Robot Kinematics

Intelligent Robotics
2014/15

Chris Burbridge

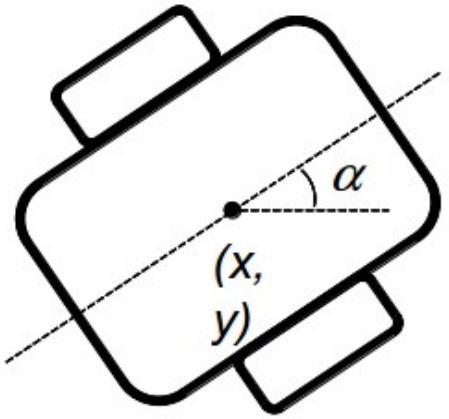
UNIVERSITY OF
BIRMINGHAM

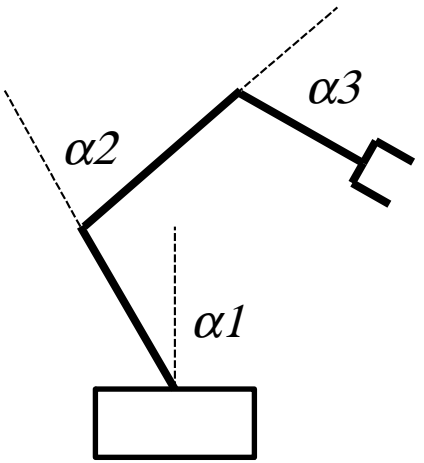
Outline

- Kinematics
  - Manipulators
  - Differential drive robots
- PRM Path planning

Robot Configuration

- A robot configuration c is uniquely described by a n-dimensional real vector.





- Example: The configuration of a differential drive vehicle consists of its position and orientation c = {x, y, a}.
- Example: The configuration of a planar robotic manipulator consists of a set of joint angles c = {a1, a2, a3}.

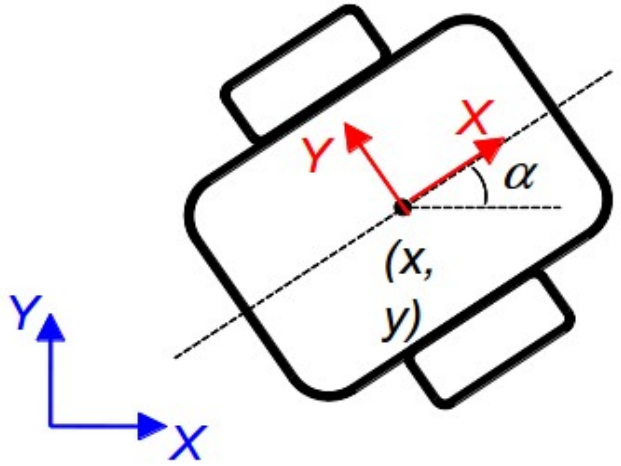
Robot Configuration Space

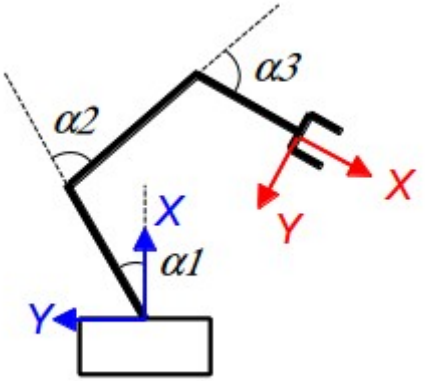
- The robot configuration space C is a n-dimensional Euclidean space - a set of all possible configurations c.

- *Example:* The configuration space C of a differential drive vehicle consists of all $c = \{x, y, \alpha\}$ such that: $a < x < b$, a $< y < b$, $-\pi < \alpha 3 < \pi$.

- *Example:* The configuration space C of a planar robotic manipulator consists of all $c = \{\alpha 1, \alpha 2, \alpha 3\}$ such that: $-\pi < \alpha 1 < \pi$, $-\pi < \alpha 2 < \pi$, $-\pi < \alpha 3 < \pi$.

Forward and Inverse Kinematics

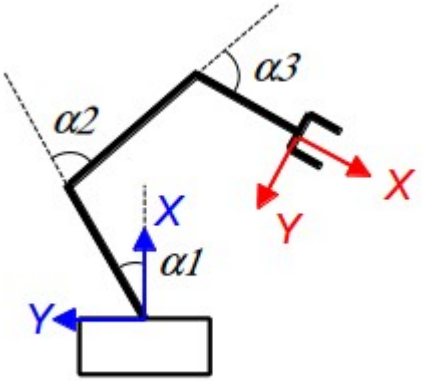




- Task space parameters (robot frame M): vehicle pose, end-effector frame.
- Forward kinematics: given robot configuration c, find robot frame M.
- Inverse kinematics: given robot frame M, find robot configuration c.

