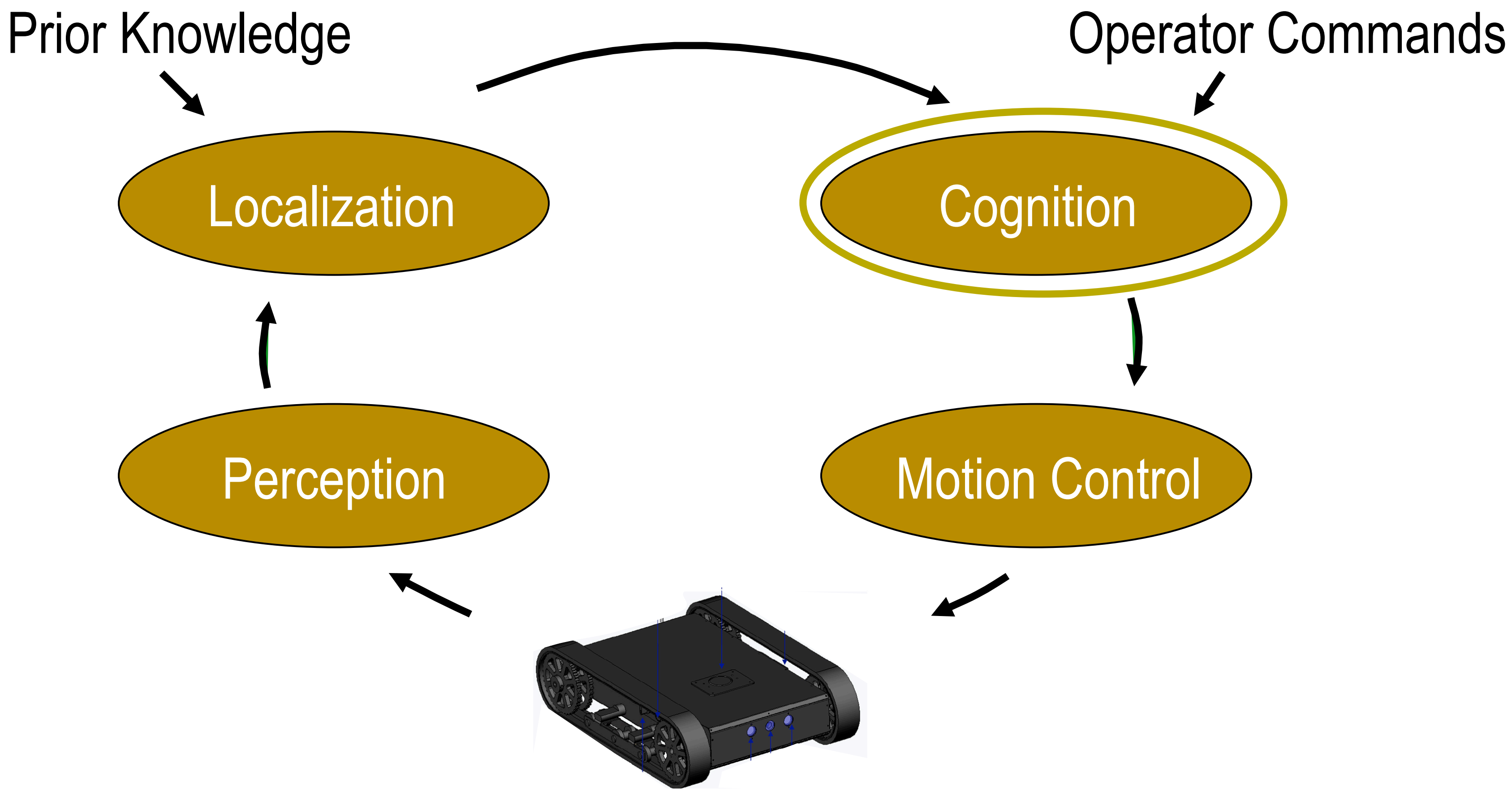
E190Q – Lecture 14
Autonomous Robot Navigation

Instructor: Chris Clark

Semester: Spring 2014

Figures courtesy of Probabilistic Robotics (Thrun et. Al.)

