Chapter - 1
Configuration Space

1. Introduction to Configuration Space

configuration of space means where is the robot.

1.1 What is Configuration Space (C-Space)?

In robotics, the **Configuration Space (C-Space)** is a mathematical representation of all possible positions and orientations a robot can take. It abstracts away the physical dimensions of the robot and represents its state using a set of **configuration variables**.

- **Definition**:
  C-Space is an **n-dimensional space** where each point represents a possible configuration (pose) of the robot.
- **Mathematical Representation**:
  If a robot has n **degrees of freedom (DOF)**, then its configuration can be expressed as a vector:

$$q = (q_1, q_2, \ldots, q_n) \in \mathbb{R}^n$$

  where each qi represents a joint parameter (e.g., an angle or displacement). R^n represents the configuration space.

1.2 Configuration Space vs. Workspace

- **Workspace** (Task Space): The physical space in which the robot operates. It is represented in **Cartesian coordinates** (x, y, z).
- **Configuration Space**: The space of all possible robot configurations, represented using **joint parameters**.

Example: A Simple Robotic Arm

Consider a **2-link planar manipulator** with two **revolute joints**:

- **Workspace Representation**:
  The end-effector moves in a 2D plane, so its position is given by **(x, y)**.
- **Configuration Space Representation**:
  The two joint angles, q1 and q2, define the arm's posture:

q=(q1,q2)

This means the C-Space is **2D**, where each axis represents a joint angle.

1.3 Degrees of Freedom (DOF) and Configuration Variables

minimum real number for configure the robot.

The **Degrees of Freedom (DOF)** define the number of independent parameters needed to specify the robot's configuration.

- **A free-moving object in 3D space has 6 DOF**:
  - 3 for position (**x, y, z**)
  - 3 for orientation (**roll, pitch, yaw**)
- **Examples**:
  - A **prismatic joint** (sliding) contributes **1 DOF** (linear displacement).
  - A **revolute joint** (rotating) contributes **1 DOF** (rotation angle).

Example: DOF of Different Robots

Robot Type	DOF	Configuration Variables
1-Link Arm	1	q1 (angle)
2-Link Arm	2	(q1, q2)
3D Drone	6	(x, y, z, θx, θy, θz)

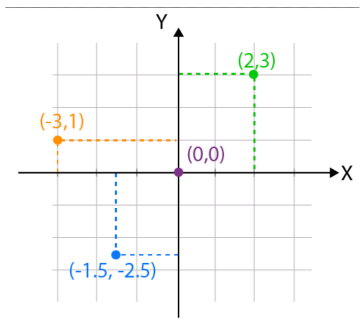
1.4 Representation of Configuration Space

1. **Discrete vs. Continuous C-Space**:
   - Discrete C-Space: A grid-based representation.
   - Continuous C-Space: Uses real numbers for smooth motion.
2. **Dimensionality of C-Space**:
   - A **single-jointed robot** has a **1D** C-Space.
   - A **2-joint planar robot** has a **2D** C-Space.
   - A **6-DOF robot arm** has a **6D** C-Space.

2. Types of Coordinate Systems in Robotics

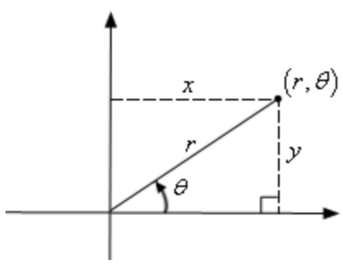
A robot's position and orientation can be represented in different **coordinate systems**. The three most common ones are:

2.1 Cartesian (Rectangular) Coordinate System



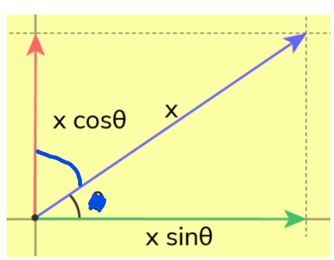
- Uses **(x, y, z)** to define a position.
- Represents position in **Euclidean space**.
- Used in **industrial robots, CNC machines, and Cartesian manipulators**.

2.2 Polar Coordinate System



- Uses **(r, θ)** to represent a point in 2D.
- rrr = Distance from the origin.
- θ\thetaθ = Angle with the reference axis.

