)$$

(6) ⇒ Even robot is in 3D-space , orientations are still planar rotations → $SO(2)$.

• a car like robot with a trailer in $\mathbb{R}^2$



(figure 6)

θ → orientation of car wrt its x.
φ → orientation of trailer wrt car
x, y → coordinates of centre of car .

# of coordinates we need describe it :

$6 - 2 = 4$ (7)

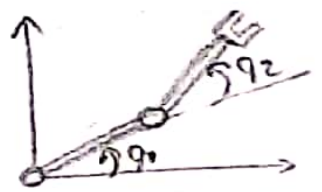
$$q = \begin{pmatrix} x \\ y \\ \theta \\ \varphi \end{pmatrix} \qquad C = \mathbb{R}^2 \times \underset{\theta}{SO(2)} \times \underset{\varphi}{SO(2)} .$$

Note: The steering angles of wheels (figure 6) doesn't affect the current position of the car. That's why we don't take them into consideration at this step. However, they will affect the future motion but not the current position we are interested in.

Topology of C

What kind of space is C ?

- examples
  - a 2R manipulator in R_2



(figure 7)

$q = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix}$

$C = SO(2) \times SO(2)$.

If we draw C :

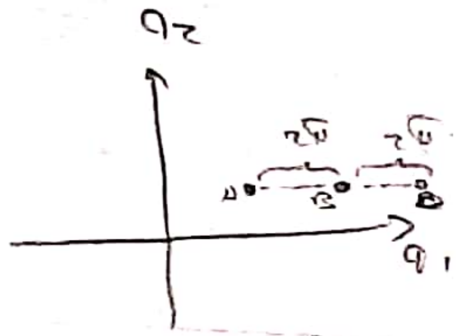


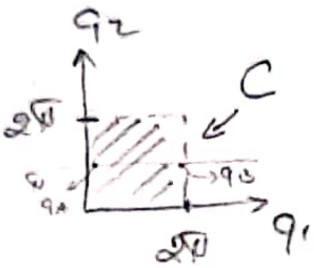
figure 8.

A, B, D points in figure(8) correspond to the same posture of the robot.

This is the problem so called as lost of injectivity.

How can we solve the injectivity lost?

Scanned with CamScanner



(figure 9)

→ It solves loss of injectivity however it is not good representation.

If we represent q_A and q_B postures (figure 9) with figure (10):

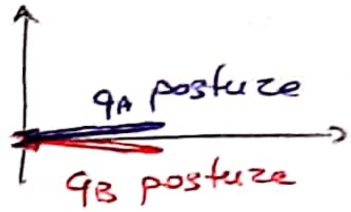


figure 10

They (q_A and q_B) are really close. Additionally, if we push q_A → to left (a bit) q_B → to right, they will collide.

In the representation of (figure 9) solved injectivity but not topology.

Then we will solve it by transforming the square to torus:

figure 10 (q_A • q_B •)

It needs to be embedded in R^3 for visualization

We solved also topology problem, too.

But, it is not a Euclidean space.

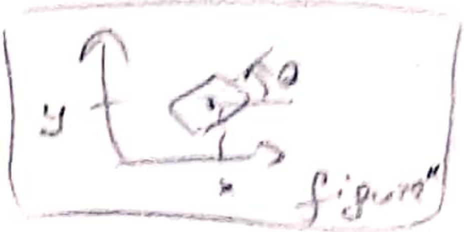
It is a manifold → dimension n: A space where each neighbourhood of a point is homeomorphic (it looks like) a neighbourhood of $\mathbb{R}^n$.

AMP
08.10.2020

Note: whenever configuration space involves orientation ($SO(2)$/$SO(3)$) it is not Euclidean but MANIFOLD.

For instance: Polygon robot in $\mathbb{R}^2$.

$q = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}$ $\quad C = \mathbb{R}^2 \times SO(2)$



We can show it as figure 12:

But it is not correct. To sum up, in this case embedding does not help ~~us~~ visualization.

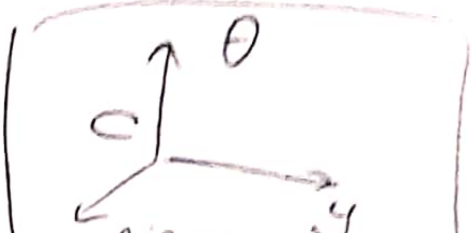


fig.12

distance

need: $d(q_A, q_B) \geq 0$

$d(q_A, q_B) = 0 \quad \text{iff} \quad q_A = q_B$

$d(q_A, q_B) = d(q_B, q_A)$

$d(q_A, q_B) + d(q_B, q_C) \leq d(q_A, q_C)$

problem:

C is not a Euclidean space $\Rightarrow$ cannot use Euclidean space.

How do we compute distances on manifolds? → Riemannian geometry.

→ We will use geodesics (path of shortest length between 2 points).

It is used only for simple manipolds. We use another method in Robotics.

in Robotics

$B(q) \to$ region of W occupied by the robot when the configuration is q.

$P(q) \to$ position of point ~~p~~ p (of the robot) in W when the configuration is q.

$p \to$ position of 2nd joint

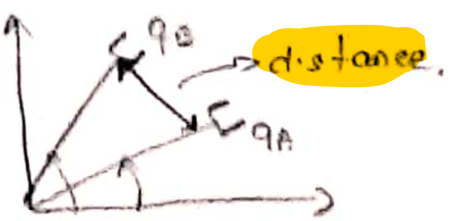
figure 13.

$$d(q_A, q_B) = \max_{p \in B} \underbrace{\| p(q_A) - p(q_B) \|}_{\text{Euclidean distance}} \quad (6)$$

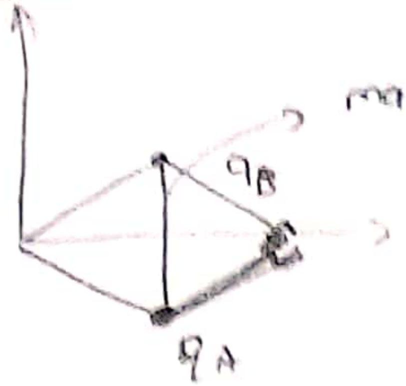
(6) => Displacement Metric (distance) (between 2 configurations)

Basic idea: Look for maximum Euclidean distance in 2 (or multiple) configurations

example:

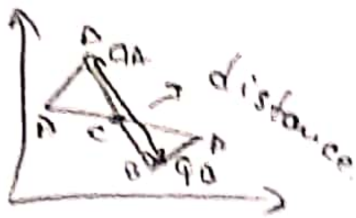


example



max distance between 2 configurations

example



we will check distances between A, B, C points in both configuration. In this configuration, the max distance is between B in q_A and B in q_B. It is our metric.

Note: In some points we will neglect that our robot is in Manifold and we'll cheat as if is in Euclidean space.