Forward Kinematics of a Manipulator

- A chain of links of length l_i connected by joints with angles α_i

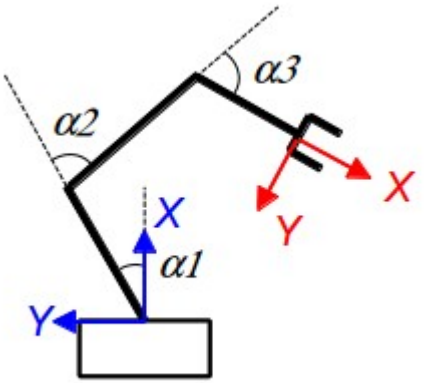


- We can solve the problem sequentially as a product of transformations
  - $M_i(\alpha_i)=\begin{bmatrix} R(\alpha_i) & R(\alpha_i)\,p_i \\ 0 & 1 \end{bmatrix} \quad p_i=\begin{bmatrix} l_i \\ 0 \end{bmatrix}$
- The overall transformation is:

$$M = M_1(\alpha_1).\,M_2(\alpha_2).\,M_3(\alpha_3)$$

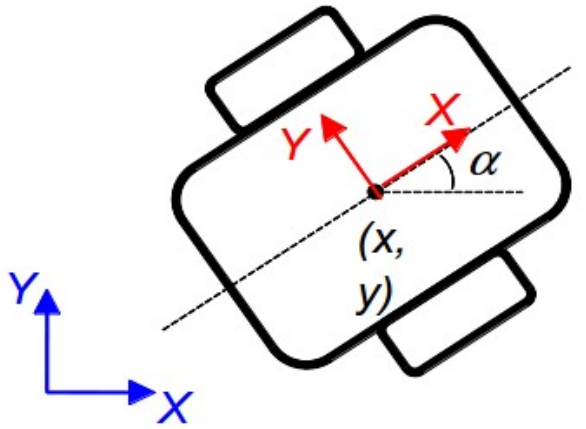
Inverse Kinematics of a Manipulator

- Some end effector poses have multiple solution configurations
- Some poses have no solutions
- IK-Fast: Formulate the problem mathematically and use a symbolic computation engine to auto-generate code.
- Jacobian based gradient ascent:

