Introduction to Robotics

Manipulation and Programming

Unit 2: Kinematics

PATH PLANNING AND TRAJECTORY PLANNING

DR. ERIC CHOU

IEEE SENIOR MEMBER

Objectives

- What is path planning and what is trajectory planning
- Where are we going and how are we going there?
- What is parametric function?
- How to perform path planning: using parametric function to control the joint variables to send the end-effector to the target position.

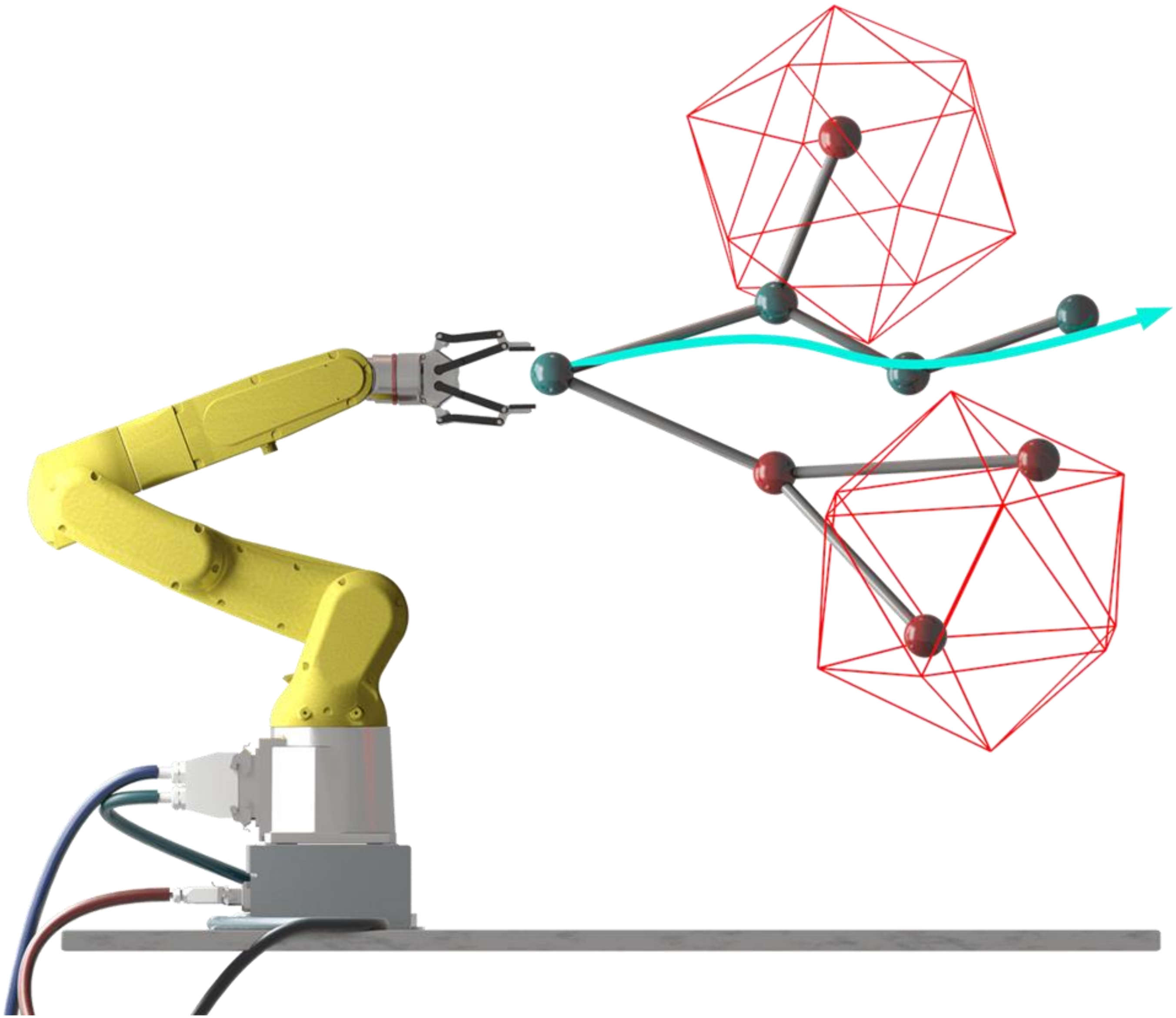
Path Planning

SECTION 1


WHERE
are we going?

