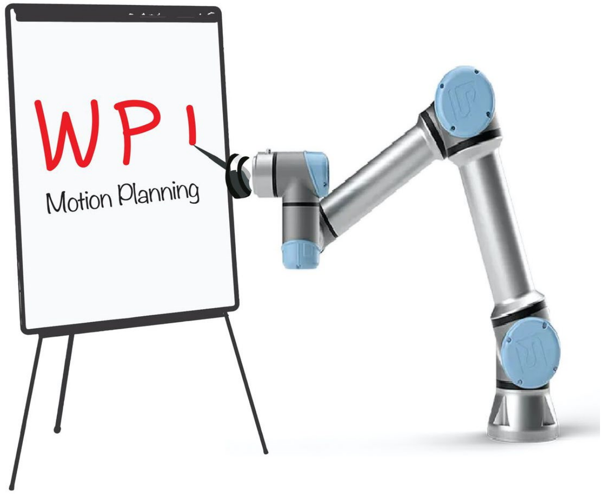
RBE550

Motion Planning

Configuration Space

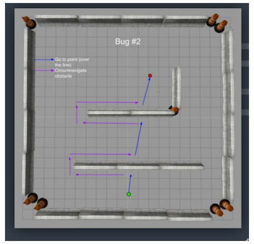


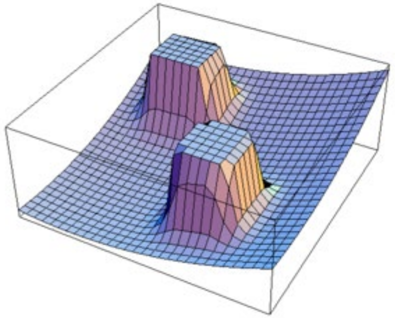
Constantinos Chamzas
www.cchamzas.com
www.elpislab.org

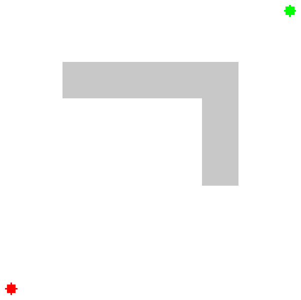
Disclaimer and Acknowledgments

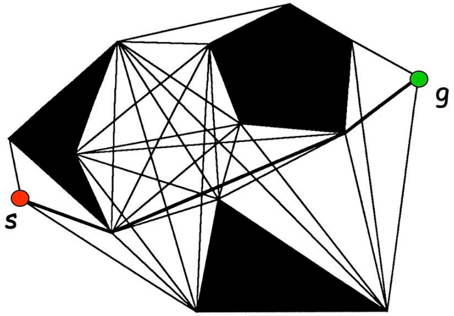
The slides are a compilation of work based on notes and slides from Constantinos Chamzas, Lydia Kavraki, Zak Kingston, Howie Choset, David Hsu, Greg Hager, Mark Moll, G. Ayorkor Mills-Tetty, Hyungpil Moon, Zack Dodds, Nancy Amato, Steven Lavalle, Seth Hutchinson, George Kantor, Dieter Fox, Vincent Lee-Shue Jr., Prasad Narendra Atkar, Kevin Tantiseviand, Bernice Ma, David Conner, Morteza Lahijanian, Erion Plaku, and students taking comp450/comp550 at Rice University.

Methods for Point Robots

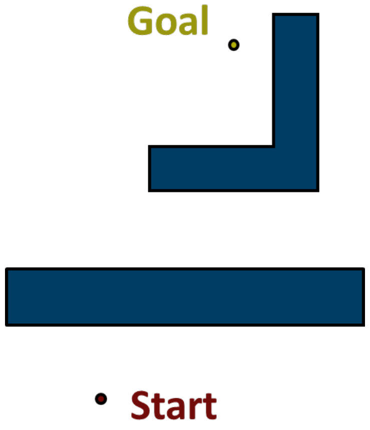








Bug Algorithms Potential Fields A* on Grids Roadmaps



Point Robot

How?

Real Robot

Today's Overview

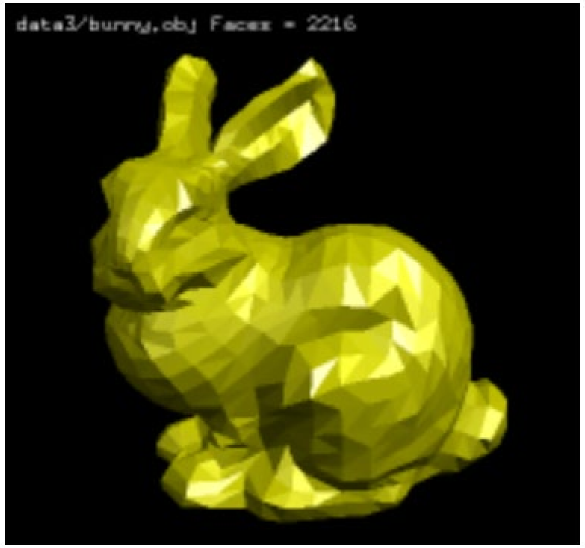
1. Represent Robot:
    I. Rigid body Representations (What shape are things)
    II. Rigid body transformations (Where objects are)
    III. Forward/Inverse Kinematics (Where sets of things are)
2. Configuration Space
    1. C-Space Idea
    2. Topology of C-space
    3. C-space Obstacles
    4. C-Space Paths

Some Assumptions

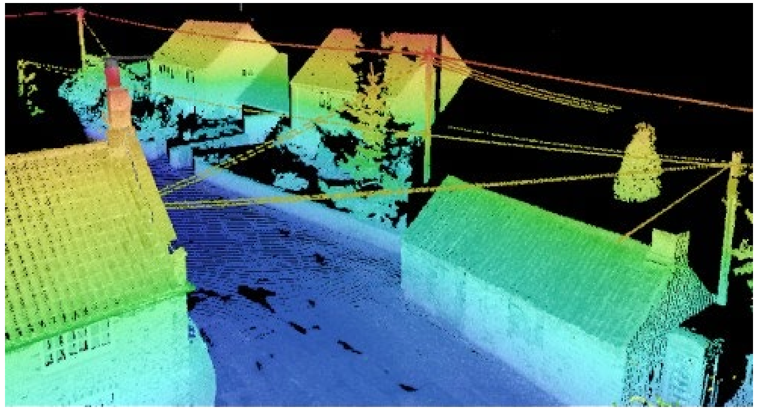
Robots are a collection of known rigid bodies that execute paths exactly

1. Geometry and position are known for all rigid bodies
    - Not knowing the exact geometry or position can introduce uncertainty in the problem (later in class)
2. Geometry is *rigid*
    - We assume objects do not deform – extensions assume "soft" bodies, elastics, deformable objects, etc.
3. Paths can be executed by robots *exactly*
    - Assume that robot has a good position controller that can execute geometric paths (the *quasistatic assumption), e.g.,* big factory robot with big motors can track positions exactly. Not true for many systems.

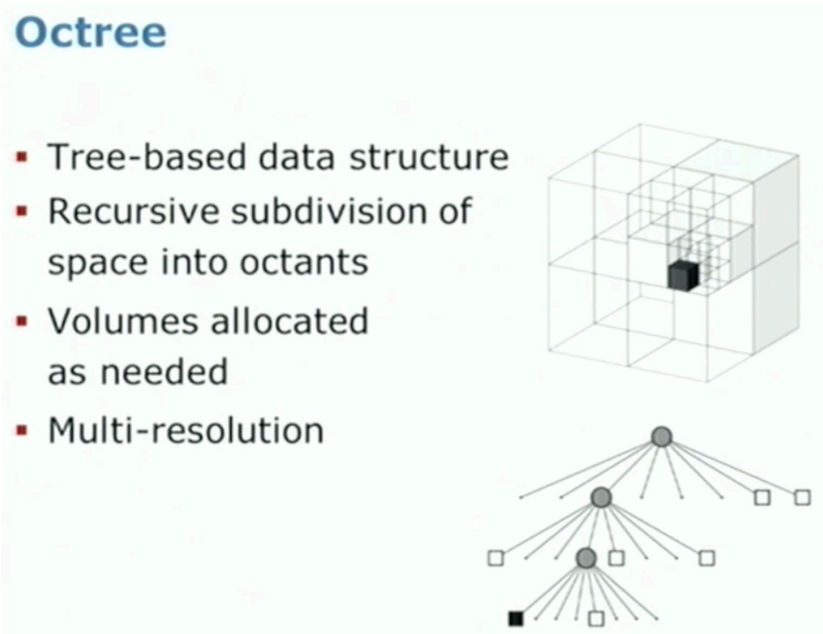
Representing Rigid Bodies in the Workspace



Meshes

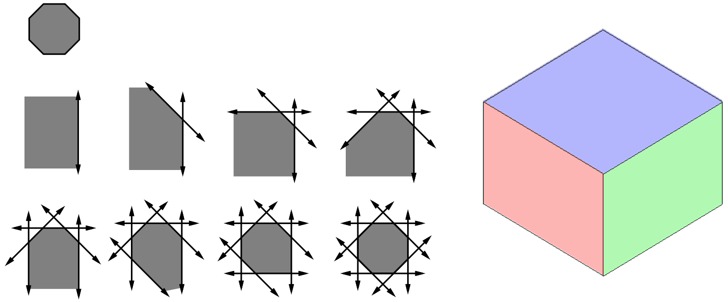


PointClouds



$$H = \{\, \boldsymbol{x} \in \mathbb{R}^n \mid f(x) \le 0 \,\}$$

Algebraic Primitives



Primitive Shapes

<u>All rigid objects exist in a world W which is either 2D or 3D</u>

$$W = \mathbb{R}^2 \; or \; \mathbb{R}^3$$

Representing Geometry of Physical Objects

Two classes of rigid objects in the world:

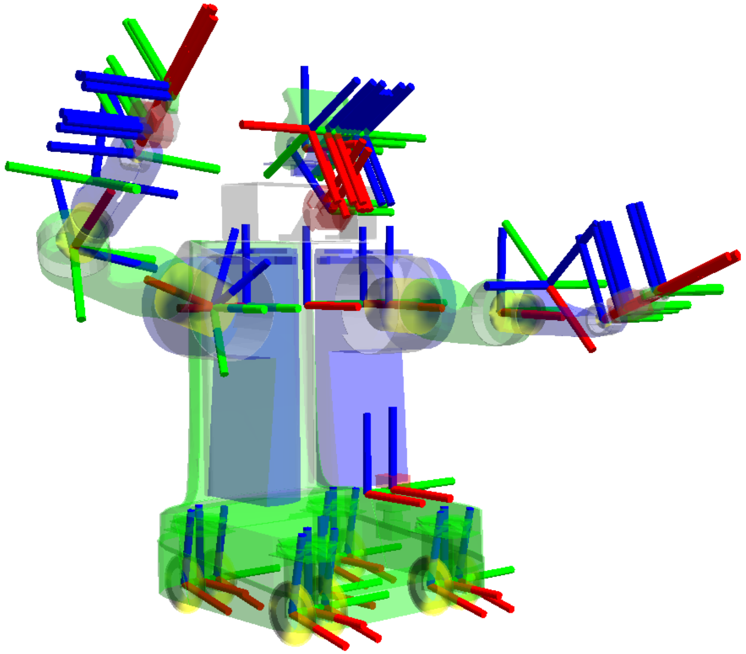
- The Robot is a set of rigid bodies that are controllable
  - e.g., a mobile robot base, a free-flying drone, the collection of links in an arm