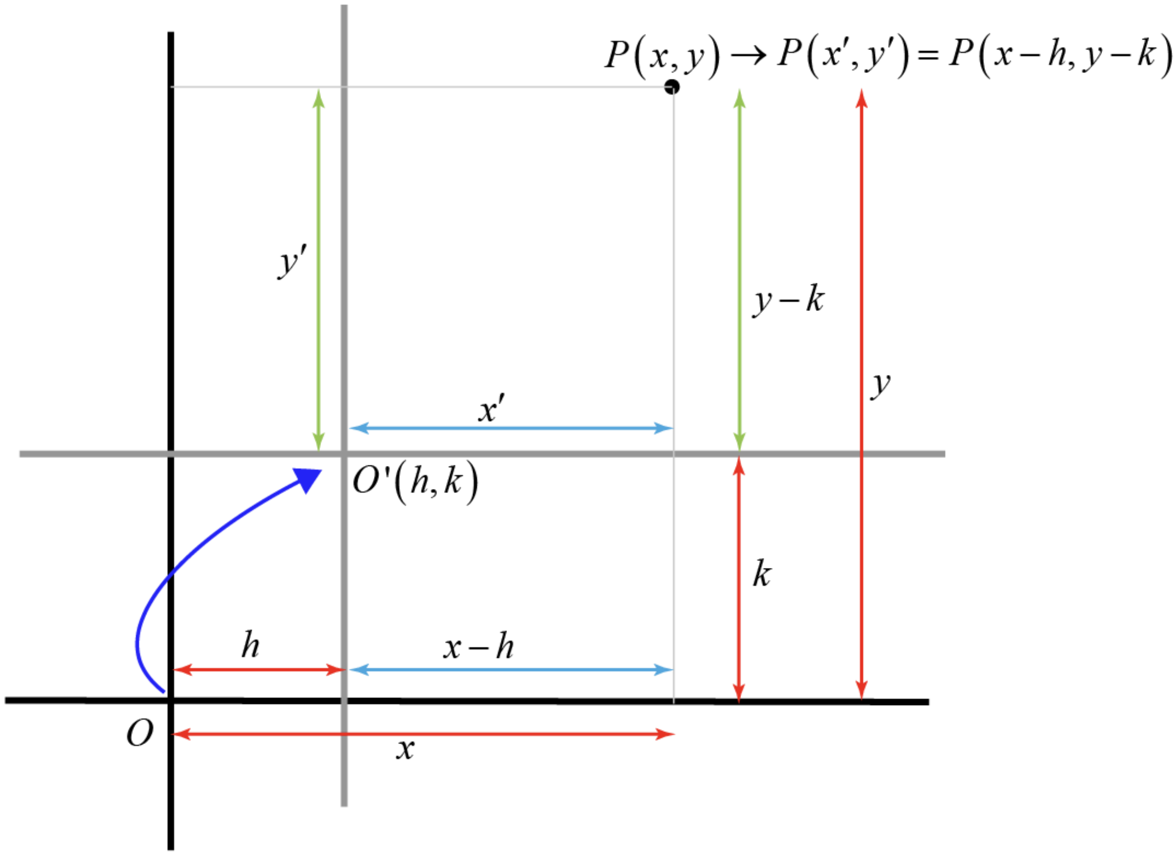
2.3 Vector Space Representation



- Represents robot configurations as **vectors in an n-dimensional space**.
- Each joint or coordinate contributes to a dimension.