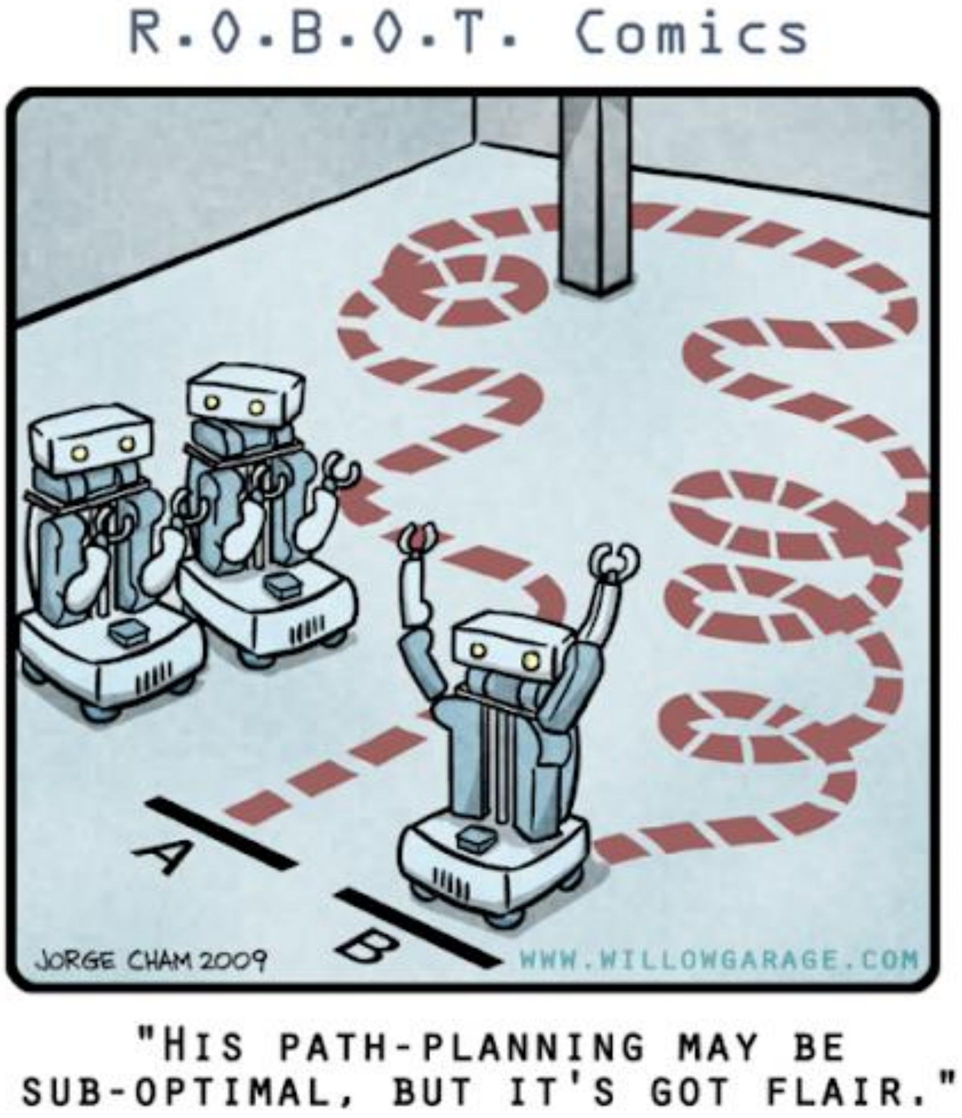
Path Planning

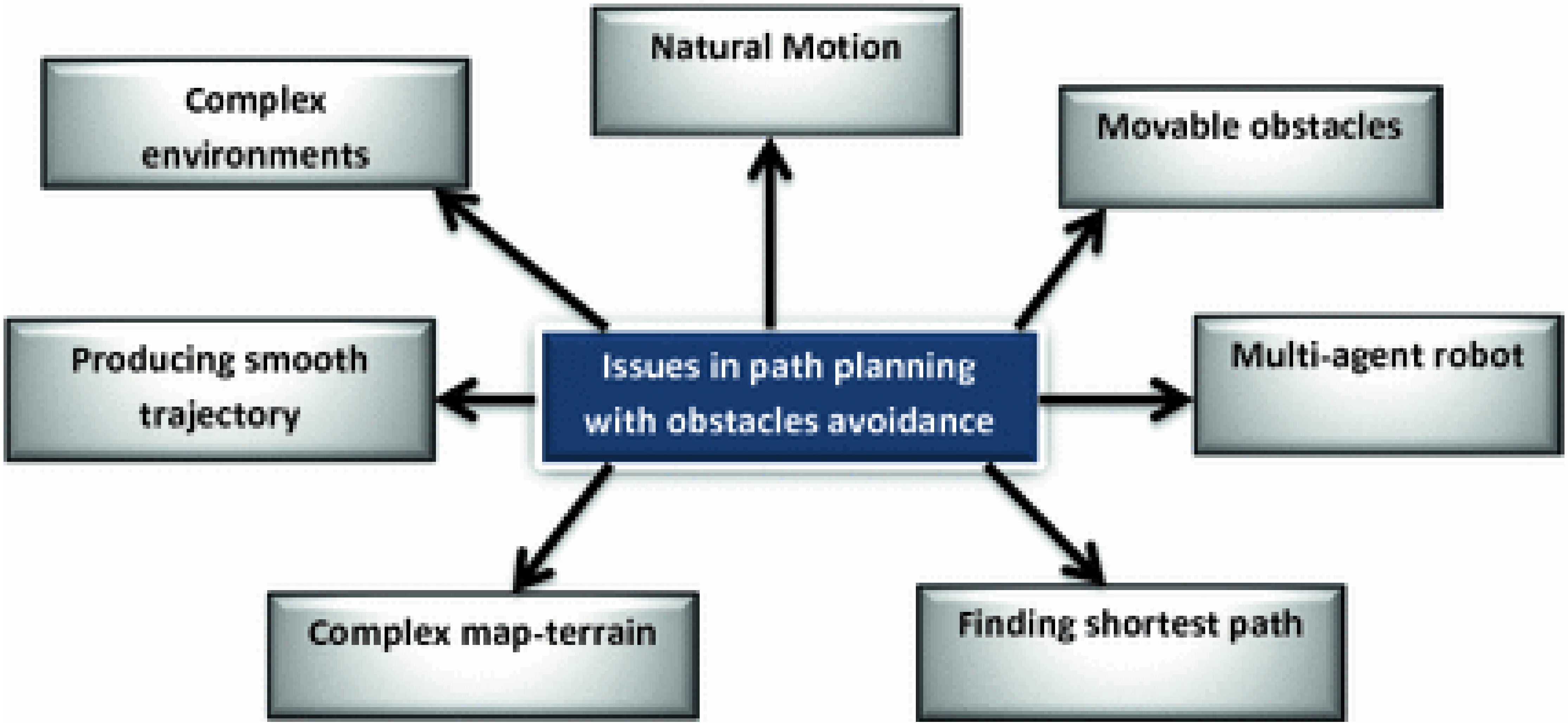
- Workspace movement planning
- Figure out the points in space through which the end-effector will pass.