- The Obstacles are a set of fixed rigid bodies that the robot cannot intersect (collide) with
  - e.g., the walls, furniture, cups and glasses on a table

Rigid Body Transformations (Where objects are)

Rigid Body Transformations are key to robotics – tell us where rigid bodies are in the world

Based on matrix and vector math, linear algebra



Rigid Body Motions in 2-D

Combining translation and rotation

- Given a vector v representing a point on rigid body A, we can describe the transformation of A to A' by a rotation θ followed by translation $t = (x_t, y_t)$ on v

$$v = (x, y) \longrightarrow v' = R(\theta)v + t = \begin{pmatrix} x\ \cos\theta - y\sin\theta + x_t \\ x\ \sin\theta + y\cos\theta + y_t \end{pmatrix}$$

- This transformation is given by *homogeneous transformation matrix*:

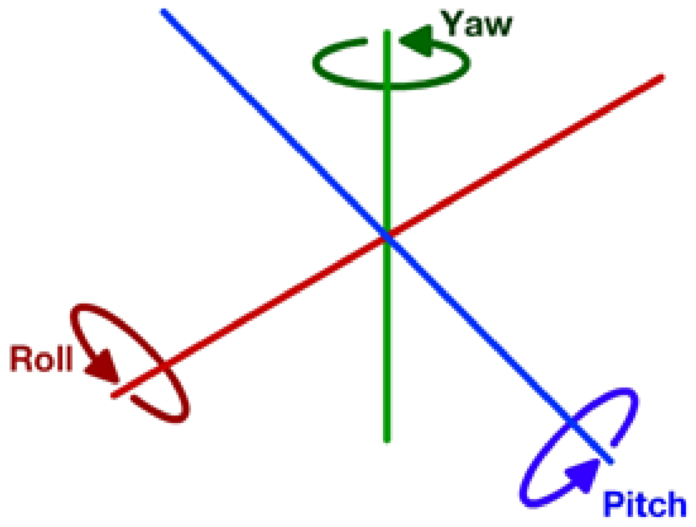
$$T = \begin{pmatrix} R(\theta) & t \\ 0 & 1 \end{pmatrix}$$

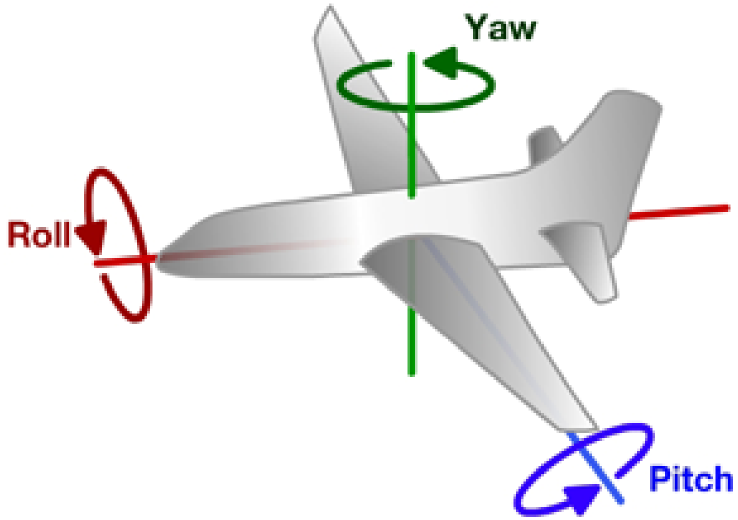
- i.e.,

$$\begin{pmatrix} v' \\ 1 \end{pmatrix} = T\begin{pmatrix} v \\ 1 \end{pmatrix} = \begin{pmatrix} R(\theta) & t \\ 0 & 1 \end{pmatrix}\begin{pmatrix} v \\ 1 \end{pmatrix} = \begin{pmatrix} R(\theta)v + t \\ 1 \end{pmatrix}$$

Rotations in 3-D

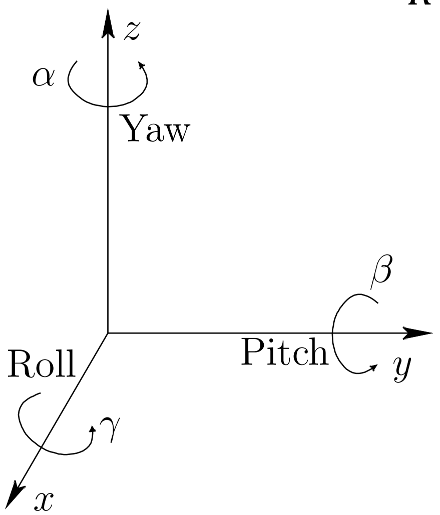
- Yaw, pitch, and roll in an aircraft





Successive Rotations in 3-D

- Yaw, pitch, and roll rotations can be used to place a 3D body in any orientation
- A single rotation matrix can be found by multiplying yaw, pitch, and roll rotation matrices:
  - e.g., Rotate about x-axis by γ $\rightarrow$ Rotate about y-axis by β $\rightarrow$ Rotate about z-axis by α



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

$$= \begin{pmatrix} \cos\alpha\cos\beta & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma \\ \sin\alpha\cos\beta & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma \\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\cos\gamma \end{pmatrix}$$

Remember Rotations are not commutative!

Properties of Rotation Matrix

- Rotation matrix R is an orthogonal matrix, i.e.,
  - i.e., the columns are mutually orthonormal

$$r_i^T r_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

$\mathrm{r_i}, r_j$ are columns of R

- Hence,

$$R^T R = R R^T = I \quad \rightarrow \quad \det(R) = \pm 1$$

- In right-handed coordinate system:

$$\det(R) = +1$$

- Special Orthogonal matrices:

$$SO(n) = \{ R \in \mathbb{R}^{n \times n} \mid R R^T = I \ \wedge \ \det(R) = 1 \}$$

Rigid Body Motions in 3D

- Given a vector v representing a point on rigid body A

$$v = (x, y, z) \longrightarrow v' = R(\alpha, \beta, \gamma)v + t$$

- Same transformation by homogeneous transformation matrix:

$$T = \begin{pmatrix} R(\alpha, \beta, \gamma) & t \\ 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} v' \\ 1 \end{pmatrix} = T \begin{pmatrix} v \\ 1 \end{pmatrix}$$

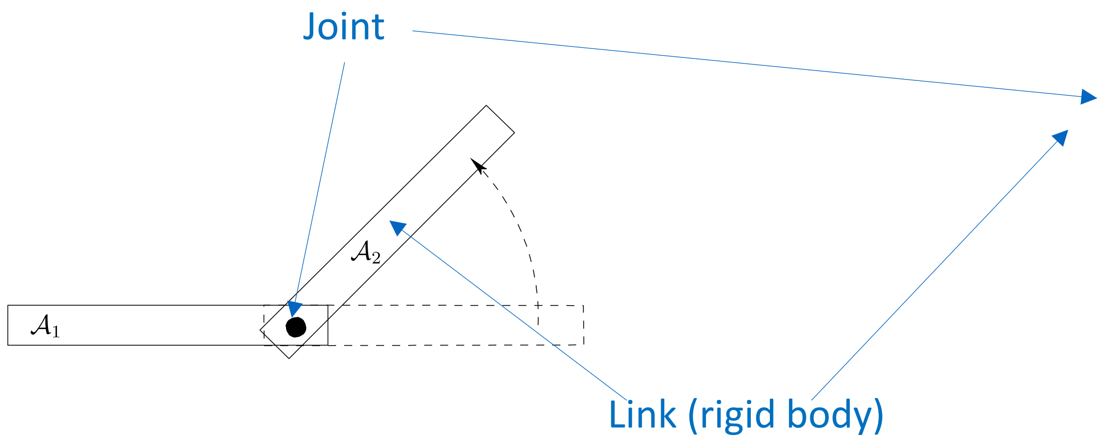
- Successive transformations can be easily computed in homogeneous coordinates (yaw, pitch, roll):

Rotate by $(\gamma_1 \to \beta_1 \to \alpha_1)$ $\to$ Translate by t_1 $\to$ Rotate by $(\gamma_2 \to \beta_2 \to \alpha_2)$ $\to$ Translate by t_2

$$\begin{pmatrix} R(\alpha_2, \beta_2, \gamma_2) & t_2 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R(\alpha_1, \beta_1, \gamma_1) & t_1 \\ 0 & 1 \end{pmatrix}$$

Forward/Inverse Kinematics (Where sets of things are)

- **Kinematics:** study of possible movements and configurations of a system - "geometry of the system"
  - **Link:** Each rigid body in a chain of rigid bodies
  - **Joint:** Connects two rigid bodies and enforces constraints
- **Forward kinematics:** Position of the end effector in terms of joint angles
- **Inverse kinematics:** Joint angles in terms of the position of the end effector



Forward Kinematics for 2D Chains

- Pose of A_1 relative to the reference (global) frame

$$T_1 = \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Pose of A_2 relative to A_1's frame

$$T_2 = \begin{pmatrix} \cos\theta_2 & -\sin\theta_2 & a_1 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

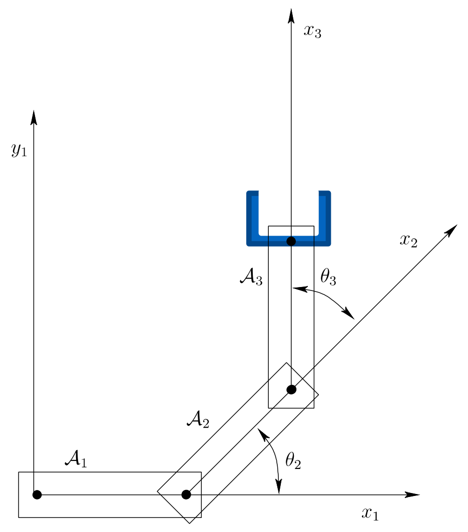
- Pose of A_3 relative to A_2's frame

$$T_3 = \begin{pmatrix} \cos\theta_3 & -\sin\theta_3 & a_2 \\ \sin\theta_3 & \cos\theta_3 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Pose of end-effector relative to A_3's frame

$$T_4 = \begin{pmatrix} 1 & 0 & a_3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- End-effector in global frame: $\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = T_1T_2T_3T_4 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$



$\theta_1 = 0$

$$T_1T_2T_3T_4 = \begin{pmatrix} c_{123} & -s_{123} & a_1c_1 + a_2c_{12} + a_3c_{123} \\ s_{123} & c_{123} & a_1s_1 + a_2s_{12} + a_3s_{123} \\ 0 & 0 & 1 \end{pmatrix}$$

$$x = a_1\cos\theta_1 + a_2\cos(\theta_1+\theta_2) + a_3\cos(\theta_1+\theta_2+\theta_3)$$