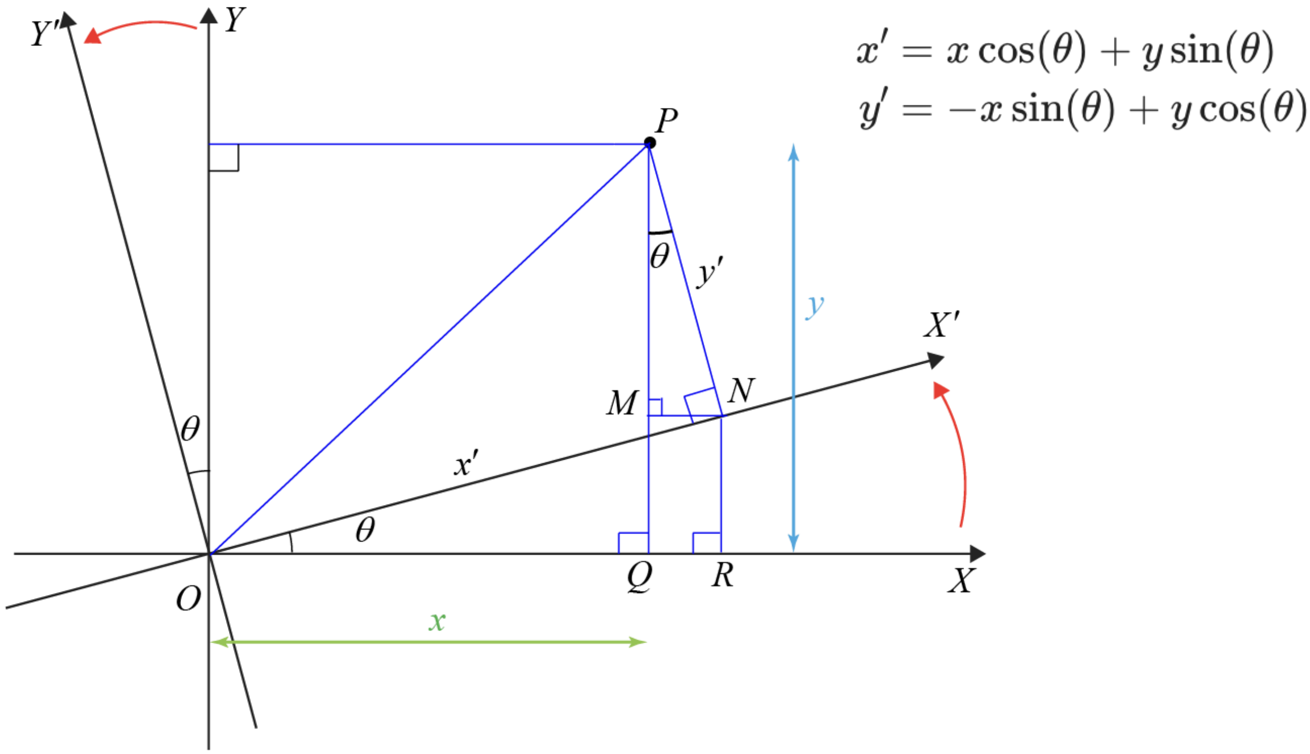
2.4 Transformations of Axes

2.4.1 Translation



Consider a Cartesian coordinate system with its origin at O. Let P be a point within this system, having coordinates (x,y). Now, if we shift the origin to a new point O′(h,k) without changing the orientation in the original system, we establish a new coordinate system. In this new system, the coordinates of O′ become (0,0), indicating its position as the new origin. As a result of this shift, the coordinates of point P will change to accommodate the new frame of reference. Suppose the new coordinates of P in this shifted system are (x′,y′). Given the translation of the origin and maintaining the orientation of the axes, it follows that the relationship between the old and new coordinates of P can be expressed as $x'=x-h$ and $y'=y-k$. This transformation effectively recalibrates the coordinate system, considering O′ as the new reference point, thus altering the perceived position of P relative to this new origin.

2.4.2 Rotation



From the construction, line PM is parallel to the OY axis and line PN is parallel to the OY' axis, which implies $\angle MPN = \theta$ since it corresponds to the angle between OY and OY'.

In $\triangle PMN$, the length MN can be determined using y' and θ:

$$MN = y'\sin(\theta)$$

In $\triangle ONR$, which is right-angled at N, OR represents x' projected onto the OX axis:

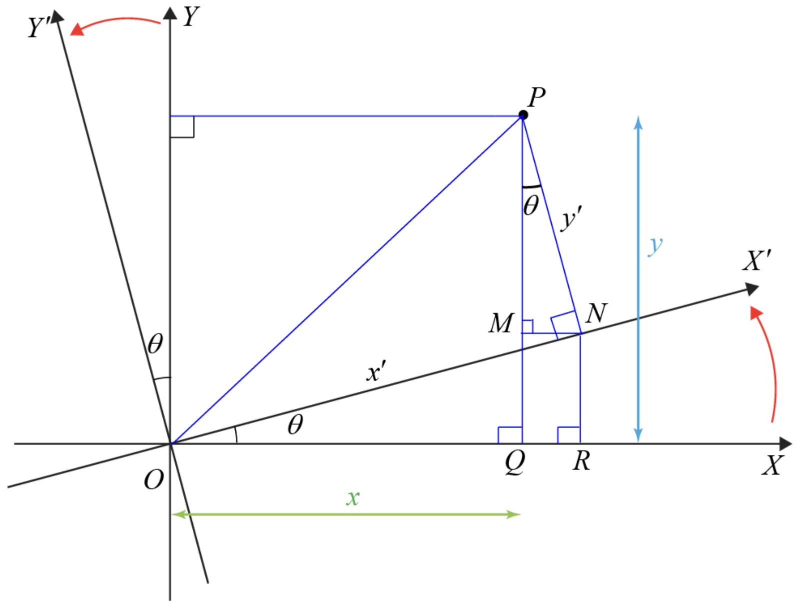
$$OR = x'\cos(\theta)$$

The horizontal distance OQ can be found by subtracting MN from OR:

$$OQ = OR - MN = x'\cos(\theta) - y'\sin(\theta)$$

The vertical component NR in $\triangle ONR$ is the projection of x' on the OY axis:

$$NR = x'\sin(\theta)$$



For PQ, which is the vertical distance from the original OX axis to point P, we consider the vertical distances PM and MQ:

$$PM = y' \cos(\theta)$$

$$MQ = NR = x' \sin(\theta)$$

Summing these gives the y coordinate in the original system:

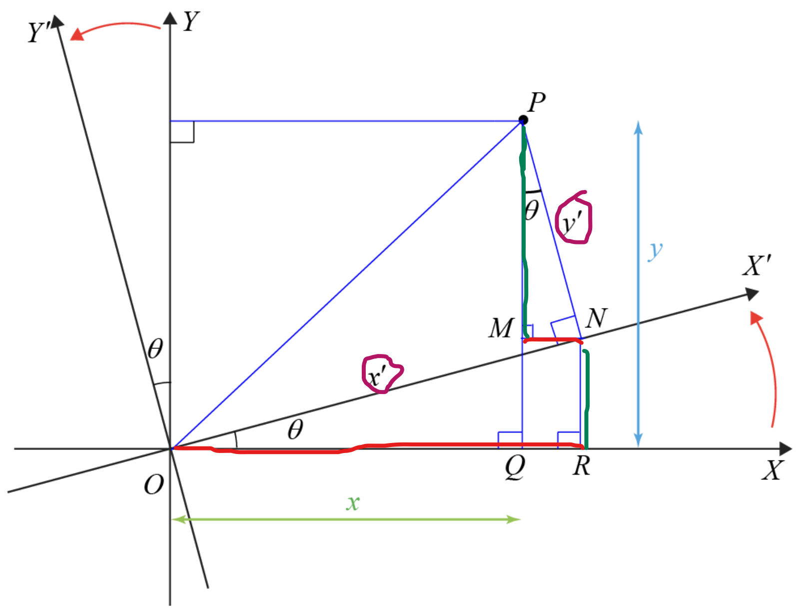
$$PQ = PM + MQ = y' \cos(\theta) + x' \sin(\theta)$$

Thus, the new coordinates (x, y) of point P after the rotation by θ in terms of the rotated coordinates (x', y') are:

$$\begin{aligned} x &= x' \cos(\theta) - y' \sin(\theta) \\ y &= x' \sin(\theta) + y' \cos(\theta) \end{aligned}$$

Starting with the equations for the coordinates (x, y) after an anticlockwise rotation by θ in terms of the new coordinates (x', y'):

$$\begin{aligned} x &= x' \cos(\theta) - y' \sin(\theta) \quad (1) \\ y &= x' \sin(\theta) + y' \cos(\theta) \quad (2) \end{aligned}$$