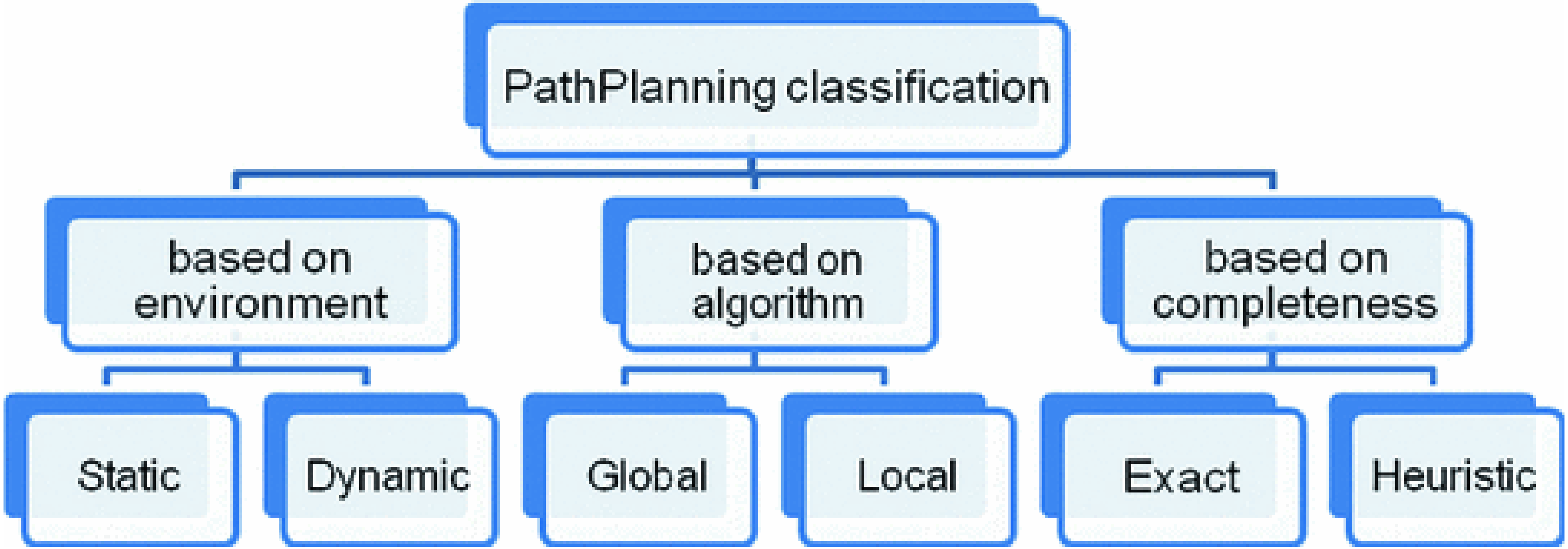


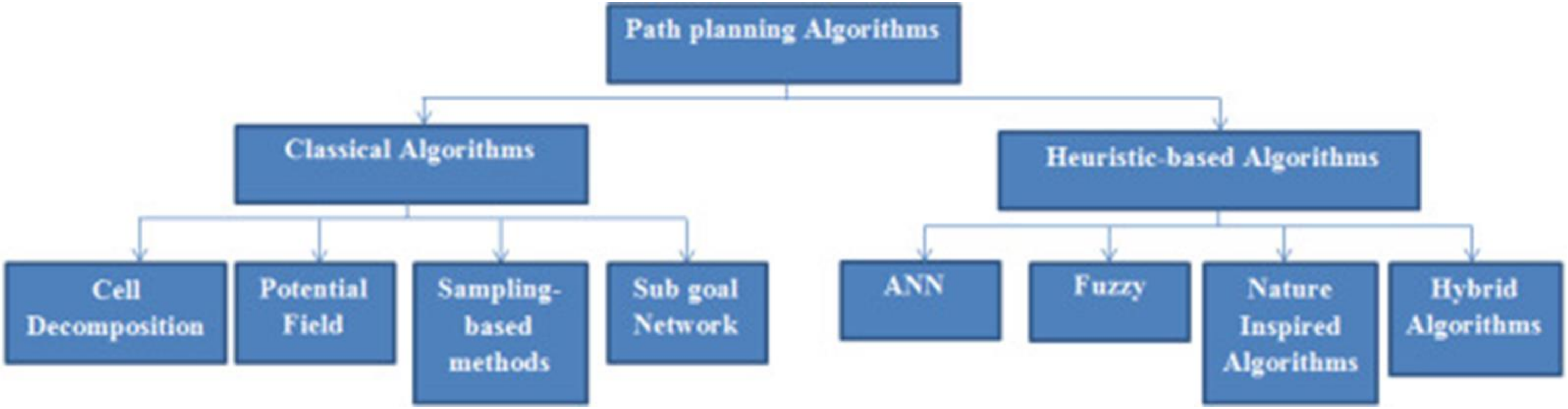
Path Planning

Movement Path Planning



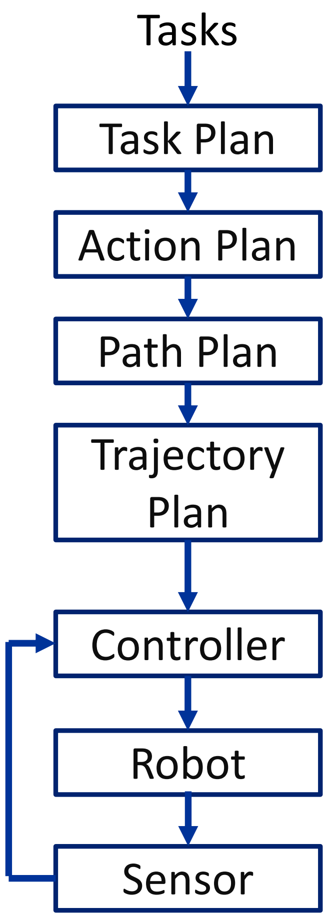

Natural Motion
Complex environments
Movable obstacles
Producing smooth trajectory
Issues in path planning with obstacles avoidance
Multi-agent robot
Complex map-terrain
Finding shortest path


PathPlanning classification
based on environment
based on algorithm
based on completeness
Static
Dynamic
Global
Local
Exact
Heuristic


Path planning Algorithms
Classical Algorithms
Heuristic-based Algorithms