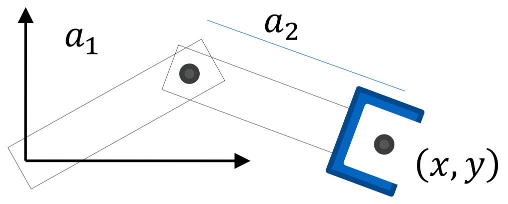
$$y = a_1\sin\theta_1 + a_2\sin(\theta_1+\theta_2) + a_3\sin(\theta_1+\theta_2+\theta_3)$$

Analytic Inverse Kinematics in 2-D

- Consider the position after two links:

(1) $$x = a_1 \cos\theta_1 + a_2 \cos(\theta_1 + \theta_2)$$

(2) $$y = a_1 \sin\theta_1 + a_2 \sin(\theta_1 + \theta_2)$$



- Give x and y, find θ_1 and θ_2

  - Recall:

    (3) $$\sin^2\theta_1 + \cos^2\theta_1 = 1$$

    (4) $$\sin^2\theta_2 + \cos^2\theta_2 = 1$$

  - 4 equations, 4 unknown. Solve for

$$\sin\theta_1, \cos\theta_1, \sin\theta_2, \cos\theta_2$$

  - Also:

$$\sin(\theta_1 + \theta_2) = \sin\theta_1 \cos\theta_2 + \cos\theta_1 \sin\theta_2$$

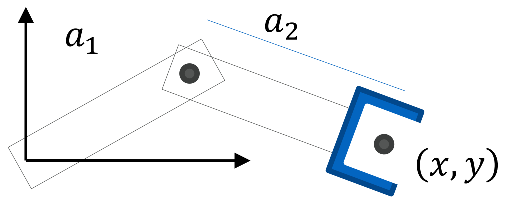
$$\cos(\theta_1 + \theta_2) = \cos\theta_1 \cos\theta_2 - \sin\theta_1 \sin\theta_2$$

Analytic Inverse Kinematics in 2-D

- Consider the position after two links:

(1) $x = a_1 \cos\theta_1 + a_2 \cos(\theta_1 + \theta_2)$

(2) $y = a_1 \sin\theta_1 + a_2 \sin(\theta_1 + \theta_2)$



- Give x and y, find θ_1 and θ_2
  - $(1)^2 + (2)^2$:

$$x^2 + y^2 = a_1^2 + a_2^2 + 2a_1a_2\ \cos^2\theta_2$$

$$\boxed{\cos\theta_2 = \frac{1}{2a_1a_2}\left((x^2+y^2) - (a_1^2 + a_2^2)\right)}$$

  - $(4) \rightarrow$

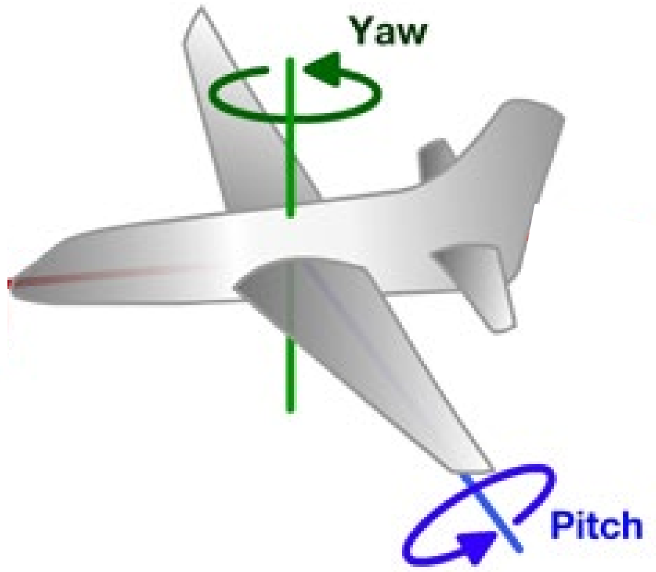
$$\boxed{\sin\theta_2 = \pm\sqrt{1\ - \cos^2\theta_2}}$$

  - Substitute $\cos\theta_2$ and $\sin\theta_2$ in (1) and (2)

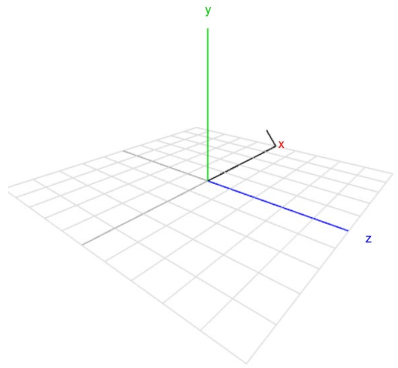
$$\boxed{\cos\theta_1 = \frac{1}{x^2+y^2}\left(x\ (a_1 + a_2\cos\theta_2) \pm y\ a_2\ \sqrt{1-\cos^2\theta_2}\right)}$$

$$\boxed{\sin\theta_1 = \frac{1}{x^2+y^2}\left(y\ (a_1 + a_2\cos\theta_2) \mp x\ a_2\ \sqrt{1-\cos^2\theta_2}\right)}$$

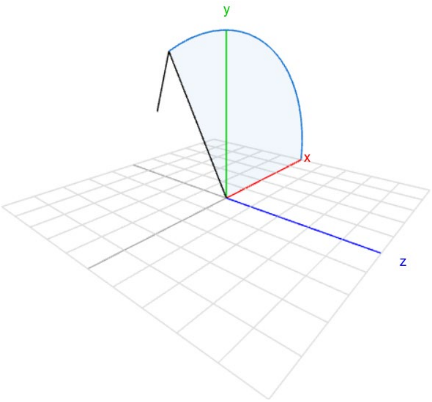
Why Rigid Body Rotation are 3D and not 2D?



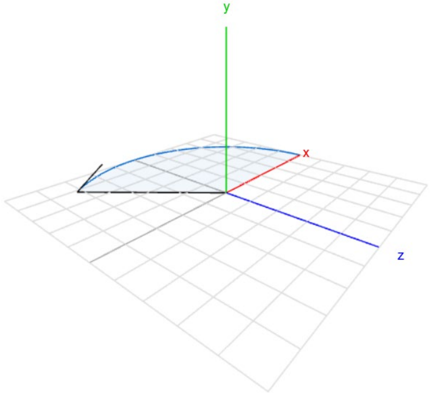
Why Rigid Body Rotation are 3D and not 2D?



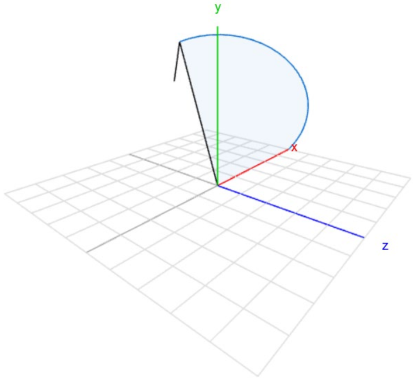
Rigid body attached to 0,0,0



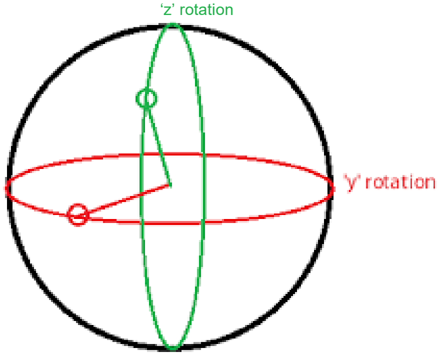
Rotate around Z axis



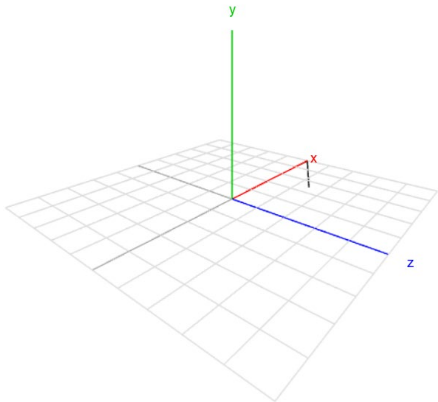
Rotate around Y axis



Rotate around Z and Y axis



Anywhere on sphere, with 2 rotations why need 3rd rotation?



Rotate around X axis,
Notice the tip of the vector!

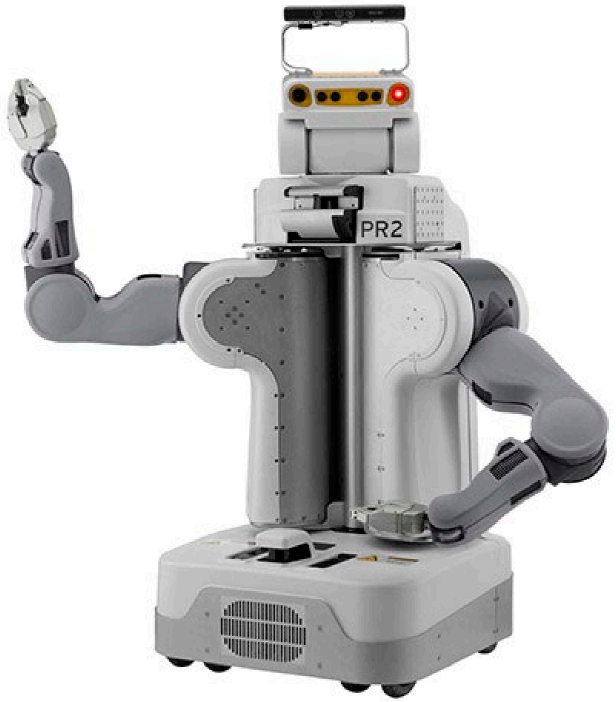
Figures from: https://quaternions.online/

Configuration Space

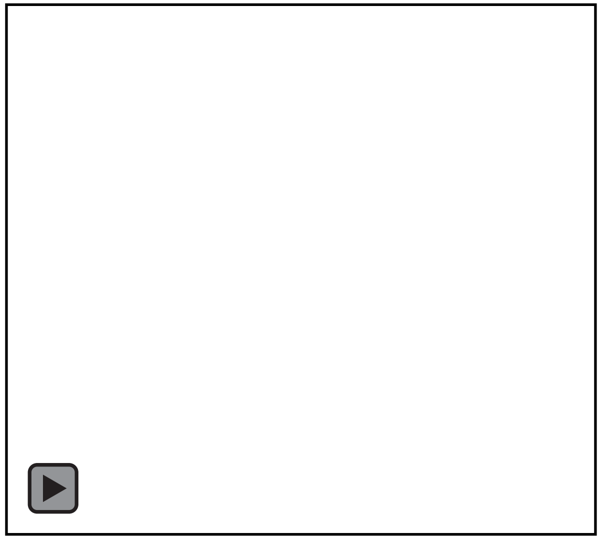
1. Represent Robot:
   I. Rigid body Representations (What shape are things)
   II. Rigid body transformations (Where objects are)
   III. Forward Kinematics (Where sets of things are)
2. **Configuration Space**
   1. C-Space Idea
   2. Topology of C-space
   3. C-space Obstacles
   4. C-Space Path Metrics, Constraints