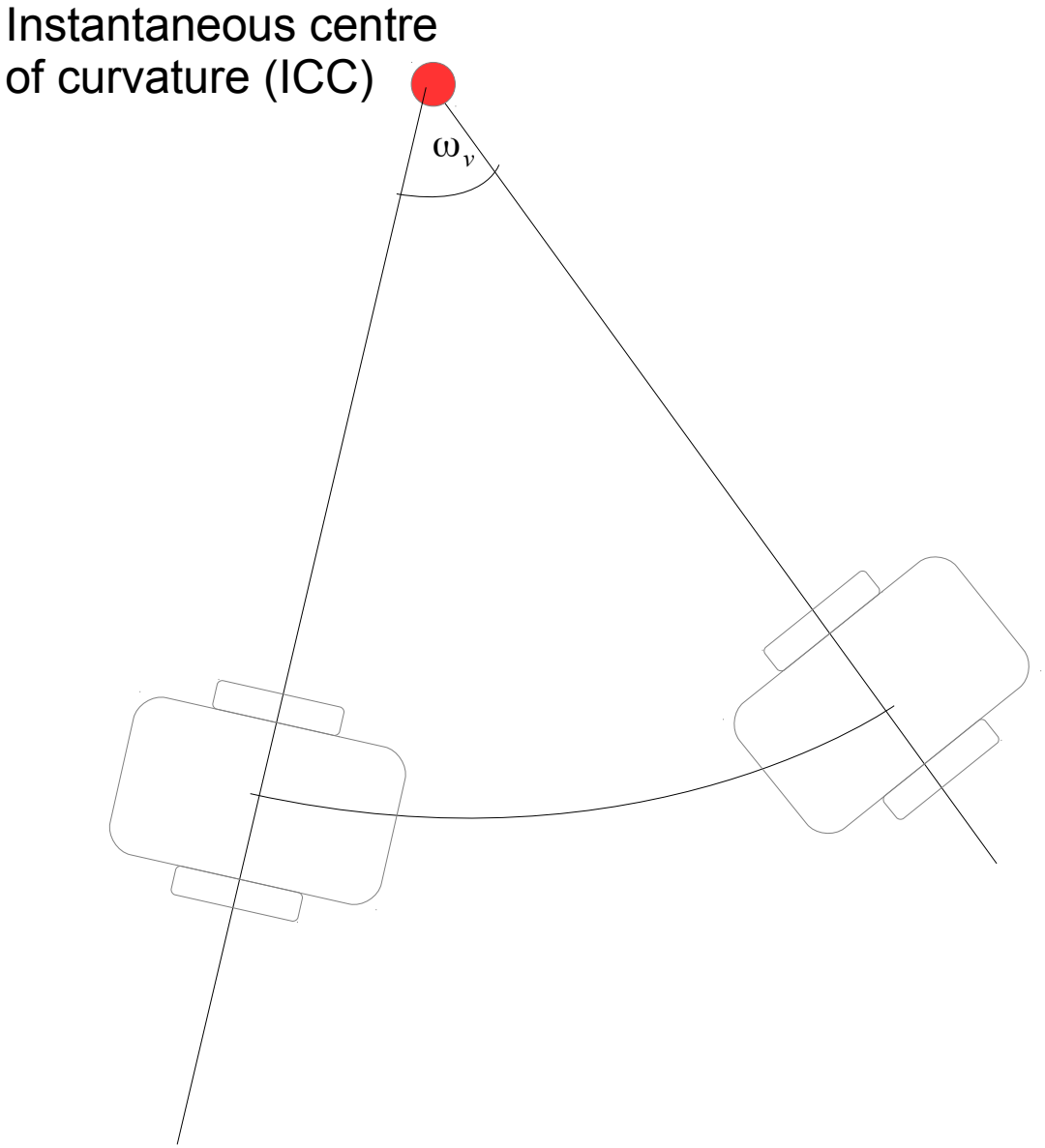


Differential Drive

- How many wheeled robots move
- Two wheels, either side of the robot driven independently
- Steering is achieved by driving the wheels at different speeds
- Two degrees of freedom -> 2 controllable values: (V_l, V_r)



Differential Drive



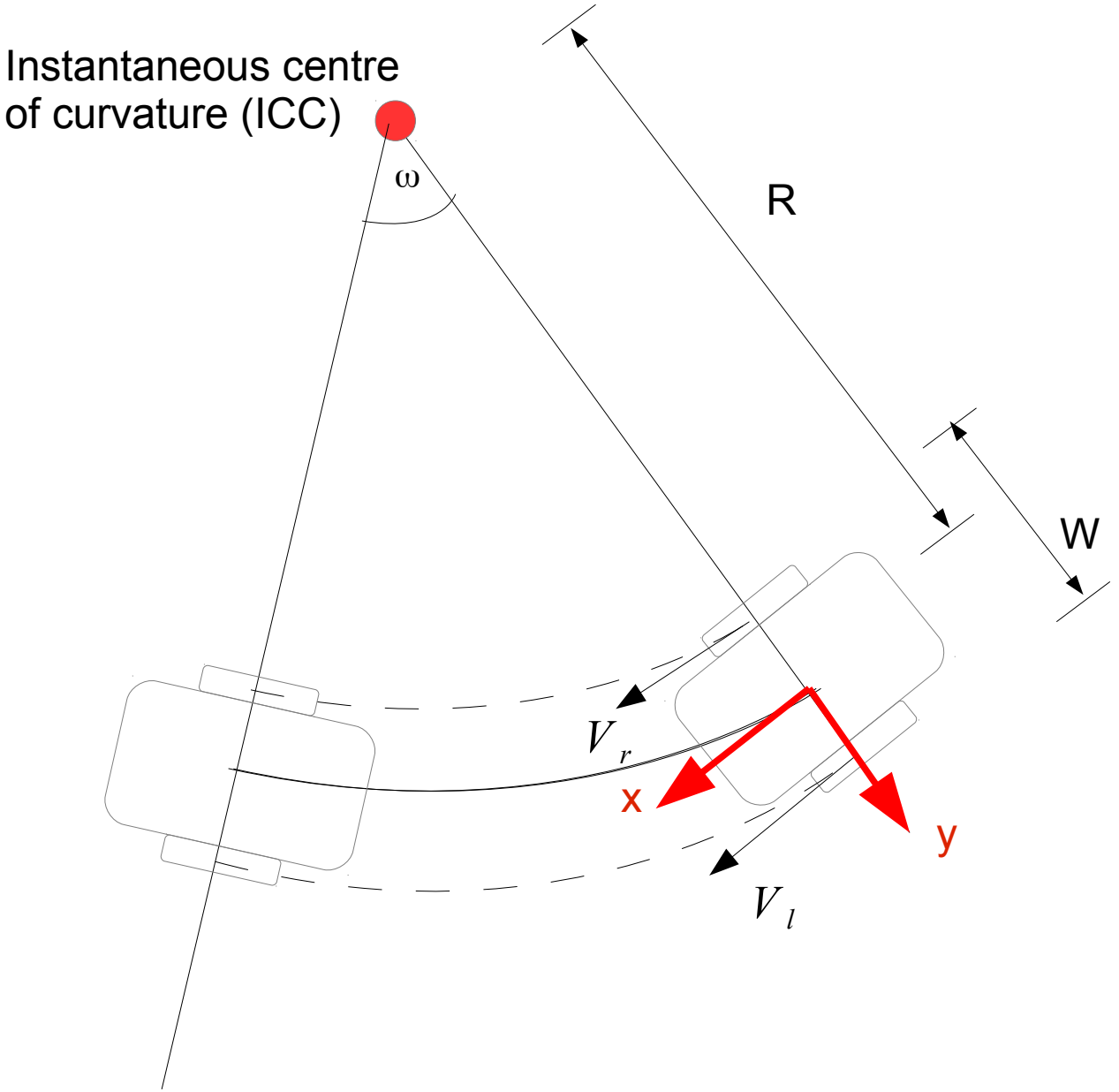
- The robot rotates about ICC
- Knowing the speed of each wheel, we need to calculate linear and angular velocity (v, ω)
- Odometry / forward kinematics:

$$x(t) = \int_0^t v \cos[\theta(t)] . dt$$

$$y(t) = \int_0^t v \sin[\theta(t)] . dt$$

$$\theta(t) = \int_0^t \omega(t) . dt$$

Differential Drive



Angular Velocity

$$\frac{d\theta}{dt} = \frac{V}{R}$$

So,

$$\omega\left(R + \frac{W}{2}\right) = V_l$$

$$\omega\left(R - \frac{W}{2}\right) = V_r$$

Therefore

$$\omega = \frac{V_r + V_l}{W}$$

$$R = \frac{W}{2} \cdot \frac{V_l + V_r}{V_r - V_l}$$

$$v = \frac{V_r + V_l}{2}$$

Forward Kinematics of Differential Drive

- We know R, and the robot's current pose (x, y, θ), so we can find the ICC

$$ICC = [x - R.sin(\theta), y + R.cos(\theta)]$$

- Given how long (Δt) the robot moved with constant wheel velocities, calculating the new robot pose (x', y', θ') is a rotation & translation about the known ICC:

$$\omega = \frac{V_r + V_l}{W}$$

$$R = \frac{W}{2} \cdot \frac{V_l + V_r}{V_r - V_l}$$

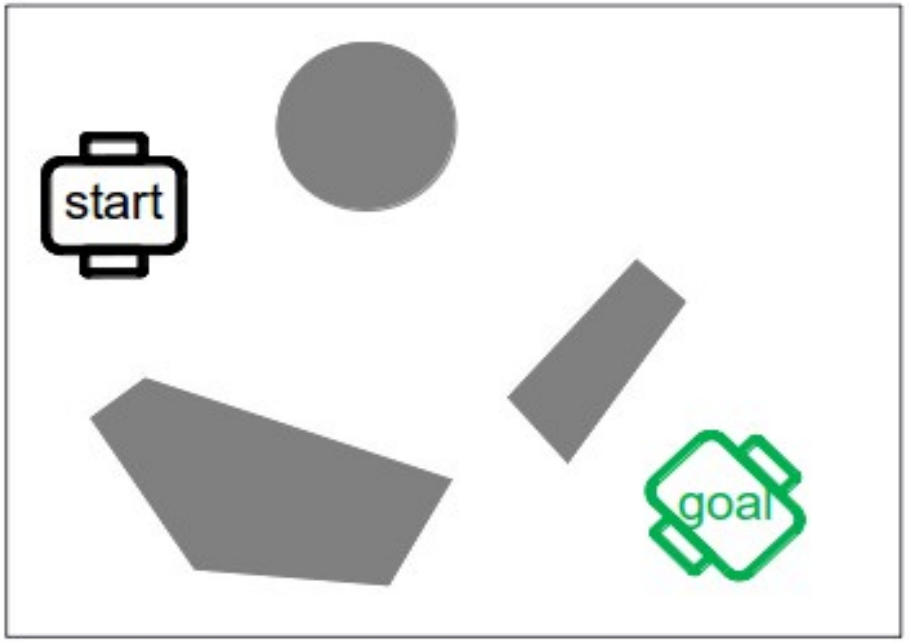
$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos(\omega \Delta t) & -\sin(\omega \Delta t) & 0 \\ \sin(\omega \Delta t) & \cos(\omega \Delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - ICC_x \\ y - ICC_y \\ \theta \end{bmatrix} + \begin{bmatrix} ICC_x \\ ICC_y \\ \omega \Delta t \end{bmatrix}$$