Cell Decomposition
Potential Field
Sampling-based methods
Sub goal Network
ANN
Fuzzy
Nature Inspired Algorithms
Hybrid Algorithms

Path Planning

- Workspace Planning and Motion Planning
- Planning:
  1. Task plan
  2. Action Plan
  3. Path Plan
  4. Trajectory Plan
- Operation:
  1. Feedback System
  2. Sensing-Optimization-Action Loop

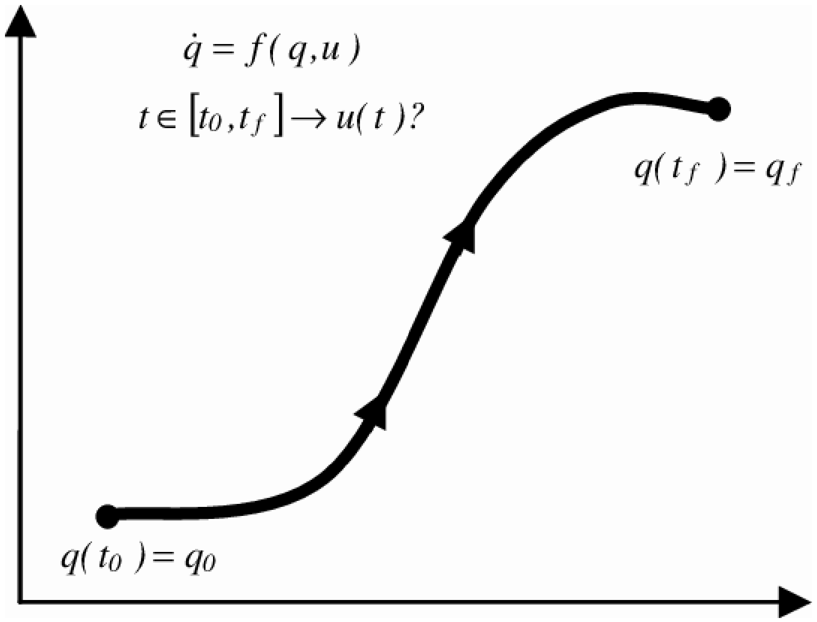


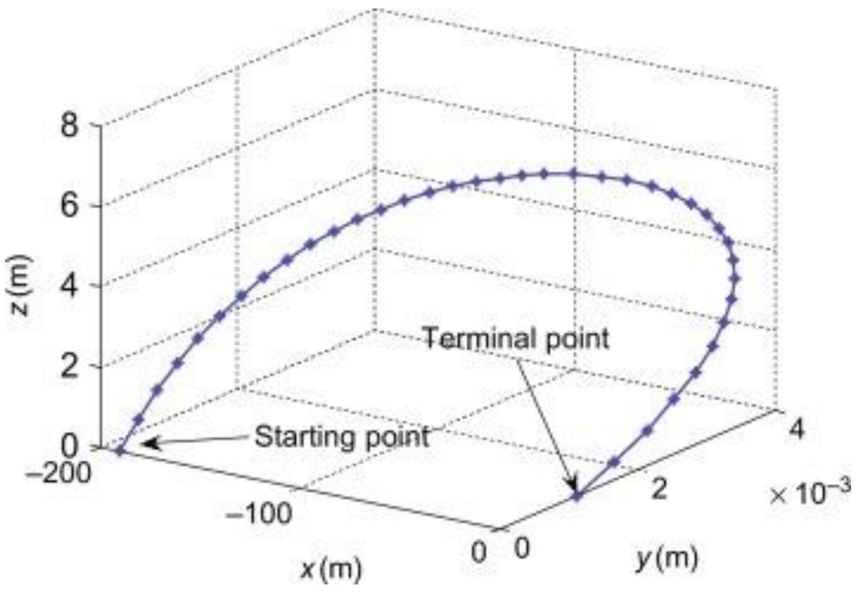
Trajectory Planning

SECTION 2

Trajectory Planning

- Trajectory generation: Figure out the velocity components of the end-effector motion along the path