What do we need to represent a robot

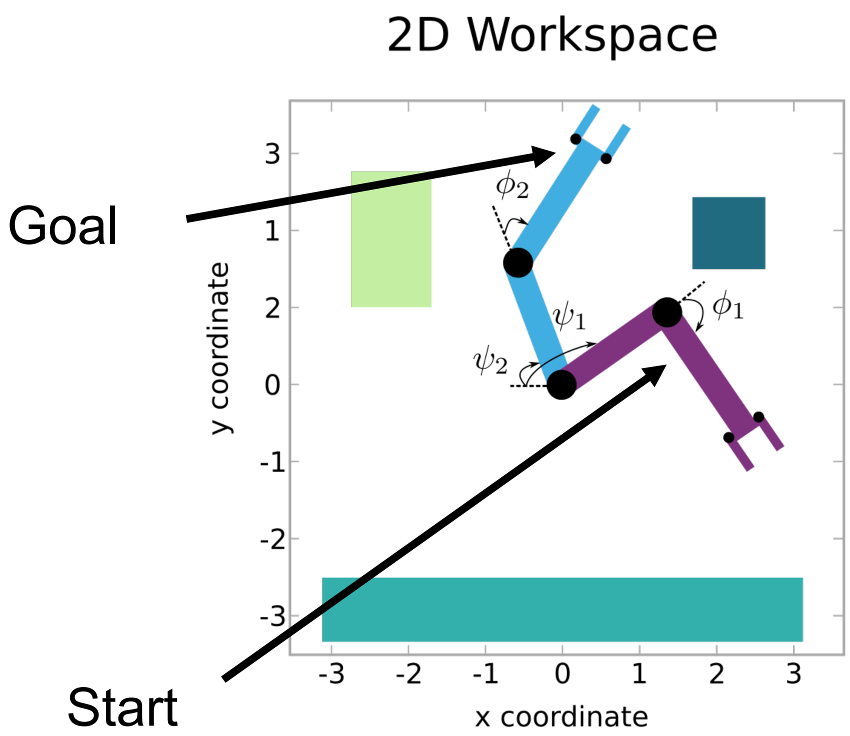
1. Rigid Body Geometry e.g. links (Fixed)
2. Kinematic Equations (Fixed)
3. Joint Values (Variable, Configurable?)