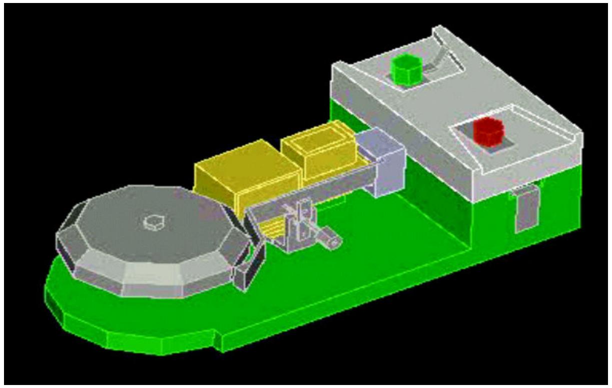
Control Structures
Planning Based Control



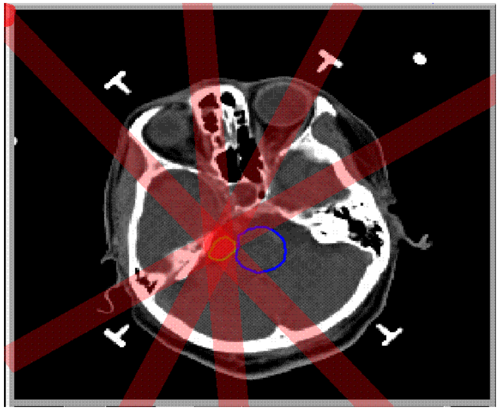
Introduction to Motion Planning

1. MP Overview
2. The Configuration Space
3. General Approach to MP
4. Metrics
5. PRMs
6. Single Query PRMs

MP Overview

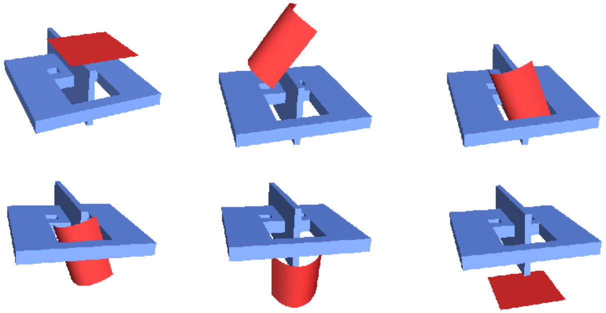


Assembly Planning, Latombe



Cross-Firing of a Tumor, Latombe

Tomb Raider 3 (Eidos Interactive)



Deformable Objects, Kavraki

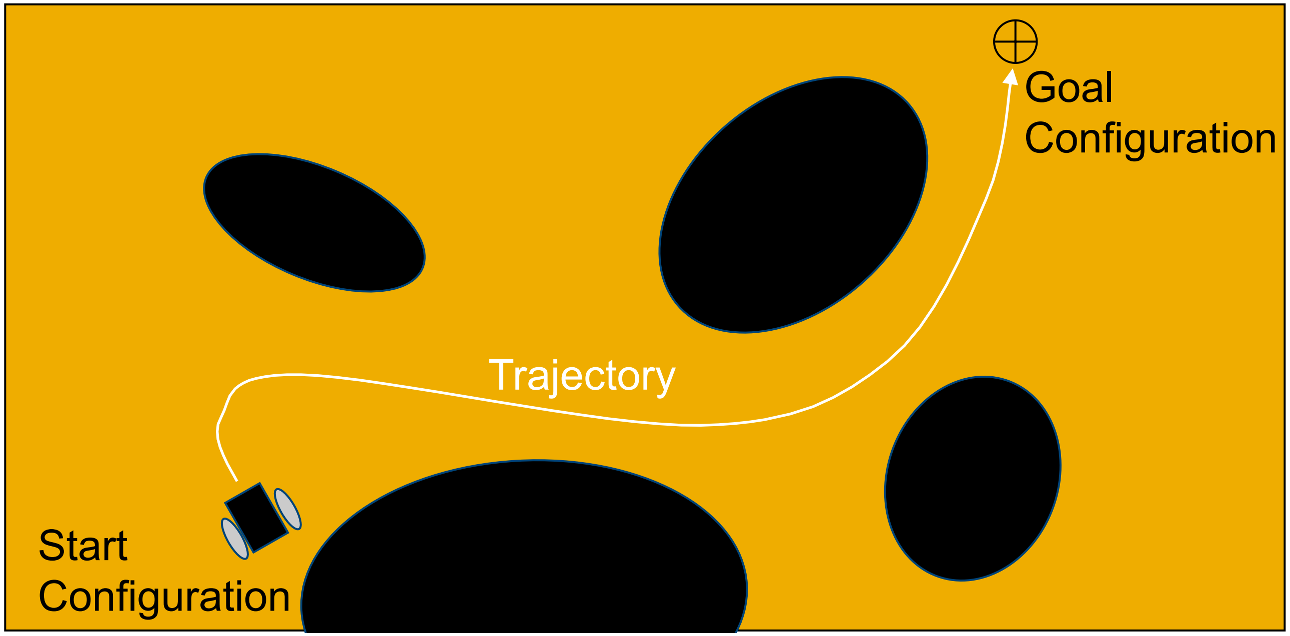
MP Overview

- Goal of robot motion planning:

  To construct a collision-free path from some initial configuration to some goal configuration for a robot within a workspace containing obstacles.

MP Overview

- Example:



MP Overview

- Inputs
  - Geometry of robots and obstacles
  - Kinematics/Dynamics of robots
  - Start and Goal configurations
- Outputs
  - Continuous sequence of configurations connecting the start and goal configurations

MP Overview

- Extensions
  - Moving obstacles
  - Multiple robots
  - Movable objects
  - Assembly planning
  - Goal is to acquire information by sensing
  - Nonholonomic constraints
  - Dynamic constraints
  - Stability constraints
  - Uncertainty in model, control and sensing
  - Exploiting task mechanics (under-actuated systems)
  - Physical models and deformable objects
  - Integration with higher-level planning

Introduction to Motion Planning

1. MP Overview
2. The Configuration Space
3. General Approach to MP
4. Metrics
5. PRMs
6. Single Query PRMs

The Configuration Space

- To facilitate motion planning, the **configuration space** was defined as a tool that can be used with planning algorithms.

(Latombe 1991)

The Configuration Space

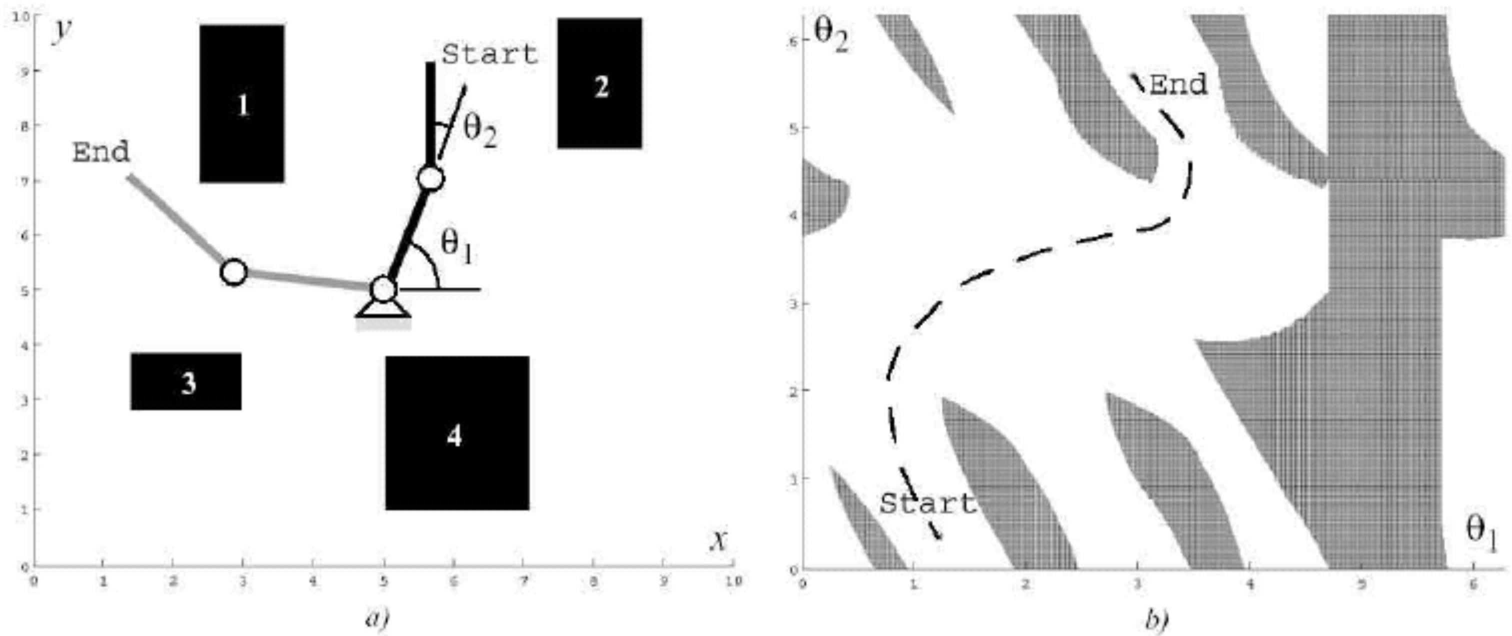
- A configuration q will completely define the state of a robot (e.g. mobile robot x, y, θ)
- The configuration space C, is the space of all possible configurations of the robot.
- The free space $F \subseteq C$, is the portion of the free space which is collision-free.