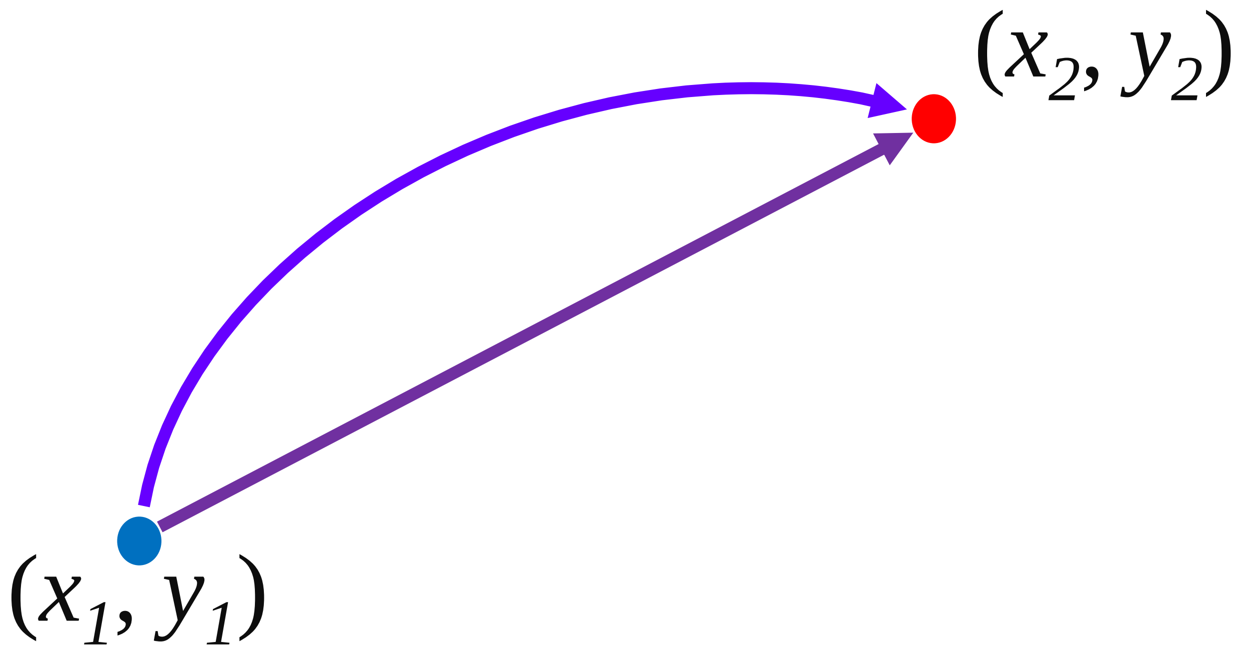




Parametric Equation

SECTION 3

Path Planning

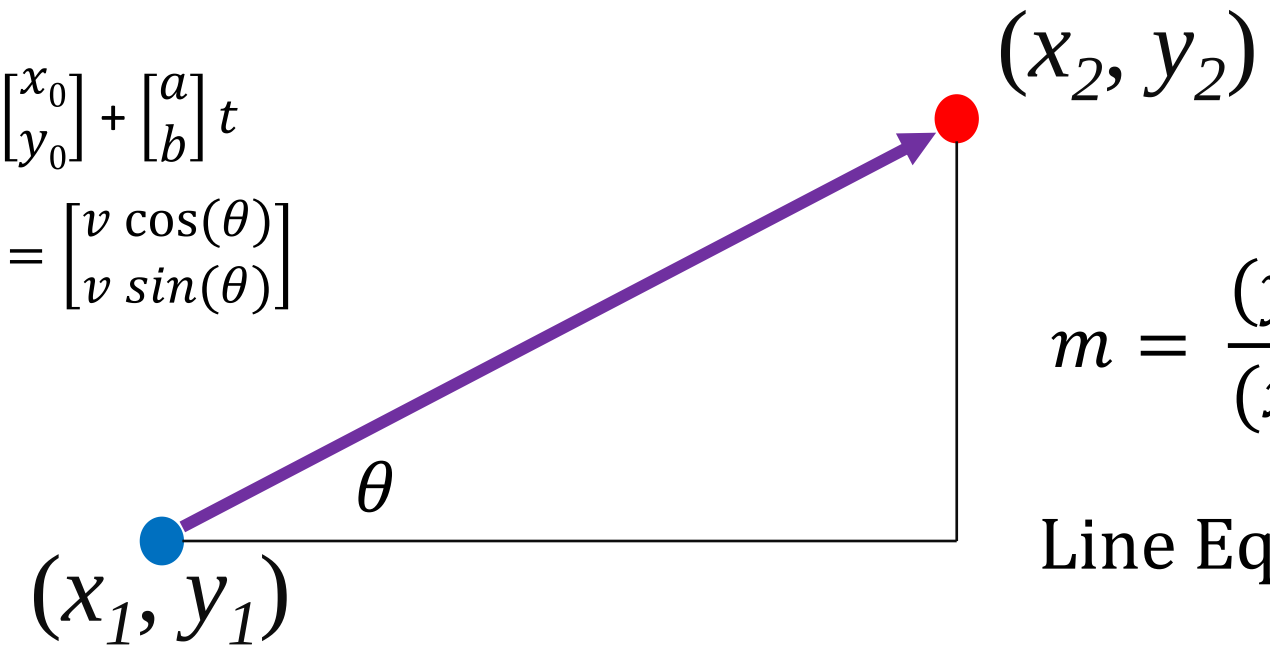


Parameter Equation

$$x(t) = x_0 + a\,t$$
$$y(t) = y_0 + b\,t$$

$$D(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} a \\ b \end{bmatrix} t$$

$$\mathrm{v}(t) = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} v\cos(\theta) \\ v\,sin(\theta) \end{bmatrix}$$