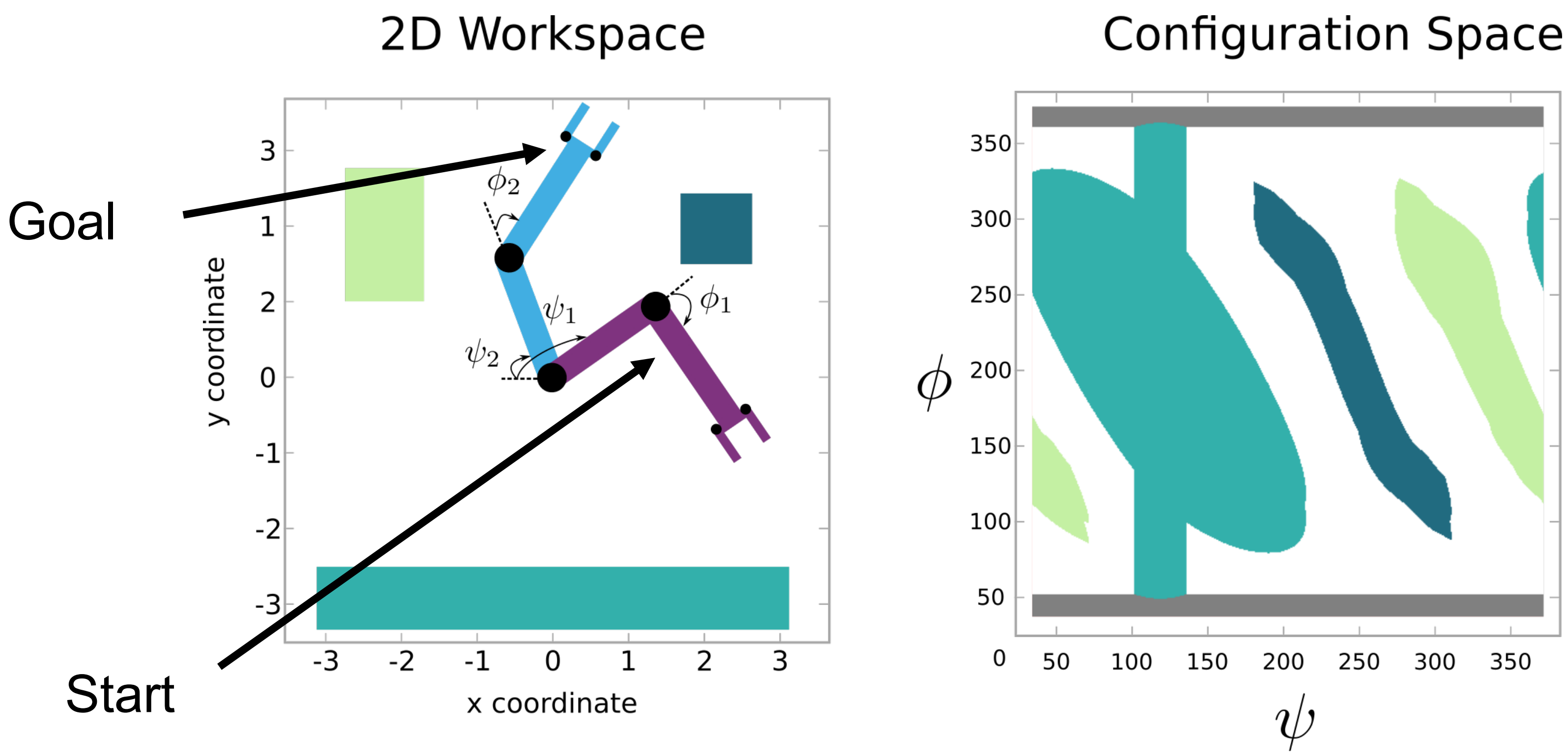
A Simple 2-link Articulated Robot



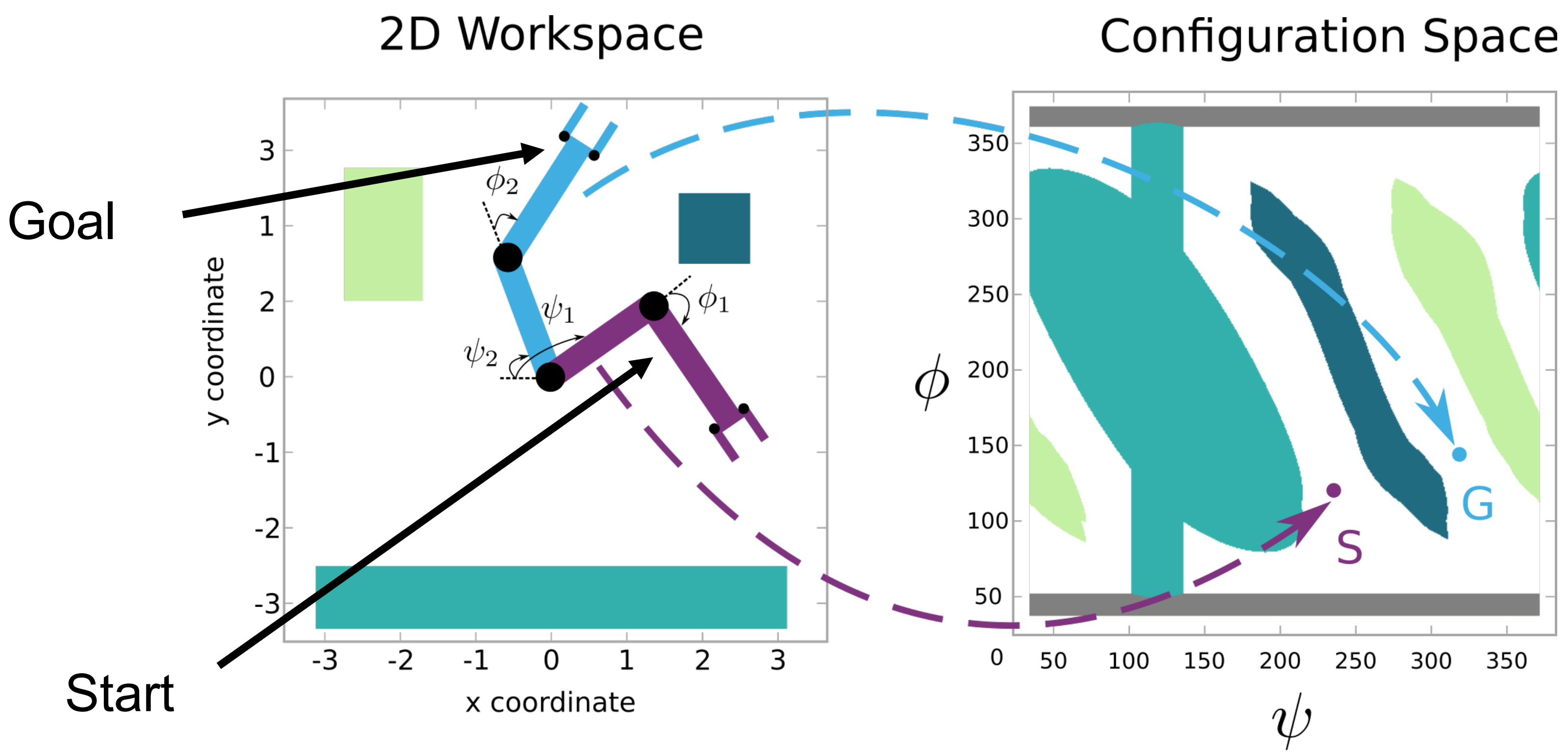
Big Idea: Configuration Space



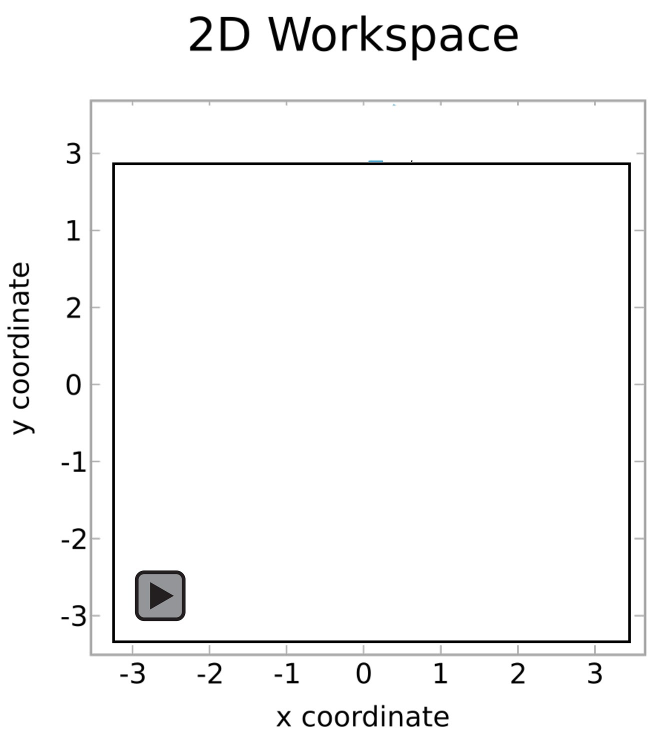
Big Idea: Configuration Space

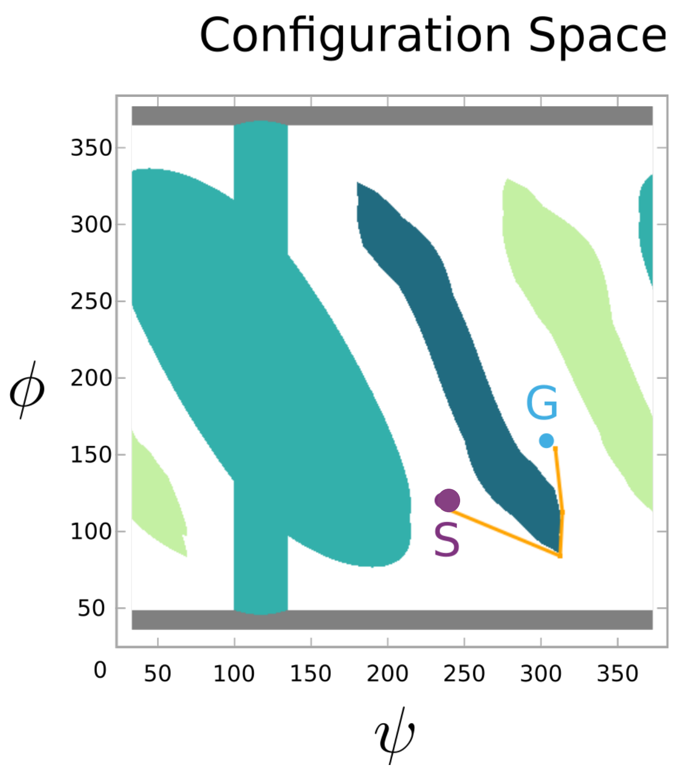


Big Idea: Configuration Space



Valid Path in Configuration space is valid in the Workspace





Now we can use point Planning Algorithms to plan in the C-Space!

Configuration Space Definitions

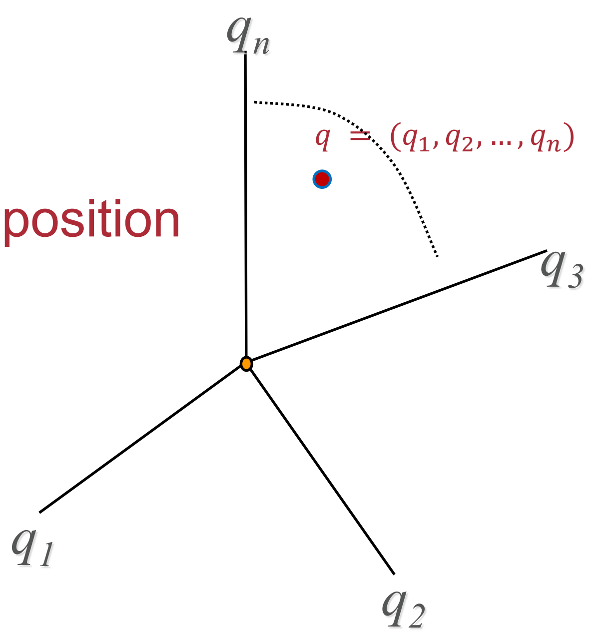
- The configuration of a moving object is a specification of the position of every point on the object.

- Usually a configuration is expressed as a vector of position & orientation parameters:

$$q = (q_1, q_2, \dots, q_n)$$

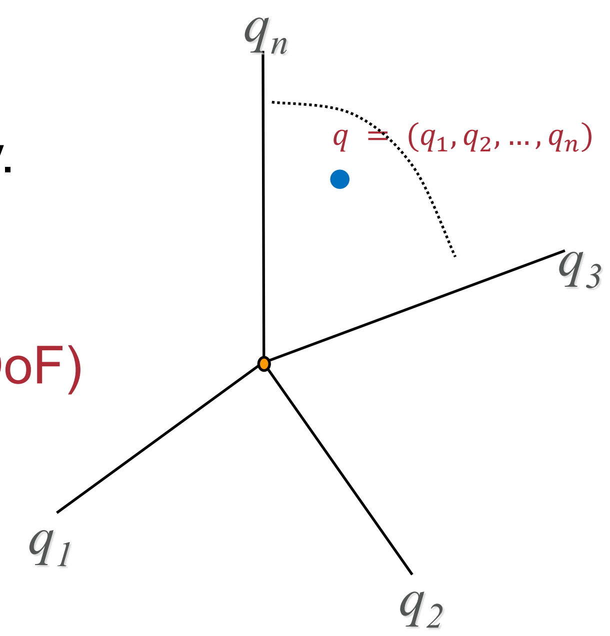
- The configuration space C is the set of all possible configurations.

- A configuration is a point in C



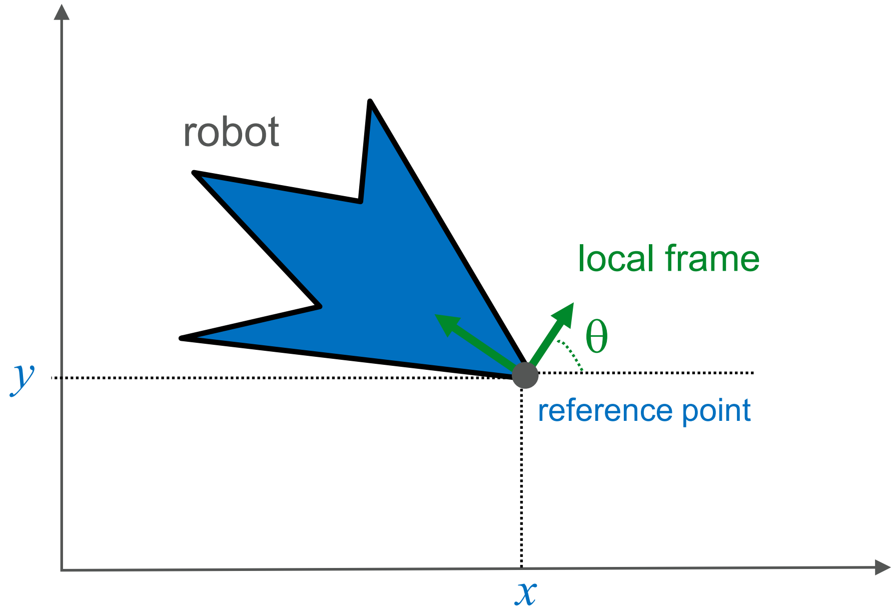
Configuration Space

- The dimension of a configuration space:
  - The minimum number of parameters needed to specify the configuration of the object completely.
  - also called the number of degrees of freedom (DoF) of a moving object (Robot).



Configuration Space Dimension of a 2D Rigid Body

- Example: rigid body in 2-D workspace



- 3-parameter specification:

$$q = (x, y, \theta) \text{ with } \theta \in [0, 2\pi)$$

- 3-D configuration space (3 DoF)

Degrees of Freedom (C-space Dim) a 2D Rigid Body

- Example: rigid body in 2-D
  - 4-parameter specification:

$$q = (x, y, u, v) \quad \text{with} \quad u^2 + v^2 = 1$$

Note: $u = \cos\theta$ and $v = \sin\theta$

- Dim of configuration space = 3

- Does the dimension of the configuration space (number of DoF) depend on the parametrization?

Degrees of Freedom (C-space Dim) for 3D Rigid Body

Example: rigid body in 3-D

- How many parameters in C-space?

$$q = (x, y, z, q')$$

- Parametrization of orientations by matrix:

$$q' = (r_{11}, r_{12}, \dots, r_{32}, r_{33})$$

where $r_{11}, r_{12}, \dots, r_{33}$ are the elements of rotation matrix

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}$$ with

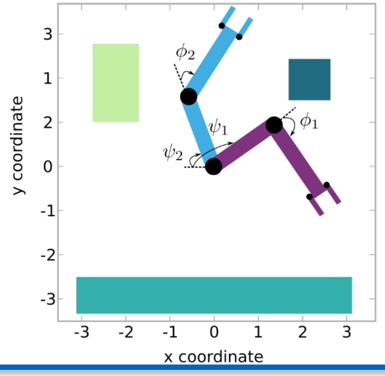
- $r_{1i}^2 + r_{2i}^2 + r_{3i}^2 = 1$ for all i
- $r_{1i}r_{1j} + r_{2i}r_{2j} + r_{3i}r_{3j} = 0$ for all $i \neq j$
- $\det(R) = +1$

- C-Space Dim: 6

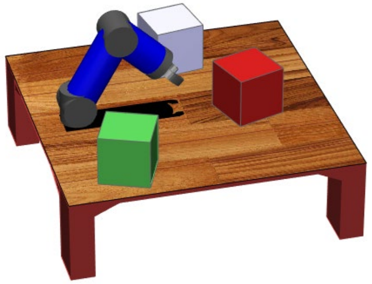
Configuration Spaces for Robots

Workspace

2-Dof Robot

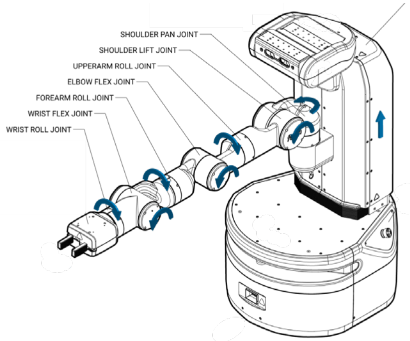


3-Dof Robot

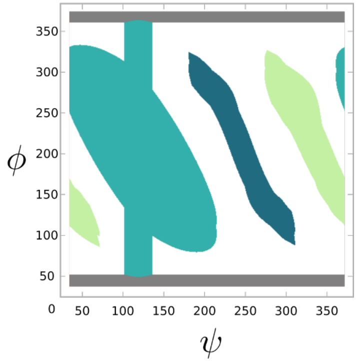


Das et al, *TRO* 2020

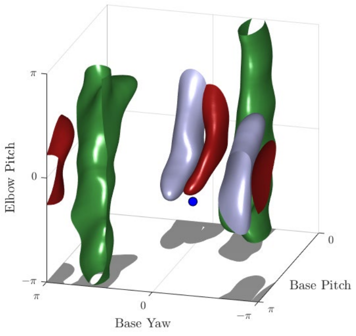
8-Dof Robot



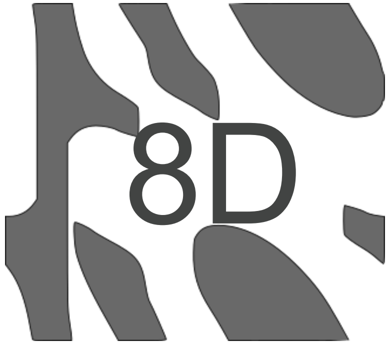
Configuration Space



2 Dimensions



3 Dimensions

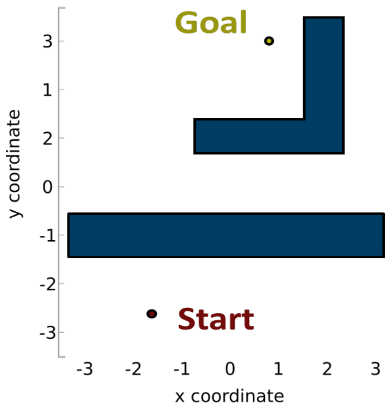


8 Dimensions

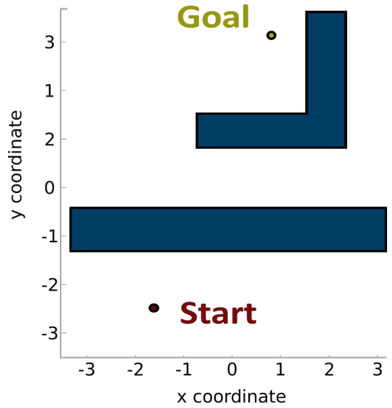
Are these the same C-space?