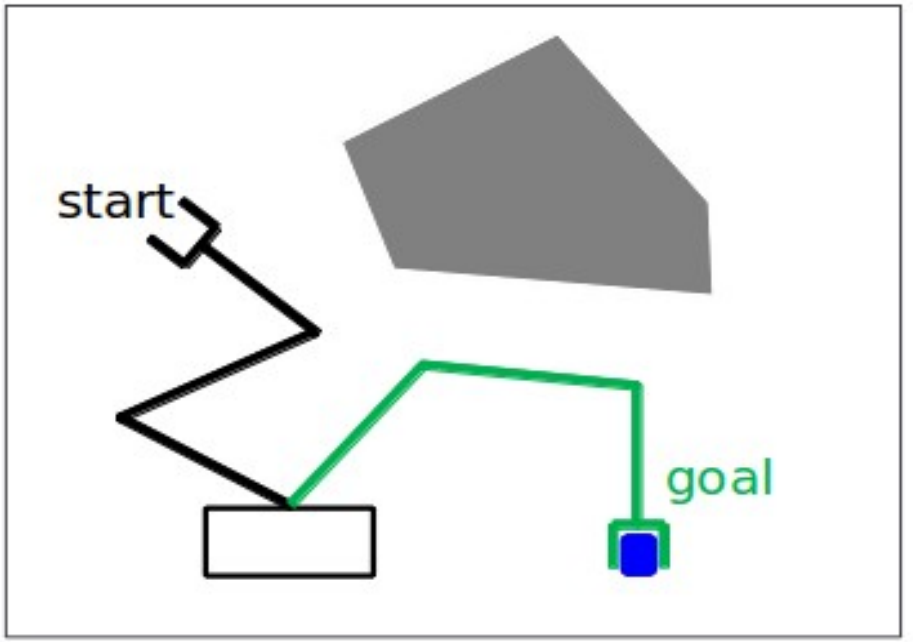
Inverse kinematics of Differential Drive

- How do we control the robot to reach a certain (x, y, θ)
- The robot motion is constrained – the robot cannot move sideways. It is *non-holonomic.*
- Can't just specify any pose and compute a linear and angular velocity to get there.
- so....