The Configuration Space

- The goal of motion planning then, is to find a path in F that connects the initial configuration q_{start} to the goal configuration q_{goal}

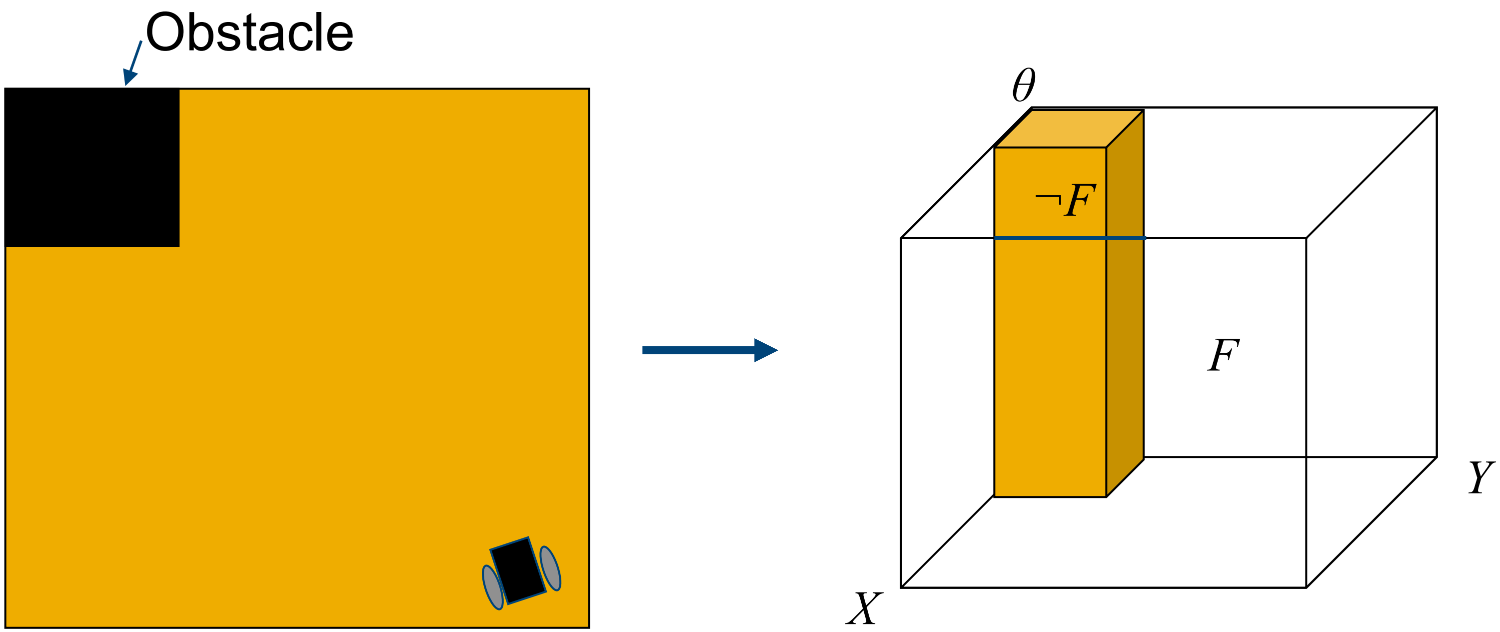
The Configuration Space

- Example 1: 2DOF manipulator:



The Configuration Space

- Example 2: Mobile Robot

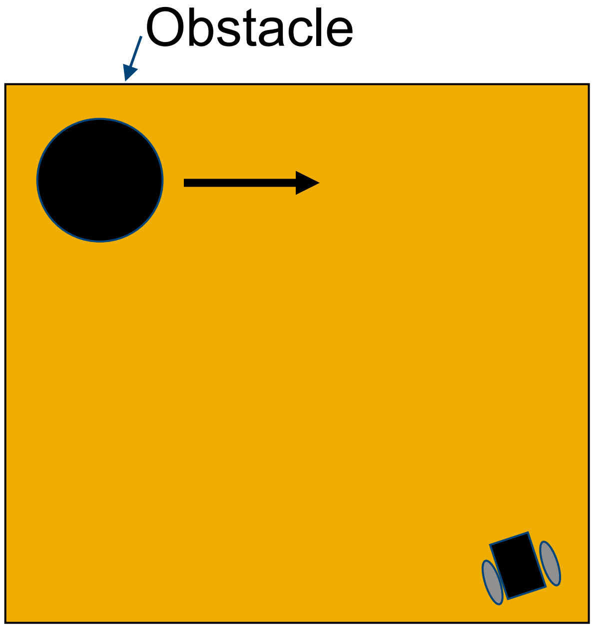


Workspace

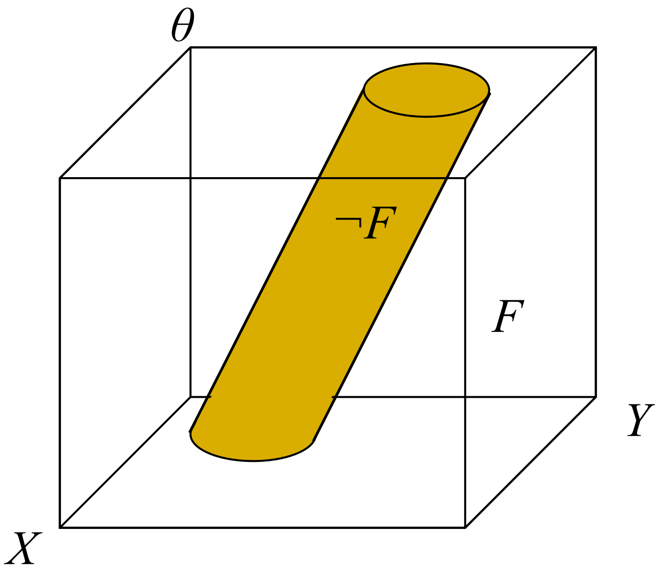
Configuration Space

The Configuration Space

- Example 3: Mobile Robot with moving obstacle



Workspace



Configuration Space

Introduction to Motion Planning

1. MP Overview
2. The Configuration Space
3. General Approach to MP
4. Metrics
5. PRMs
6. Single Query PRMs

General Approach to MP

- Motion planning is usually done with three steps:

  1. Define C
  2. Discretize C
  3. Search C

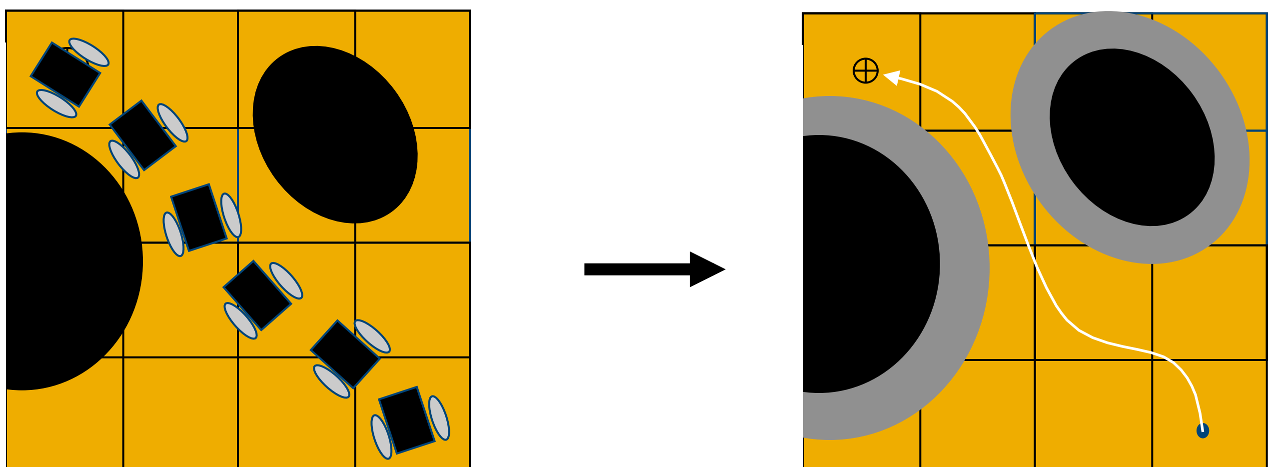
1. Define *C*

- Each planning problem may have a different definition of *C*.