2D Point Robot

Workspace



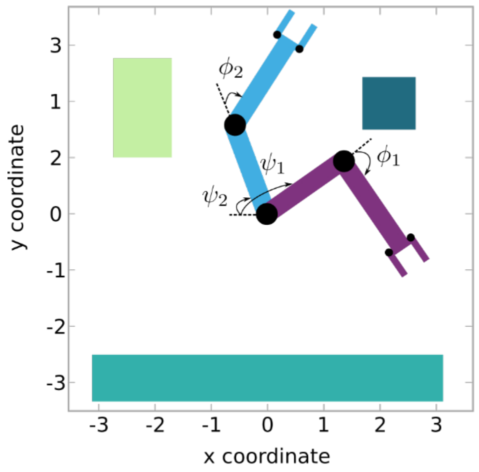
Configuration Space

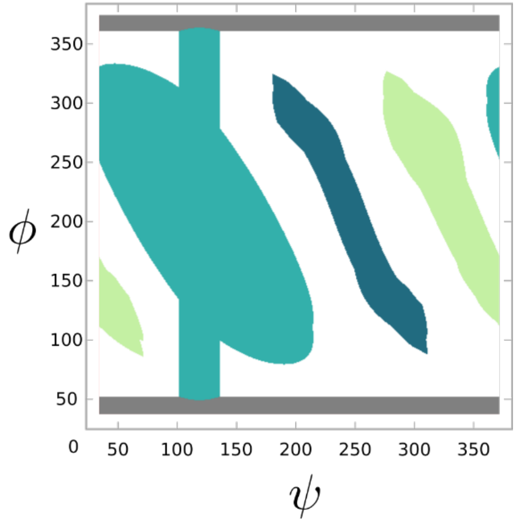


C-space Parameterization:

$$q = (x, y) \; with$$
$$x \in [0, 2\pi]$$
$$y \in [0, 2\pi]$$

2-Link Robot

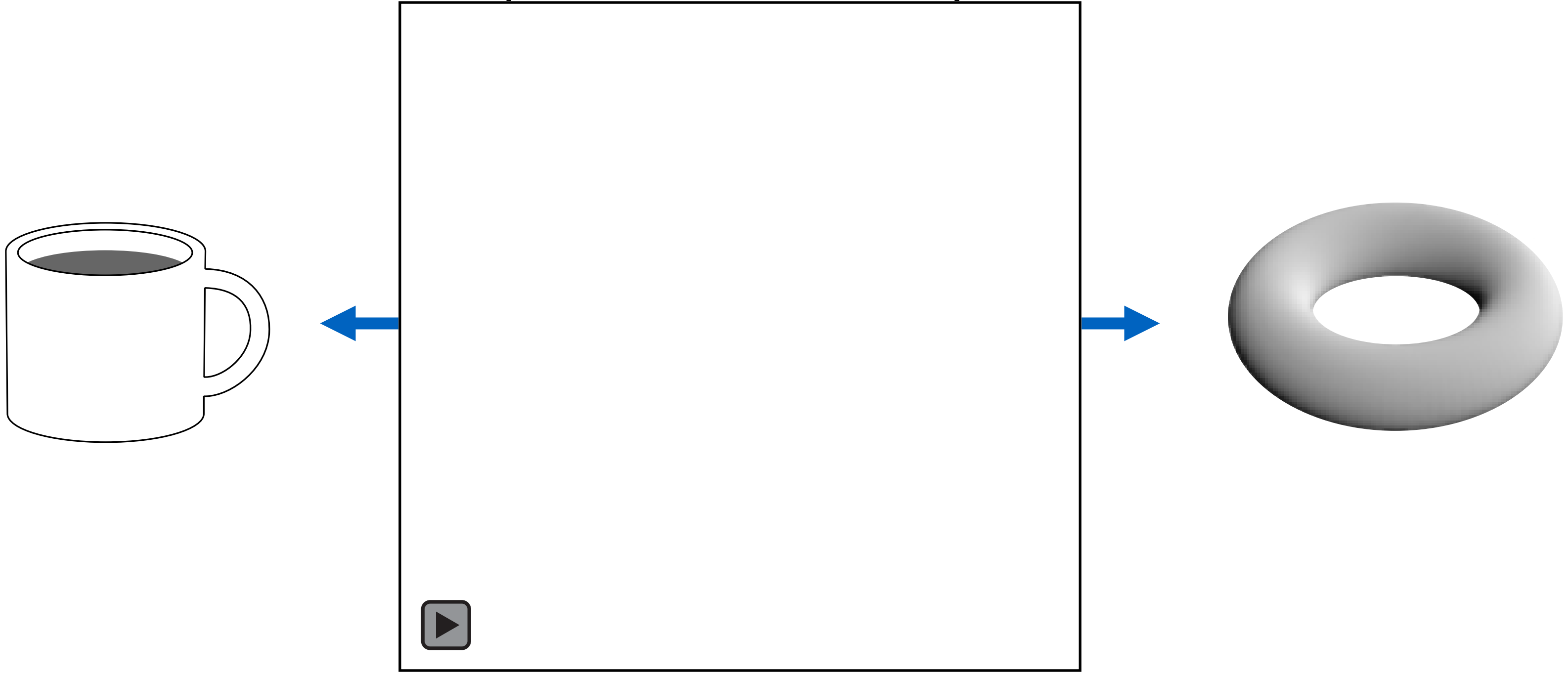




$$q = (\phi, \psi) \; with$$
$$\phi \in [0, 2\pi]$$
$$\psi \in [0, 2\pi]$$

Topology: Characterizing High-Dim Spaces

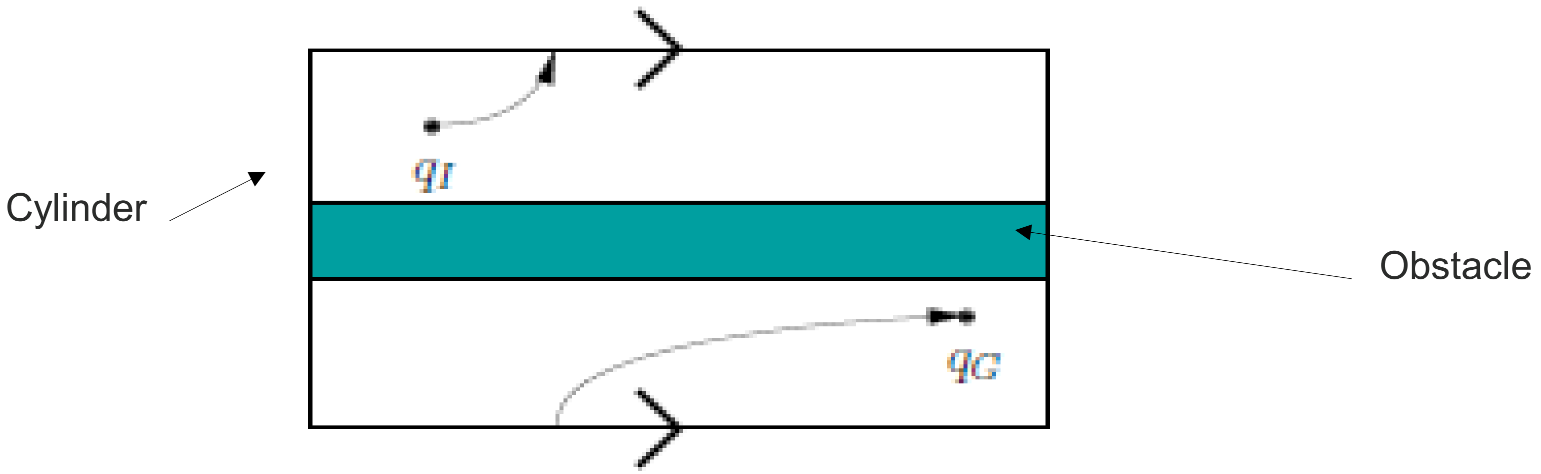
- Transform Coffee cup into a donut using smooth transformations



No cutting, tearing or pasting !!

Why Study Topology?

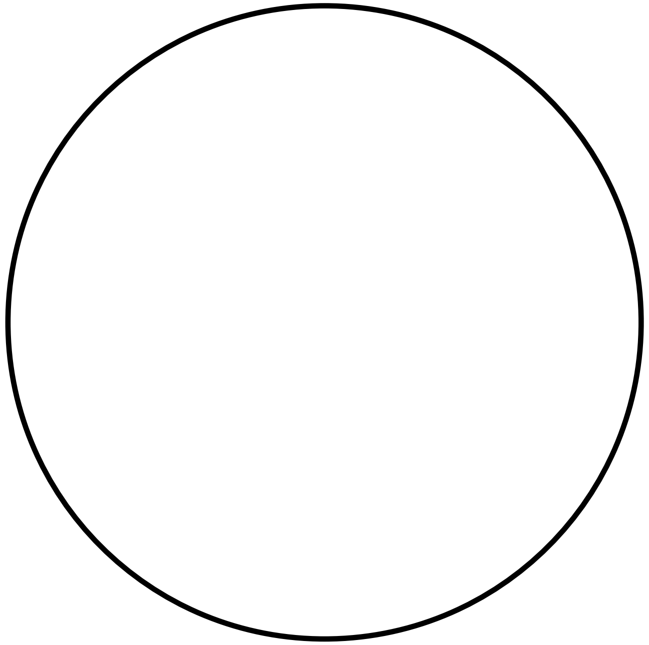
- Extend results from one space to another: **Donut to Coffee cup**
- Impacts the representation of the configuration space
- Helps to classify the class of planning problems that can be solved by a particular planning algorithm
- Serious problems can arise if the topology of the space is ignored



Does there exist a path in the configuration space for the robot from q_I to q_G?