Motion Planning

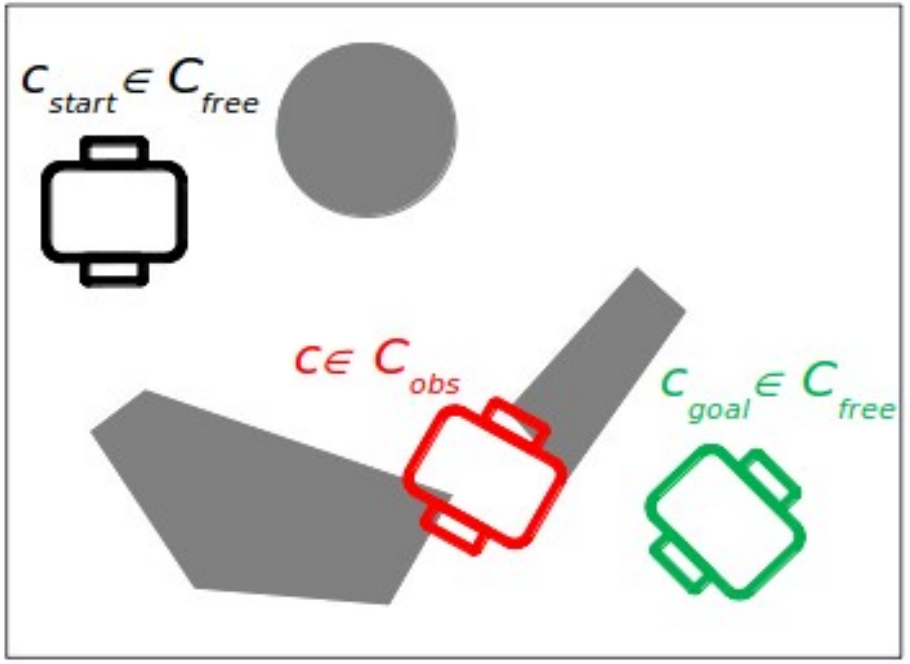
Motion Planning

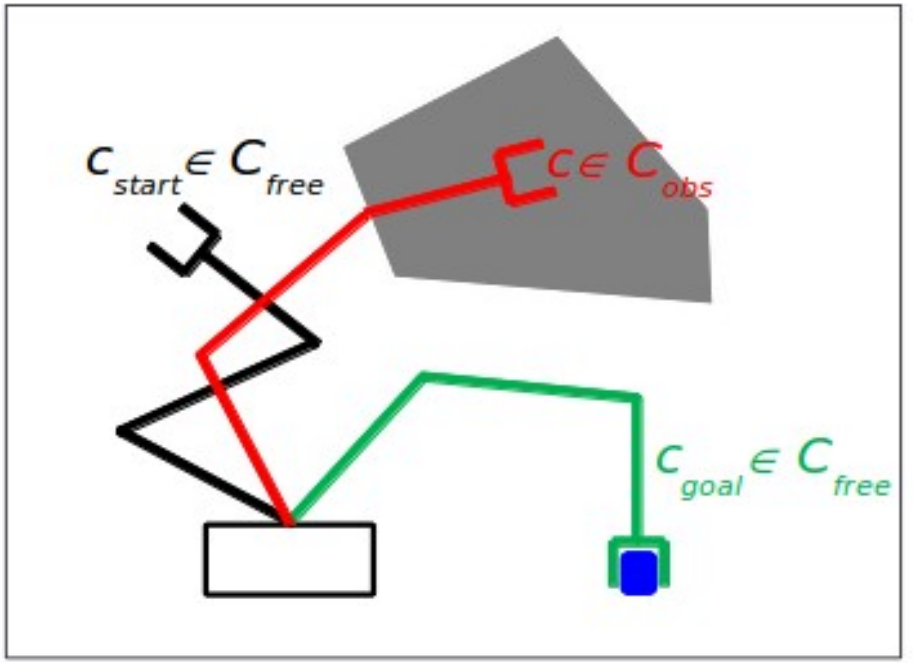




- Find a route from start to goal through the free space.
- Trajectory planning: a time indexed sequence of positions and velocities in the robot configuration space.
- Path planning: a sequence of positions in the robot configuration space.

Path Planning



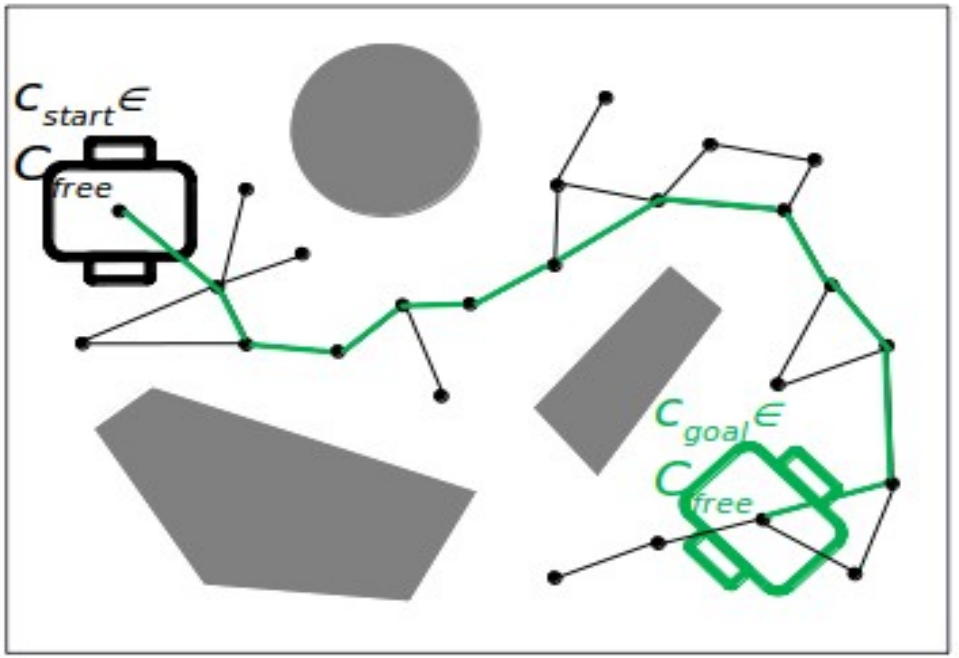


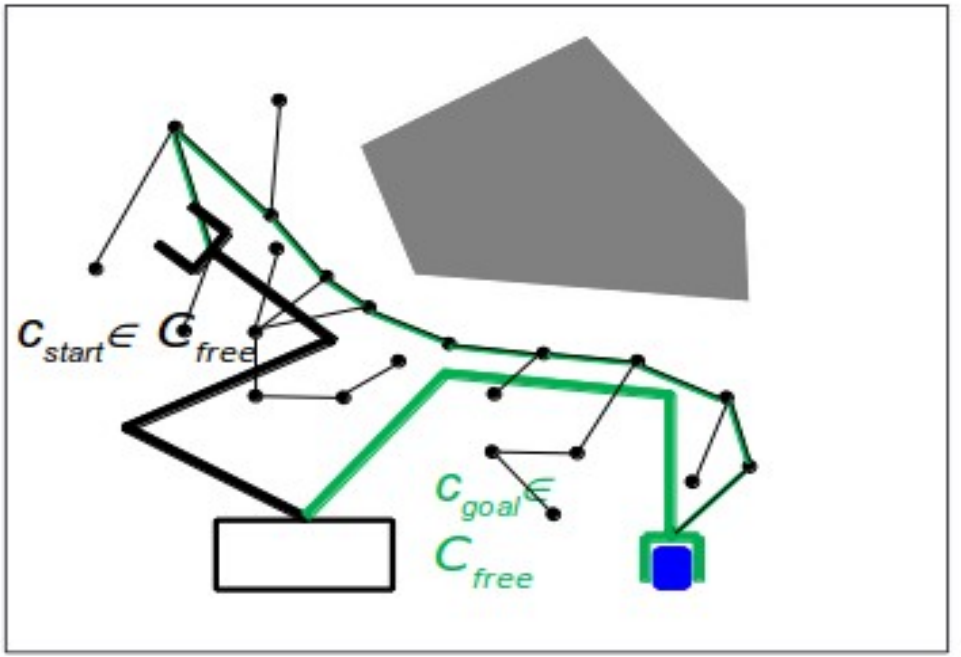
- Robot configuration space $C = C_{free} \cup C_{obs}$.
- Vehicle $C = \{x, y, \alpha\}$, manipulator $C = \{\alpha_1, \alpha_2, \alpha_3\}$.
- Find a continuous path in the robot configuration space $\tau:[0, 1] \rightarrow C_{free}$, such that $\tau(0) = cstart$ and $\tau(1) = c_{goal}$.
- The path might not exist.