To express x' in terms of x and y, we multiply equation (1) by $\cos(\theta)$ and equation (2) by $\sin(\theta)$, then add the results:

$$x\cos(\theta) = x'\cos^2(\theta) - y'\sin(\theta)\cos(\theta) \quad (3)$$

$$y\sin(\theta) = x'\sin^2(\theta) + y'\sin(\theta)\cos(\theta) \quad (4)$$

Adding equations (3) and (4):

$$x\cos(\theta) + y\sin(\theta) = x'(\cos^2(\theta) + \sin^2(\theta))$$

$$x' = x\cos(\theta) + y\sin(\theta)$$

For y', we multiply equation (1) by $\sin(\theta)$ and equation (2) by $\cos(\theta)$, then subtract equation (1) from equation (2):

$$x\sin(\theta) = x'\sin(\theta)\cos(\theta) - y'\sin^2(\theta) \quad (5)$$

$$y\cos(\theta) = x'\sin(\theta)\cos(\theta) + y'\cos^2(\theta) \quad (6)$$

Now, subtract equation (5) from equation (6):

$$y\cos(\theta) - x\sin(\theta) = y'(\cos^2(\theta) + \sin^2(\theta))$$

$$y' = (y\cos(\theta) - x\sin(\theta))$$

Gr¨ubler's Formula:

The number of degrees of freedom of a mechanism with links and joints can be calculated using **Grübler's formula**, which is an expression of Equation (2.3).

Proposition 2.2. *Consider a mechanism consisting of N links, where ground is also regarded as a link. Let J be the number of joints, m be the number of degrees of freedom of a rigid body ($m = 3$ for planar mechanisms and $m = 6$ for spatial mechanisms), f_i be the number of freedoms provided by joint i, and c_i be the number of constraints provided by joint i, where $f_i + c_i = m$ for all i. Then*

Grübler's formula for the number of degrees of freedom of the robot is

$$\begin{aligned} \text{dof} &= \underbrace{m(N-1)}_{\text{rigid body freedoms}} - \underbrace{\sum_{i=1}^{J} c_i}_{\text{joint constraints}} \\ &= m(N-1) - \sum_{i=1}^{J}(m - f_i) \\ &= m(N-1-J) + \sum_{i=1}^{J} f_i. \end{aligned} \tag{2.4}$$

This formula holds only if all joint constraints are independent. If they are not independent then the formula provides a lower bound on the number of degrees of freedom.

[See Mathematical Examples from Reference Book.]

Configuration Space Topology:

Figure 2.9: An open interval of the real line, denoted (a, b), can be deformed to an open semicircle. This open semicircle can then be deformed to the real line by the mapping illustrated: beginning from a point at the center of the semicircle, draw a ray that intersects the semicircle and then a line above the semicircle. These rays show that every point of the semicircle can be stretched to exactly one point on the line, and vice versa. Thus an open interval can be continuously deformed to a line, so an open interval and a line are topologically equivalent.

The idea that the two-dimensional surfaces of a small sphere, a large sphere, and a football all have the same kind of shape, which is different from the shape of a plane, is expressed by the **topology** of the surfaces.

Topologically distinct one-dimensional spaces include the circle, the line, and a closed interval of the line. The circle is written mathematically as S or S^1, a one-dimensional "sphere." The line can be written as $\mathbb{E}$ or $\mathbb{E}^1$, indicating a one-dimensional Euclidean (or "flat") space. Since a point in $\mathbb{E}^1$ is usually represented by a real number (after choosing an origin and a length scale), it is often written as $\mathbb{R}$ or $\mathbb{R}^1$ instead. A closed interval of the line, which contains its endpoints, can be written $[a, b] \subset \mathbb{R}^1$. (An open interval (a, b) does not include the endpoints a and b and is topologically equivalent to a line, since the open interval can be stretched to a line, as shown in Figure 2.9. A closed interval is not topologically equivalent to a line, since a line does not contain endpoints.)

In higher dimensions, $\mathbb{R}^n$ is the n-dimensional Euclidean space and S^n is the n-dimensional surface of a sphere in $(n+1)$-dimensional space. For example, S^2 is the two-dimensional surface of a sphere in three-dimensional space.

Note that the topology of a space is a fundamental property of the space itself and *is independent of how we choose coordinates to represent points in the space.* For example, to represent a point on a circle, we could refer to the point by the angle θ from the center of the circle to the point, relative to a chosen zero angle. Or, we could choose a reference frame with its origin at the center of the circle and represent the point by the two coordinates (x, y) subject to the constraint $x^2 + y^2 = 1$. No matter what our choice of coordinates is, the space itself does not change.