  - Example 1: Include 3DOF for a mobile robot in static environment - (x,y,θ).
  - Example 2: Include only 2DOF for a mobile robot in static environment - (x,y).
  - Example 3: Include 5DOF for a mobile robot in dynamic environment - (x,y,θ,v,t).

1. Define *C*

- Plan paths for a point robot
  - Instead of using a robot of fixed dimensions/size, "grow" the obstacles to reflect how close the robot can get.



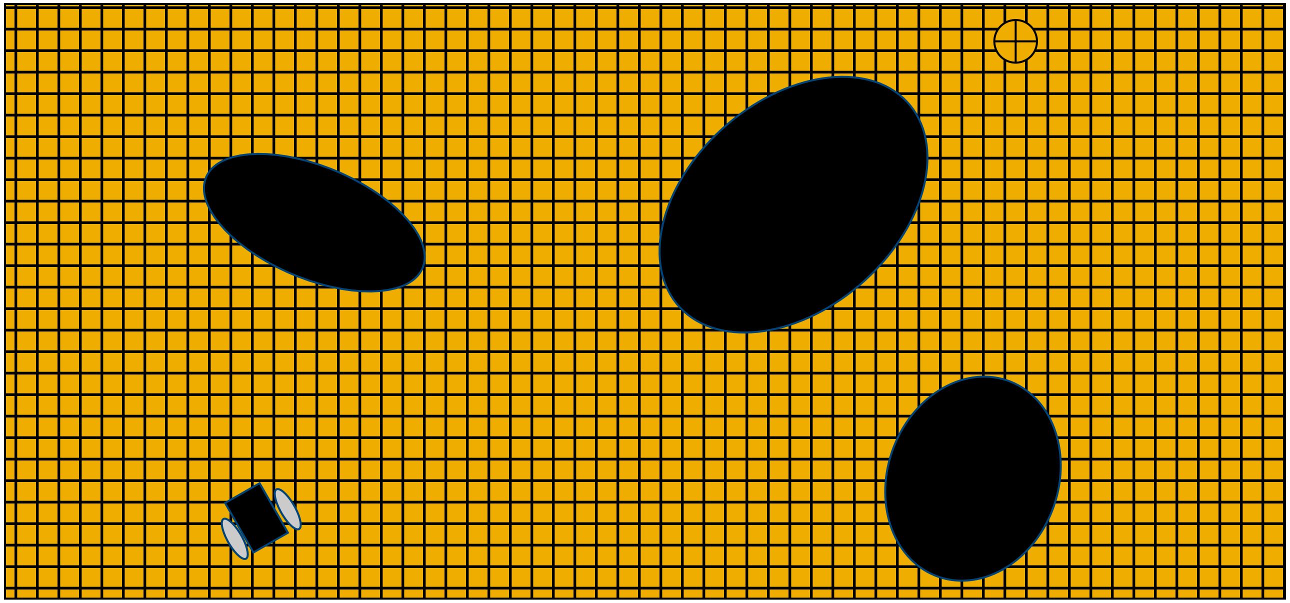
2. Discretize *C*

- Typical Discretizations:

1. Cell decomposition
2. Roadmap
3. Potential field

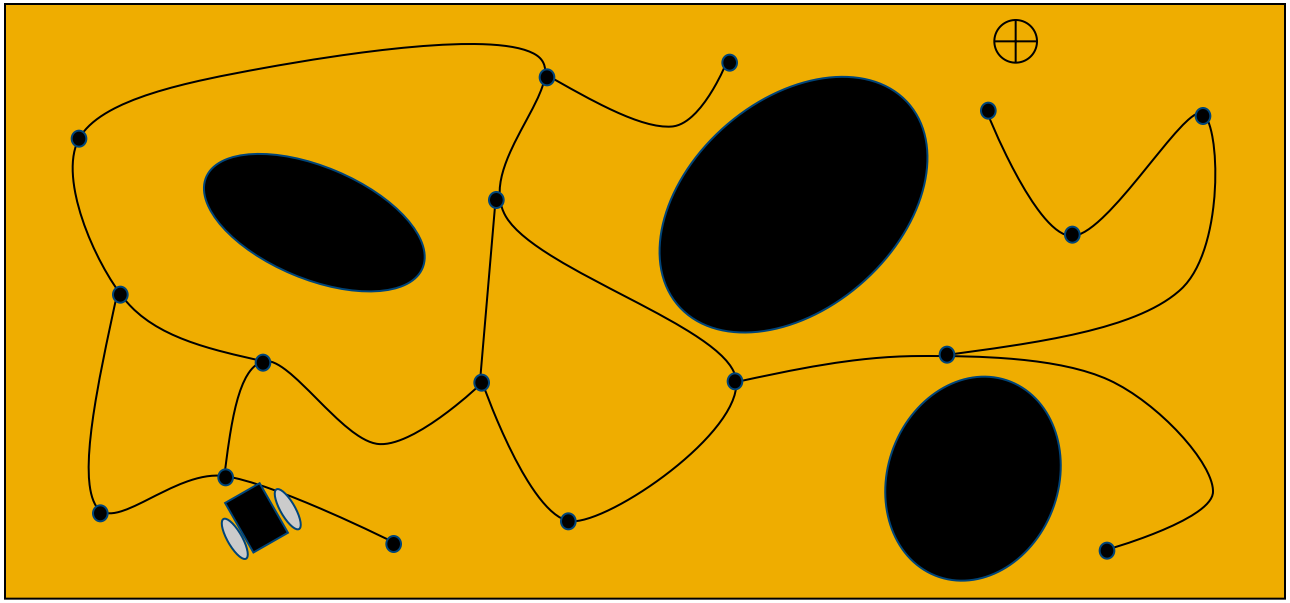
2. Discretize *C*

- Cell decomposition
  - Decompose the free space into simple cells and represent the connectivity of the free space by the adjacency graph of these cells



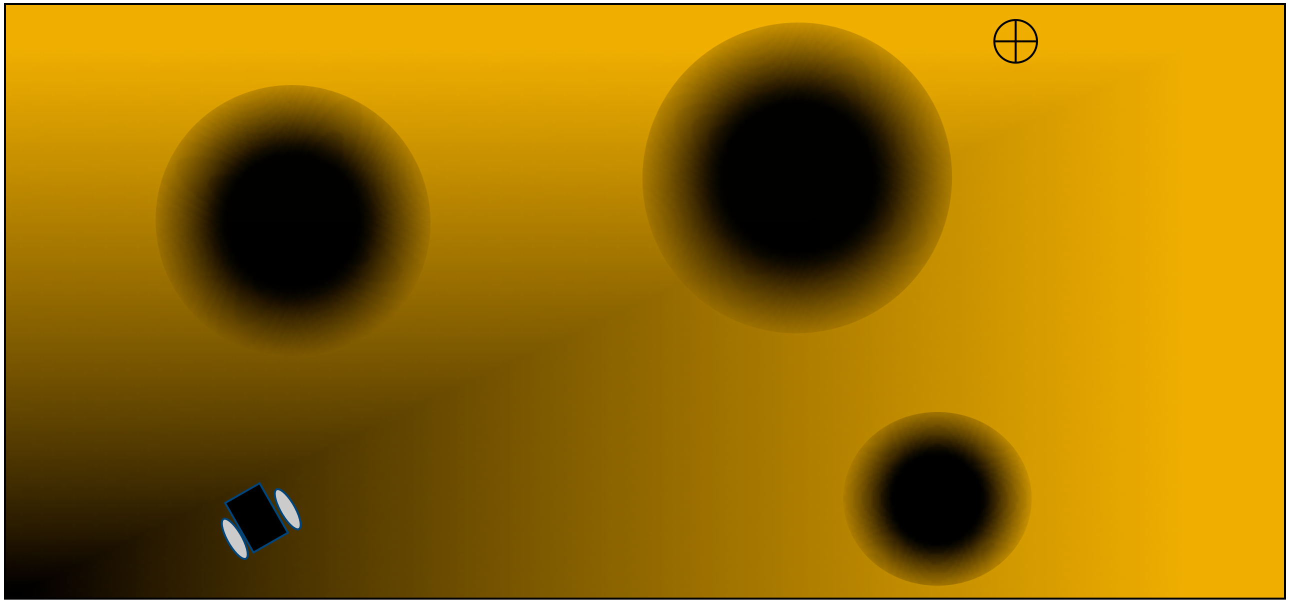
2. Discretize C

- Roadmap
  - Represent the connectivity of the free space by a network of 1-D curves