Sampling based path planning

- Sample configuration space *C* and construct a roadmap which approximates a collision free path τ:[*0, 1*] $\rightarrow C_{free}$.
- A roadmap is an undirected graph G(V, E) where:
  - Vertices V are sampled configurations from the free space C_{free}.
  - Edges E connecting vertices V are collision-free paths determined by a local planner.
- In general the graph G(V, E) can be constructed in two ways:
  - Multi-query planners construct a complete graph G(V, E) which approximates the connectivity of *Cfree*.
  - Single-query planners construct a graph G(V, E) online, for each new planning query.

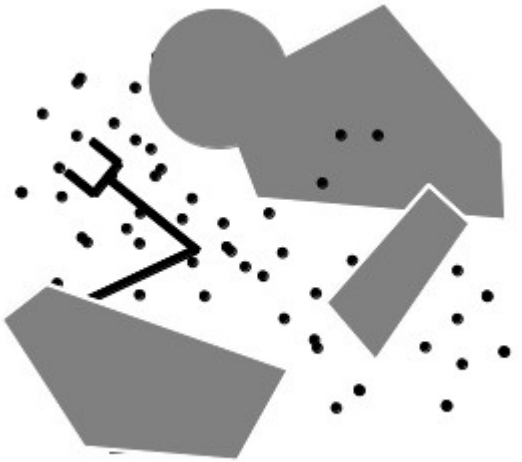
Probabilistic Road Map (PRM)





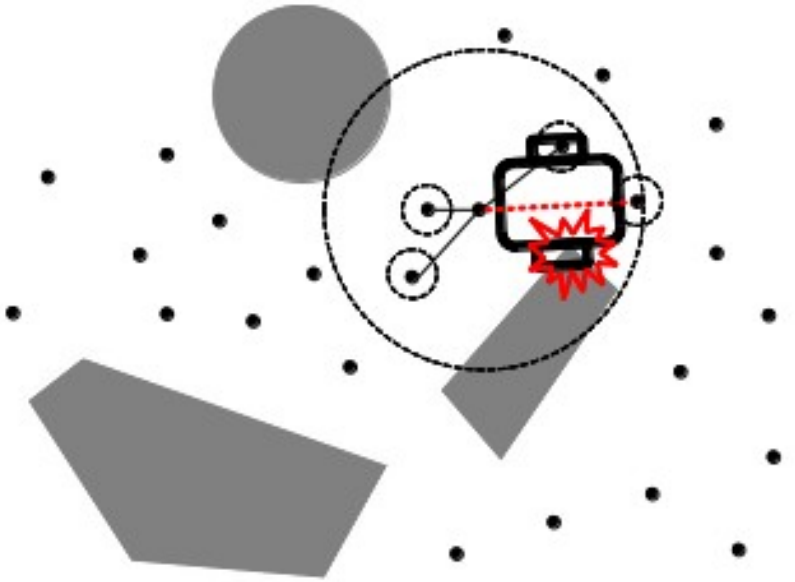
- Create vertices V: generate a (large) number of random configurations $c \in C_{free}$ (for each query also insert c_{start} and c_{goal}).
- Create edges E: attempt to connect vertices V with local collision-free paths, retain only connected vertices.
- Use A* graph search to find the (shortest) route from c_{start} to c_{goal}.