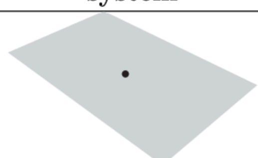
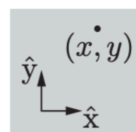
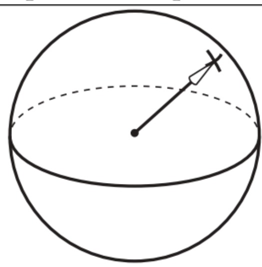
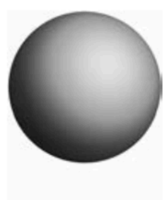
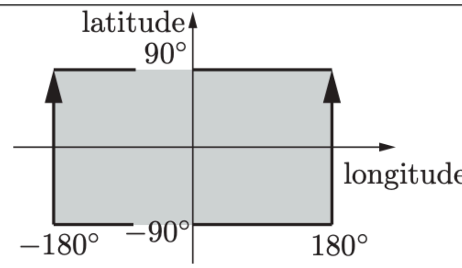
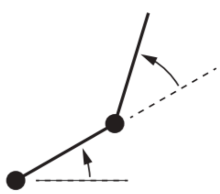
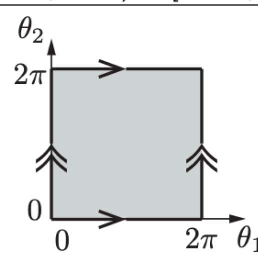
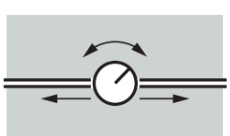
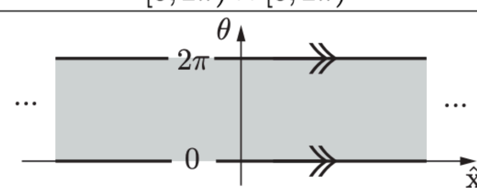
Some C-spaces can be expressed as the **Cartesian product** of two or more spaces of lower dimension; that is, points in such a C-space can be represented as the union of the representations of points in the lower-dimensional spaces. For example:

- The C-space of a rigid body in the plane can be written as $\mathbb{R}^2 \times S^1$, since the configuration can be represented as the concatenation of the coordinates (x, y) representing $\mathbb{R}^2$ and an angle θ representing S^1.

- The C-space of a PR robot arm can be written $\mathbb{R}^1 \times S^1$. (We will occasionally ignore joint limits, i.e., bounds on the travel of the joints, when expressing the topology of the C-space; with joint limits, the C-space is the Cartesian product of two closed intervals of the line.)

- The C-space of a 2R robot arm can be written $S^1 \times S^1 = T^2$, where T^n is the n-dimensional surface of a torus in an $(n+1)$-dimensional space. (See Table 2.2.) Note that $S^1 \times S^1 \times \cdots \times S^1$ (n copies of S^1) is equal to T^n, not S^n; for example, a sphere S^2 is not topologically equivalent to a torus T^2.
- The C-space of a planar rigid body (e.g., the chassis of a mobile robot) with a 2R robot arm can be written as $\mathbb{R}^2 \times S^1 \times T^2 = \mathbb{R}^2 \times T^3$.
- As we saw in Section 2.1 when we counted the degrees of freedom of a rigid body in three dimensions, the configuration of a rigid body can be described by a point in $\mathbb{R}^3$, plus a point on a two-dimensional sphere S^2, plus a point on a one-dimensional circle S^1, giving a total C-space of $\mathbb{R}^3 \times S^2 \times S^1$.

Configuration Space Representation:

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	 $\mathbb{E}^1 \times S^1$	 $\mathbb{R}^1 \times [0, 2\pi)$

Configuration and Velocity Constraints:

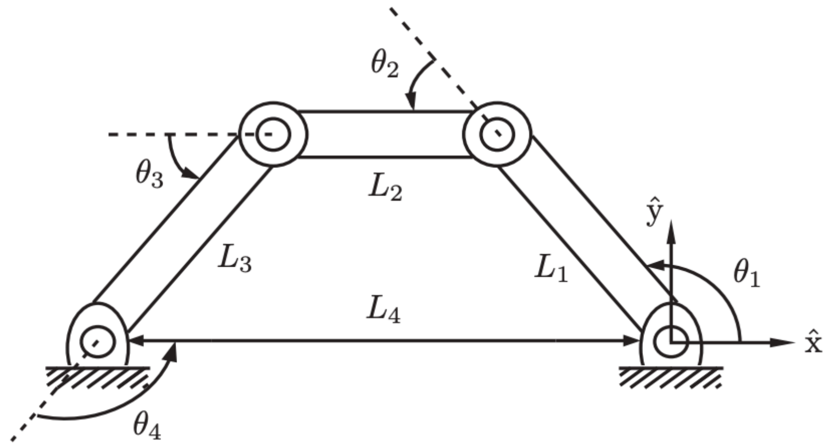


Figure 2.10: The four-bar linkage.