Fundamental 1D spaces, $\mathbb{R}, S$

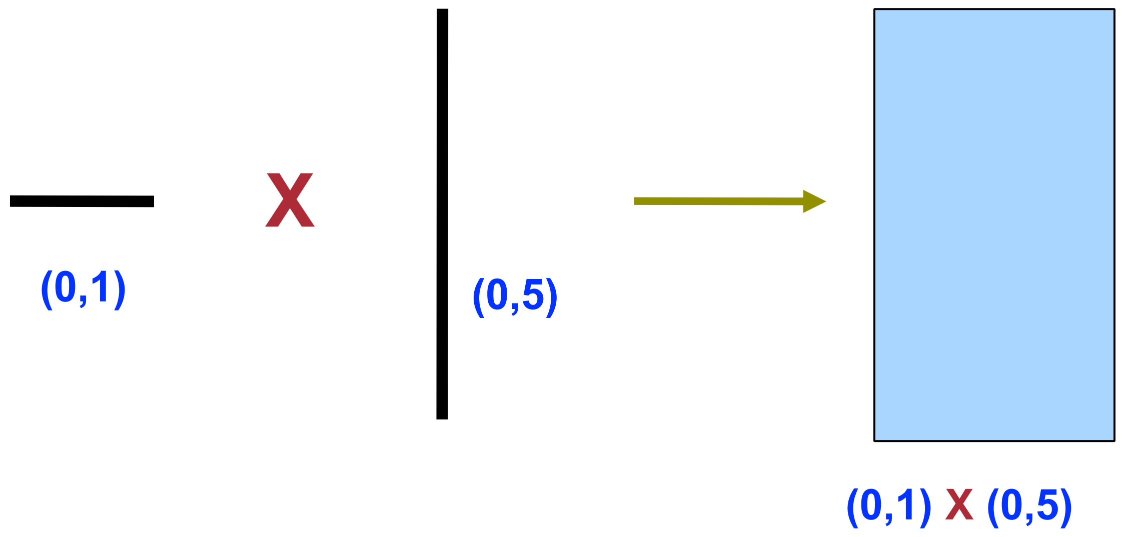
The Real Line
Symbol: $\mathbb{R}$



Boundary of a 2D circle
Symbol: S

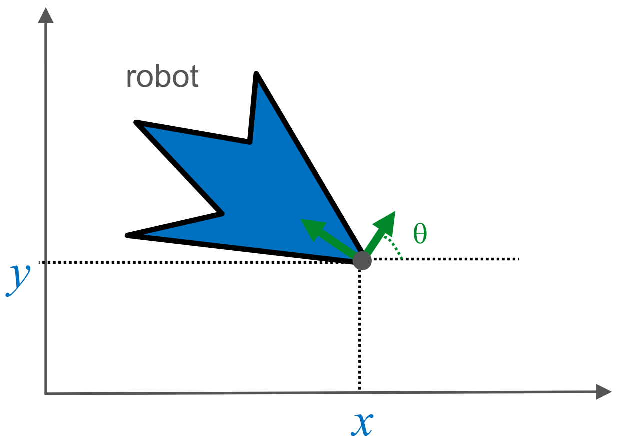
Cartesian Products

- **Cartesian Products:** Given two topological spaces ***X***, ***Y***, ***X x Y*** is the cartesian product
- A point ***x*** from ***X***, and ***y*** from ***Y*** each generates a point (x,y) in ***X x Y***
- Topology of ***X x Y*** comes from topology of ***X*** and ***Y***

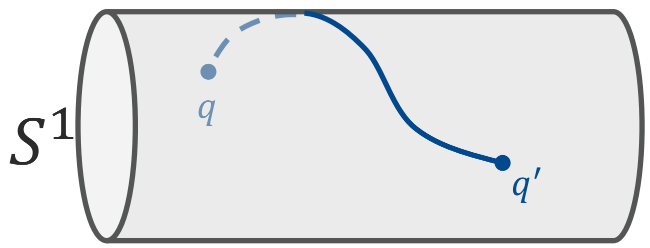


Topology of Configuration Space

- The topology of C is usually not that of a Cartesian space $\mathbb{R}^n$
- Here the robot **can only move in the x direction** and



3-D cylinder

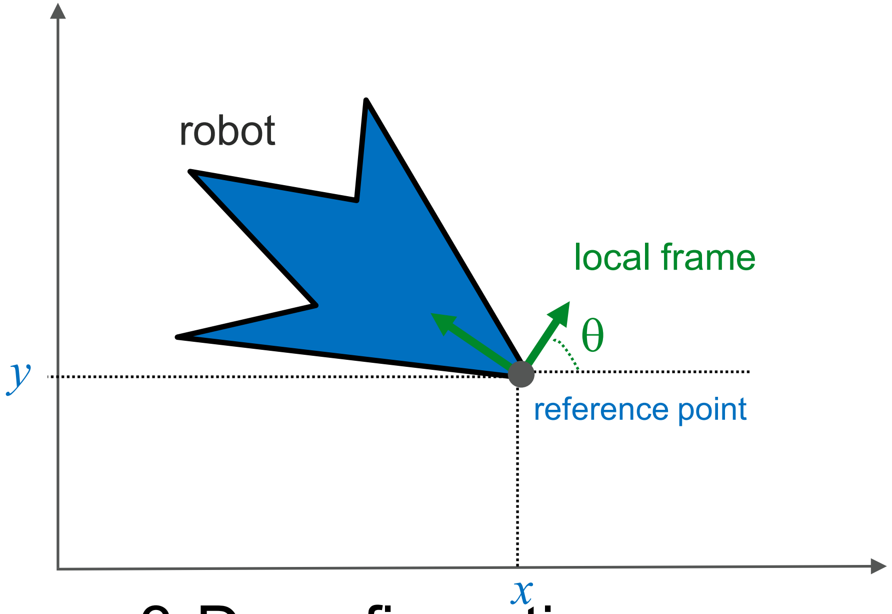


$\mathbb{R} \times S^1$

(note this is not $\mathbb{R}^2 \times S^1$)

C-Space Topology of a 2D Rigid Body

- Example: rigid body in 2-D workspace
- Here the robot **can move both in the x direction y direction** and rotate



- 3-D configuration space:

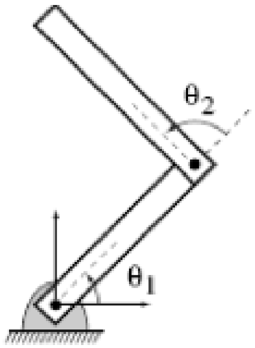
$$q = (x, y, \theta) \text{ with } \theta \in [0, 2\pi)$$

- Topology Space: a 3-D cylinder $C = \mathbb{R}^2 \times S^1$

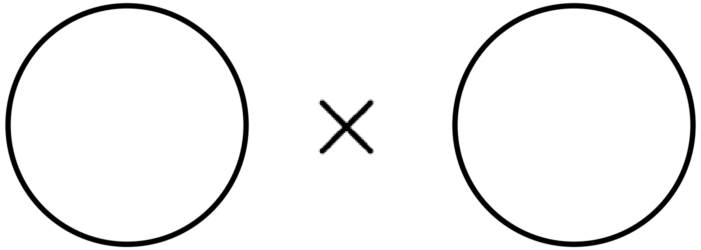
$\mathbb{R}^2 \times S^1$

C-Space Topology of a 2-link manipulator

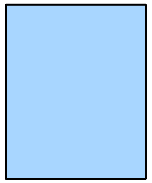
A 2-link manipulator that can self-intersect



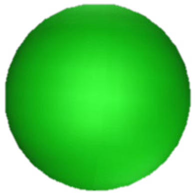
Topology: $S^1 \times S^1$



What is this space?



2D plane?



2D sphere?

2D Torus?

?

Something else?

What is the topology of a 2 link manipulator? (self intersections are allowed)

0

- 2D plane
- 2D Sphere
- 2D Torus
- Something Else

What is the topology of a 2 link manipulator? (self intersections are allowed)

0

2D plane

0%

2D Sphere

0%

2D Torus

0%

Something Else

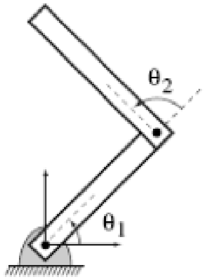
0%

What is the topology of a 2 link manipulator? (self intersections are allowed)

0

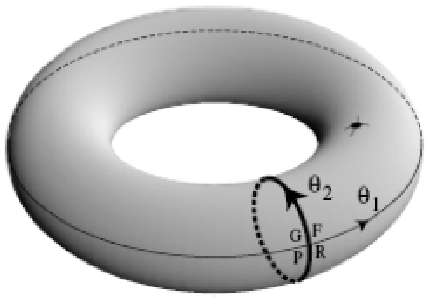
2D plane	0%
2D Sphere	0%
2D Torus	0%
Something Else	0%

C-Space Topology of a 2-link manipulator

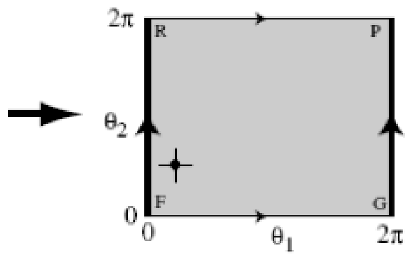


(a)

$$T = S^1 \times S^1$$



(b)