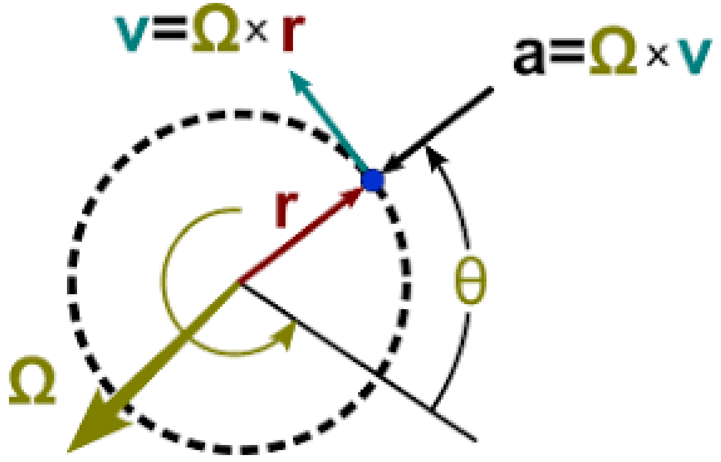


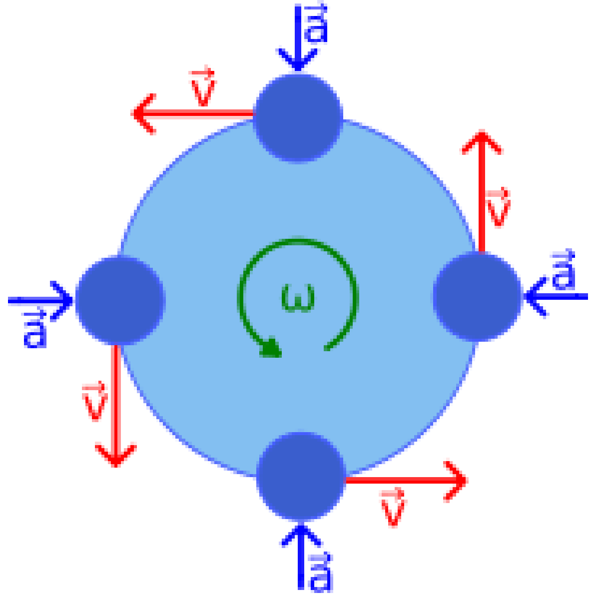
$$m = \frac{(y_2 - y_1)}{(x_2 - x_1)} = \tan(\theta)$$

Line Equation:

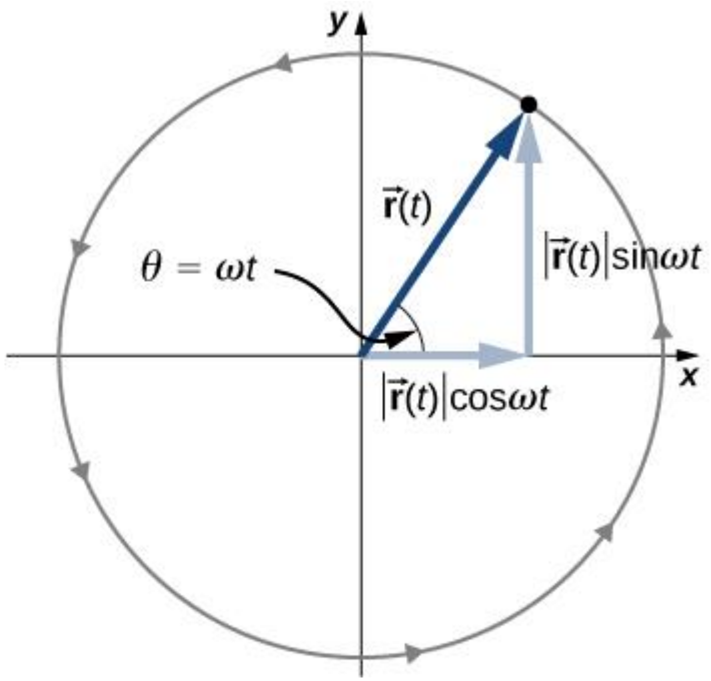
$$m = \frac{(y - y_0)}{(x - x_0)}$$

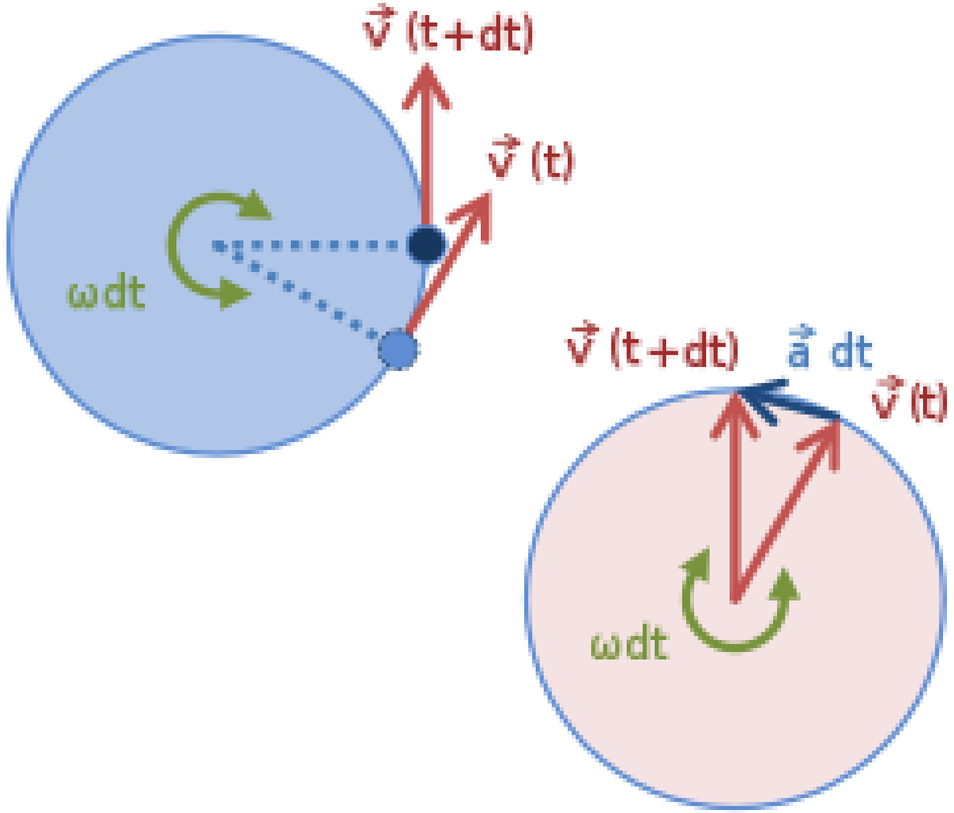
Unit Circle Motion



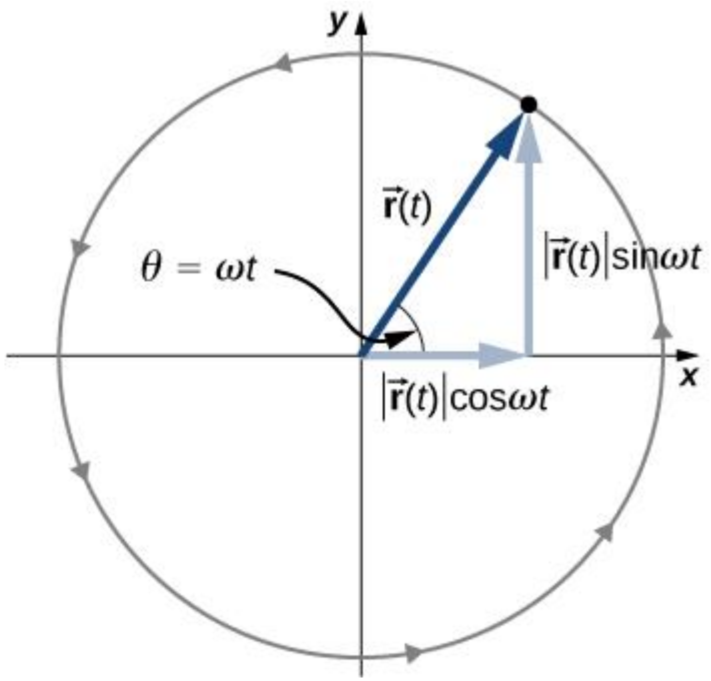


Unit Circle Motion





Unit Circle Motion



$$D(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} r\,cos(\omega t) \\ r\,sin(\omega t) \end{bmatrix}$$

$$v(t) = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \mathrm{r} \begin{bmatrix} -\omega\,sin(\omega t) \\ \omega\,cos(\omega t) \end{bmatrix}$$

Parametric Function

- Workspace variables can be linear or angular $q_i(t)$, for all i , q can be $\theta, or\ d$
- Describe the workspace variables as functions of joint variables and time)

$$X = \begin{bmatrix} x \\ y \\ z \\ \phi \\ \theta \\ \varphi \end{bmatrix} = h(\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{bmatrix})_{6\times 1} = \begin{bmatrix} h_1(q_1, q_2, \cdots, q_6) \\ h_2(q_1, q_2, \cdots, q_6) \\ h_3(q_1, q_2, \cdots, q_6) \\ h_4(q_1, q_2, \cdots, q_6) \\ h_5(q_1, q_2, \cdots, q_6) \\ h_6(q_1, q_2, \cdots, q_6) \end{bmatrix}_{6\times 1}$$

Jacobian Matrix

$$\begin{bmatrix} V \\ \Omega \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = J \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix}$$

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{bmatrix} \xrightarrow{\dot{X} = J(q)\dot{q}} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{bmatrix} \xleftarrow[\dot{q} = J^{-1}(q)\dot{X}]{} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

Joint Space

Task Space

Finding the position variable q (θ or d)