For robots containing one or more closed loops, usually an implicit representation is more easily obtained than an explicit parametrization. For example, consider the planar four-bar linkage of Figure 2.10, which has one degree of freedom. The fact that the four links always form a closed loop can be expressed by the following three equations:

$$\begin{aligned}
L_1 \cos\theta_1 + L_2 \cos(\theta_1+\theta_2) + \cdots + L_4 \cos(\theta_1+\cdots+\theta_4) &= 0, \\
L_1 \sin\theta_1 + L_2 \sin(\theta_1+\theta_2) + \cdots + L_4 \sin(\theta_1+\cdots+\theta_4) &= 0, \\
\theta_1+\theta_2+\theta_3+\theta_4 - 2\pi &= 0.
\end{aligned}$$

These equations are obtained by viewing the four-bar linkage as a serial chain with four revolute joints in which (i) the tip of link L_4 always coincides with the origin and (ii) the orientation of link L_4 is always horizontal.

These equations are sometimes referred to as **loop-closure equations**. For
constrained eq.
the four-bar linkage they are given by a set of three equations in four unknowns. The set of all solutions forms a one-dimensional curve in the four-dimensional joint space and constitutes the C-space.

Thus, for general robots containing one or more closed loops, the configuration space can be implicitly represented by the column vector $\theta = [\theta_1 \ \cdots \ \theta_n]^{\mathrm{T}} \in \mathbb{R}^n$ and loop-closure equations of the form

$$g(\theta) = \begin{bmatrix} g_1(\theta_1, \ldots, \theta_n) \\ \vdots \\ g_k(\theta_1, \ldots, \theta_n) \end{bmatrix} = 0, \tag{2.5}$$

a set of k independent equations, with $k \leq n$. Such constraints are known as **holonomic constraints**, ones that reduce the dimension of the C-space.[6] The C-space can be viewed as a surface of dimension $n - k$ (assuming that all constraints are independent) embedded in $\mathbb{R}^n$.

Suppose that a closed-chain robot with loop-closure equations $g(\theta) = 0$, $g : \mathbb{R}^n \to \mathbb{R}^k$, is in motion, following the time trajectory $\theta(t)$. Differentiating both sides of $g(\theta(t)) = 0$ with respect to t, we obtain

$$\frac{d}{dt} g(\theta(t)) \quad = \quad 0; \tag{2.6}$$

thus

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

Chain Rule

Each constraint equation $g_i(\theta)$ is a function of the joint angles $\theta_1, \theta_2, \ldots, \theta_n$. So the derivative with respect to time is:

$$\frac{d}{dt} g_i(\theta(t)) = \frac{\partial g_i}{\partial \theta_1}\dot{\theta}_1 + \frac{\partial g_i}{\partial \theta_2}\dot{\theta}_2 + \cdots + \frac{\partial g_i}{\partial \theta_n}\dot{\theta}_n$$

$$\dot{\theta}_n = \frac{d\theta_n}{dt}$$

This can be expressed as a matrix multiplying a column vector $[\dot{\theta}_1 \ \cdots \ \dot{\theta}_n]^{\mathrm{T}}$:

$$\begin{bmatrix} \frac{\partial g_1}{\partial \theta_1}(\theta) & \cdots & \frac{\partial g_1}{\partial \theta_n}(\theta) \\ \vdots & \ddots & \vdots \\ \frac{\partial g_k}{\partial \theta_1}(\theta) & \cdots & \frac{\partial g_k}{\partial \theta_n}(\theta) \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_n \end{bmatrix} \quad = \quad 0,$$

Task Space and Workspace:

Find the description of Task Space and Workspace from the reference book and explain them.

THE END