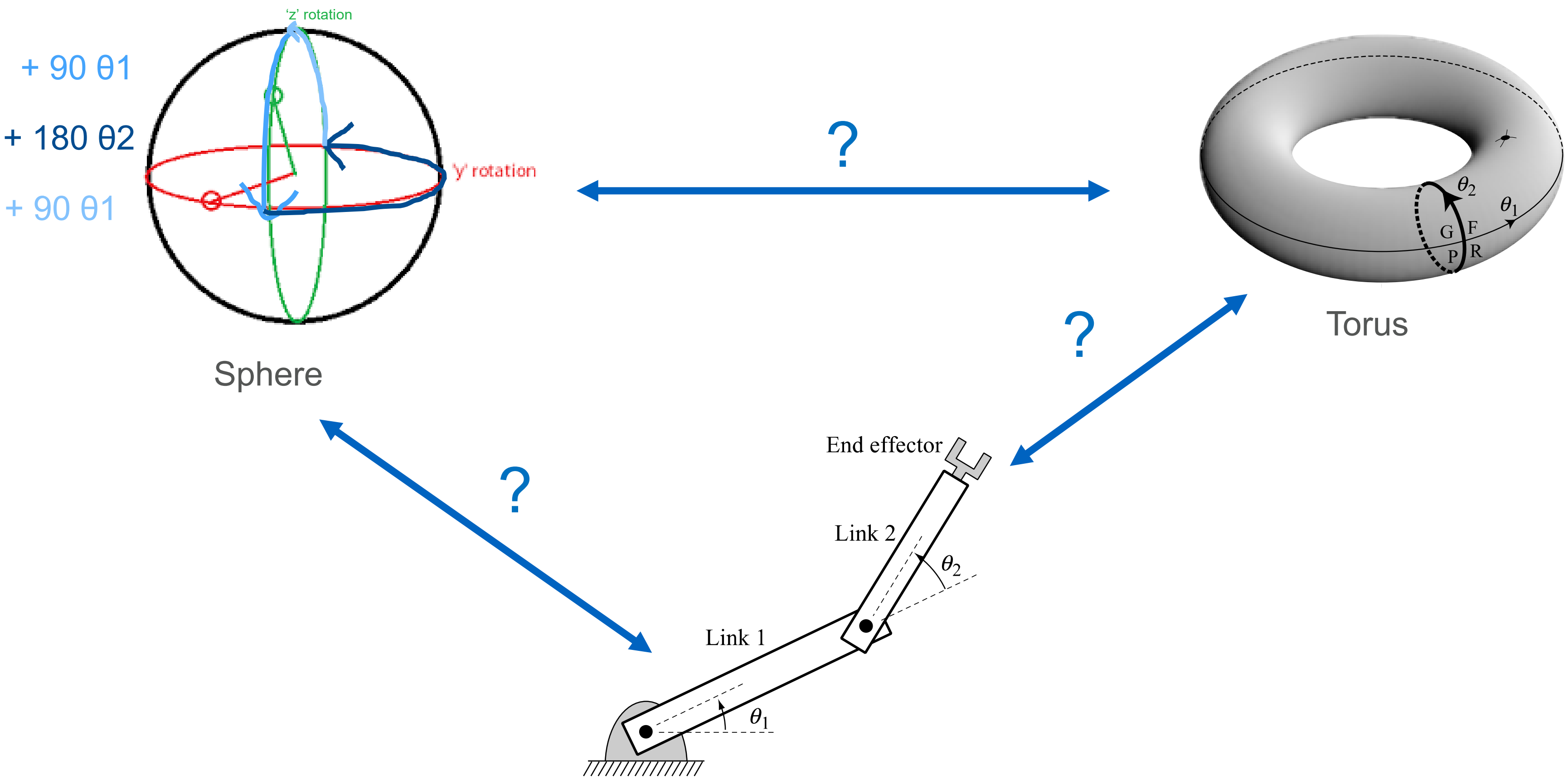


(c)

$(\theta_1, \theta_2) \in \mathbb{R}^2$;

problems at $\theta_i = \{0, 2\pi\}$.

Why 2 link manipulator is a Torus and not a Sphere ?



+ 90 rotation Θ1,+ 180 Rotation Θ2, + 90 rotation Θ1 does not result in the same configuration on this robot!

Topological Equivalence (Homeomorphism)

- **Interval Homeomorphism:** Any open interval of real line is homeomorphic to any other open interval

$$x \mapsto 5x$$

(0,1) $\cong$ (0,5)

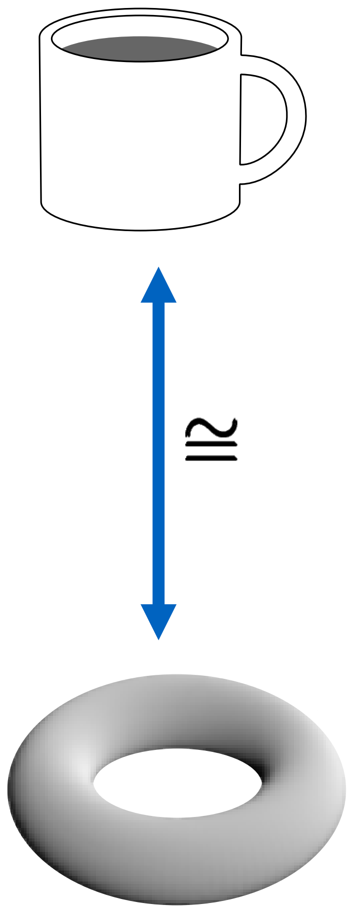
(0,1) $\ncong$ (0,1]

$$x \mapsto 1/x$$

(0,1) $\cong$ (1,∞)

$$x \mapsto \tan^{-1}(\pi x/2)$$

(-1,1) $\cong$ (-∞,∞)



Special Orthogonal Group SO(2) and SO(3)

Special Orthogonal Group SO(2), SO(3), also known as the group of rotation matrices, is the set of all 2x2, 3x3 (respectively) real matrices that satisfy (a) $R^T R = I$ and det R = 1

(A group consists of a set of elements and an operation –matrix multiplication here- such that for all A, B in the group, the following properties are satisfied (a) closure, (b) associativity, and (c)identity element existence.)

Special Euclidean Group SE(2)

Special Euclidean Group SE(2) characterizes rotations and translations for 2D rigid bodies through a single matrix.

$$\left\{ \begin{pmatrix} R(\theta) & t \\ 0 & 1 \end{pmatrix} \middle| R(\theta) \in SO(2), t \in \mathbb{R}^2 \right\}$$

$R^2 x\, SO(2)$: topological representation

SE(2): matrix group representation

We will use interchangeably: SE(2) $\sim R^2\, x\, SO(2)$

Special Euclidean Group SE(3)

Special Euclidean Group SE(3) characterizes rotations and translations for 3D rigid bodies through a single matrix.

$$\left\{ \begin{pmatrix} \boldsymbol{R}(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}) & \boldsymbol{t} \\ \mathbf{0} & \mathbf{1} \end{pmatrix} \mid \boldsymbol{R}(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}) \in \boldsymbol{SO}(\mathbf{3}), \boldsymbol{t} \in \boldsymbol{R}^{\mathbf{3}} \right\}$$

$R^3 x\ SO(3)$: topological representation

SE(3): matrix group representation

We will use interchangeably: SE(3) $\sim R^3\ x\ SO(3)$

Recap:Topological Spaces

- Some important topological spaces:
  - $\mathbb{R}$: real number line
  - $\mathbb{R}^n$: n-dimensional Cartesian space
  - S^1: boundary of circle in 2D
  - S^2: surface of sphere in 3D
  - $SO(2)$, $SO(3)$: set of 2D, 3D orientations (special orthogonal group)
  - $SE(2)$, $SE(3)$: set of rigid 2D, 3D translations and rotations (special Euclidean group)
  - $A \times B$: Cartesian product, power notation $A^n = A \times A \times \cdots \times A$
  - $T = S^1 \times S^1$: torus
- Homeomorphism $\sim$ denotes topological equivalence
  - Continuous mapping with continuous inverse (bijective)
    - Cube $\sim S^2$
    - $SO(2) \sim S^1$
    - $SE(3) \sim \mathbb{R}^3 \times SO(3)$

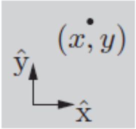
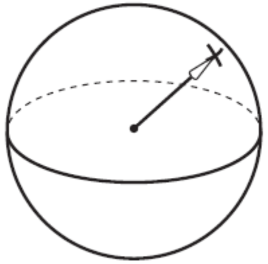
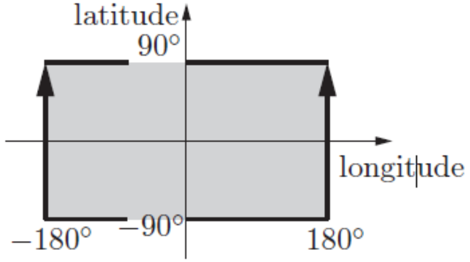
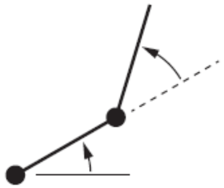
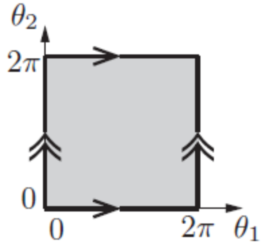
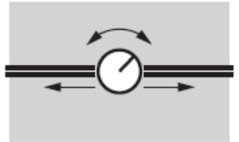
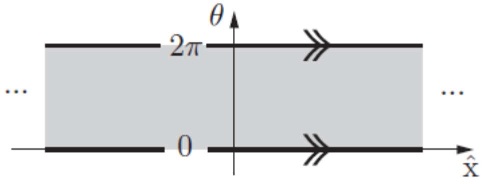
Topology of some common robots C-spaces

- Examples of some common robots

Type of robot	Representation of C-Space
Mobile robot translating in the plane	$\mathbb{R}^2$
Mobile robot translating and rotating in the plane	$SE(2)$ or $\mathbb{R}^2 \times S^1$
Rigid body translating in 3-D	$\mathbb{R}^3$
A spacecraft	$SE(3)$ or $\mathbb{R}^3 \times SO(3)$
An n-joint revolute arm	T^n
A planar mobile robot with an attached n-joint arm	$SE(2) \times T^n$

- Note that:
- $S^1 \times S^1 \times \cdots \times S^1 = T^n$, n-dimensional torus
- $S^1 \times S^1 \times \cdots \times S^1 \neq S^n$, n-dimensional sphere
- $S^1 \times S^1 \times S^1 \neq SO(3)$
- $SE(2) \neq \mathbb{R}^3$
- $SE(3) \neq \mathbb{R}^6$

Some more Topology Examples

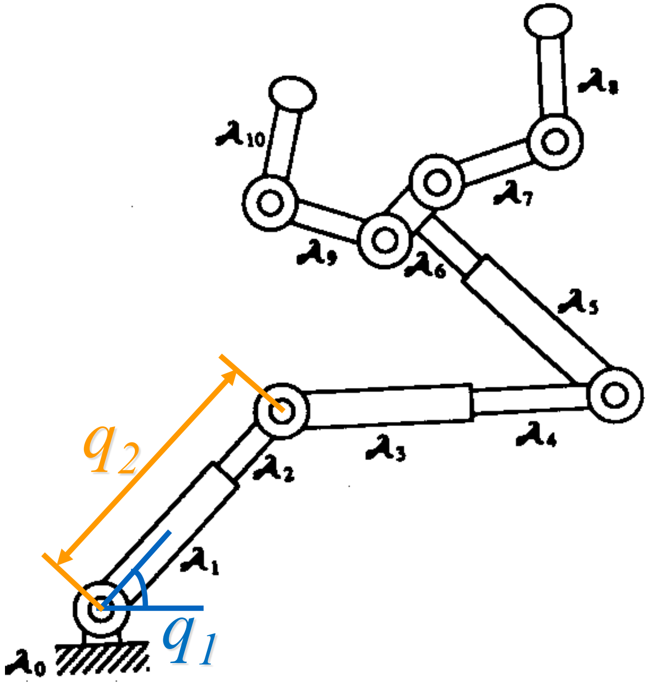
system	topology	sample representation
point on a plane	 $\mathbb{E}^2$	 $\mathbb{R}^2$
 spherical pendulum	 S^2	 $[-180°, 180°) \times [-90°, 90°]$
 2R robot arm	 $T^2 = S^1 \times S^1$	 $[0, 2\pi) \times [0, 2\pi)$
 rotating sliding knob	 $\mathbb{E}^1 \times S^1$	 $\mathbb{R}^1 \times [0, 2\pi)$

Topology of C-Spaces

- Example: articulated robot
  - An articulated object is a set of rigid bodies connected at the joints.

C-space: $(S^1)^7 \times I^3$

I: Real intervals



Topology: Some fun videos

https://www.youtube.com/watch?v=VOKgMJEc_ro
https://www.youtube.com/watch?v=AmgkSdhK4K8
https://www.youtube.com/watch?v=aTZBO8Ql6rc

Path in Configuration Space must avoid C-space obstacles