Inverse of Jacobian Matrix

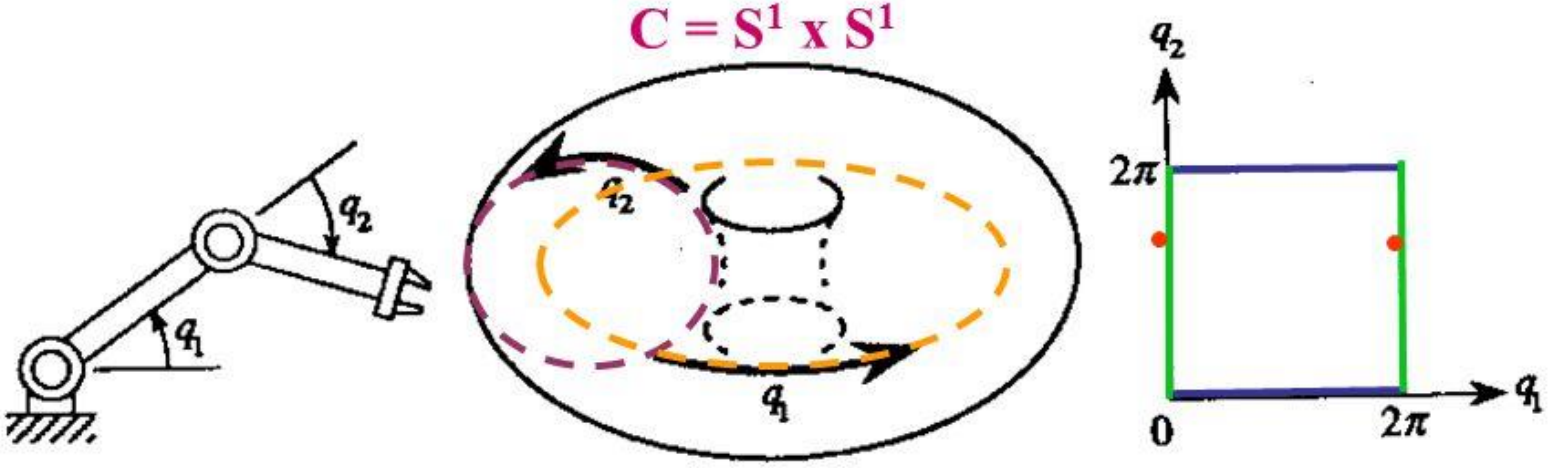
Finding the control functions described in time and joint variables. Given the positions (x, y, z) and angles (ϕ, θ, φ), to find the joint control function as a parametric function of time, or the velocity functions related to these joint functions.

Configuration Space

SECTION 4

Configuration Space of a Robot

- Space of all its possible configurations
- But the topology of this space is usually not that of a Cartesian space



Configuration **space**

- Robot **configuration** is described by a vector of **generalised** joint coordinates
- Each coordinate can be:
  - ➥ an angle, for a rotational (**revolute**) joint
  - ➥ a length, for a sliding (**prismatic**) joint

Joint configuration **Joint coordinate** **Number of joints** **Space of all possible configurations**

$$\boldsymbol{q} = \{q_j, j \in [1 \cdots N]\} \in \mathcal{C}$$

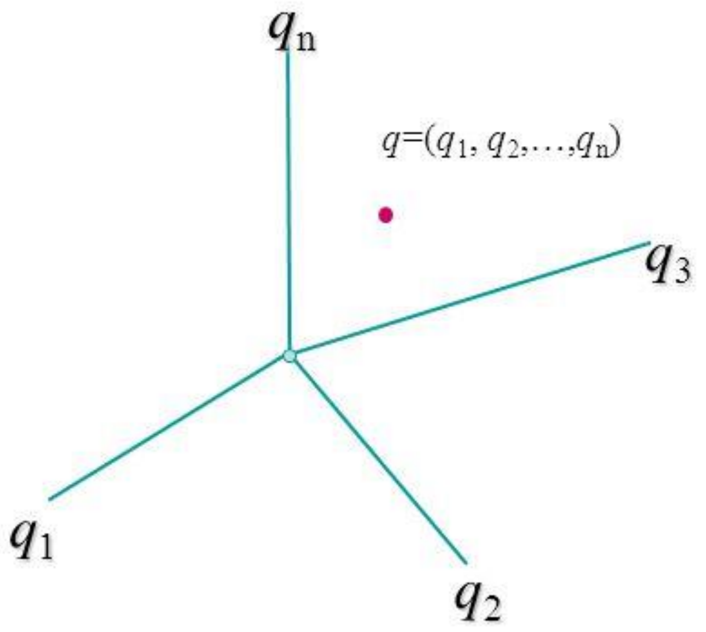
Configuration space $\mathcal{C} \subset \mathbb{R}^N$

Definitions

- **Configuration:**
  - Specification of all the variables that define the system completely
  - Example: Configuration of a $\boldsymbol{n}$ DOF robot is $q = (q_0, q_1, \dots q_{n-1})$
- **Configuration space (C-space):**
  - Set of all configurations
- **Free configuration:**
  - A configuration q that does not collide with obstacles
- **Free space (F) :**
  - Set of all free configurations
  - It is a subset of C

Configuration space

- The **configuration space** C is the set of all possible configurations.
  - A configuration is a point in C.
  - Similar to a
    - State space
    - Parameter space
- The **workspace** is all points reachable by the robot (or sometimes just the end effector)
- C can be very high dimensional while the workspace is just 2D or 3D



Path v/s Trajectory

- **Path:**
  - A sequence of robot configurations in a particular order without regard to the timing of these configurations.
- **Trajectory:**
  - It concerned about when each part of the path must be attained, thus specifying timing.

Path Planning

- **Problem statement:**
  - Compute a collision-free path for a rigid or articulated moving object among static obstacles.
- **Input**
  - Geometry of a moving object (a robot, a digital actor, or a molecule) and obstacles
  - How does the robot move?
  - Kinematics of the robot (degrees of freedom)
  - Initial and goal robot configurations (positions & orientations)
- **Output**
  - Continuous sequence of collision-free robot configurations connecting the initial and goal configurations

Trajectory Planning

- **Problem statement**
  - Turn a specified Cartesian-space trajectory of Pe into appropriate joint position reference values
- **Input**
  - Cartesian space path
  - Path constraints including velocity and acceleration limits and singularity analysis.
- **Output**
  - a series of joint position/velocity reference values to send to the controller

Trajectory Planning