Sampling techniques



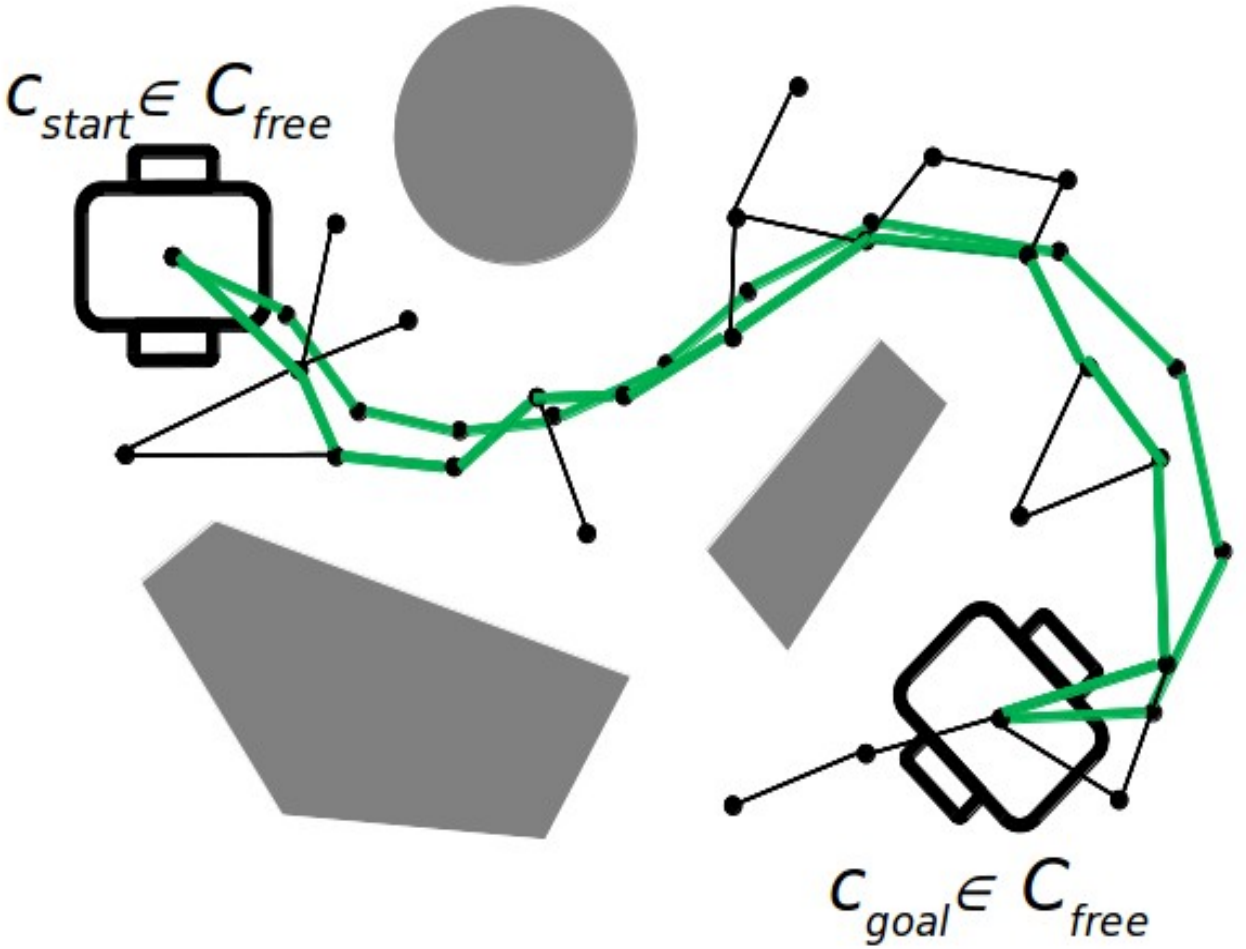
- Sample directly in the configuration space C, reject c if $c \notin C_{free}$.
- Sample on a grid in the robot task space.
- Generate samples near Voronoi regions of obstacles.

Local path planning



- Attempt to connect vertices only within some maximum distance.
- Linearly interpolate between configurations.
- Reject edges which collide with obstacles.

Path profiling and optimisation



Summary

- Forward and inverse kinematics for manipulation is mostly just the application of rigid body transformations.
- Differential drive robots introduce constrained motion (non-holonomic).
- Probabilistic Road Map planning is one useful way of planning robot paths.
- Highly recommended reading:
  - Murphy R,R,, Introduction to AI Robotics, MIT Press, ISBN 0-262-13383-0
  - Kortenkamp D., Bonasso R.P., Murphy R., Artificial Intelligence and Robotics, MIT Press, ISBN 0-262-61137-6
  - Dudek and Jenkin, Computational Principles of Mobile Robotics
  - Murray, R. M., Li, Z. & Sastry, S. S. A mathematical introduction to robotic manipulation. (CRC press, 1994).
  - Russell, S. & Norvig, P. Artificial Intelligence: A Modern Approach (3rd Edition). (Prentice Hall, 2009).