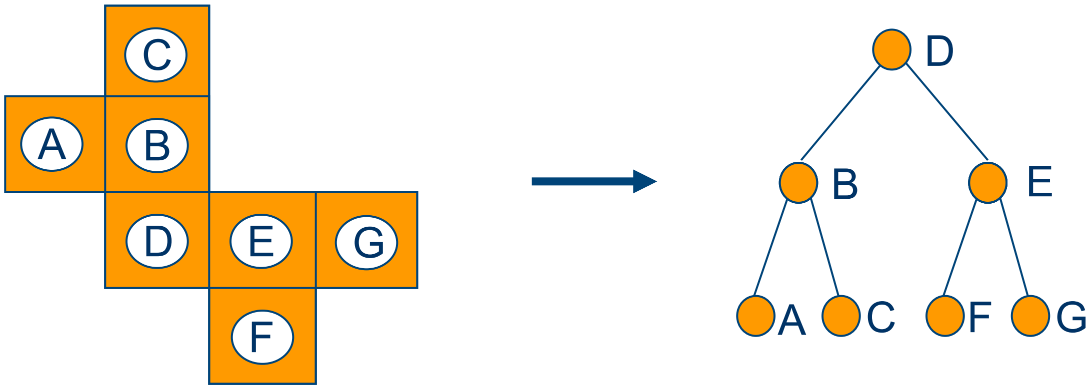
2. Discretize *C*

- Potential field
  - Define a function over the free space that has a global minimum at the goal configuration and follow its steepest descent



2. Search C

- Given a discretization of C, a search can be carried out using a Graph Search or gradient descent, etc.
  - Example: Find a path from D to G



Introduction to Motion Planning

1. MP Overview
2. The Configuration Space
3. General Approach to MP
4. Metrics

Metrics

- Metrics for which to compare planning algorithms:
  1. Speed or Complexity
  2. Completeness
  3. Optimality
  4. Feasibility of solutions

Metrics

1. Speed or Complexity
   - Often, planners are compared based on the **running time** of an algorithm.
     - Must specify the hardware when reporting, (e.g. processor type, …)
   - Example:
     - Planner A outperformed Planner B in that it took half the time to solve the same planning problem.

Metrics

1. Speed or Complexity
   - Planners are also compared based on the algorithm's **run time complexity**
     - i.e. the number of steps or operations an algorithm must take as a function of the size of the input.

Metrics

1. Speed or Complexity
   - Example: For M particles and N sensors, calculate the weights assuming expected measurements are known

```
for (int i=0; i<M; i++) {
        w(i) = 0.0001;
        for (int j=0; j<N; j++){
                w(i) *= gauss(z-z_exp(i,j));
        }
}
```

- In this example there are on the order of MxN operations, i.e $O(MN)$

Metrics

2. Completeness

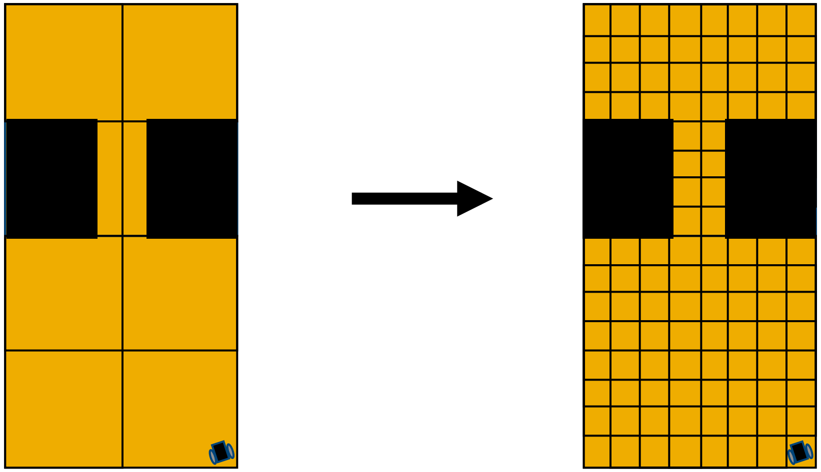
- A **complete** algorithm is one that is guaranteed to find a solution if one exists, or determine if no solution exists.

- Time Consuming!
  - An exhaustive search will search every possible path to see if it is a feasible solution.
  - A complete planner usually requires exponential time in the number of degrees of freedom, objects, etc.

Metrics

2. Completeness

- A **resolution complete** planner discretizes the space and returns a path whenever one exists in the discretized representation.



No Solution

Solution!

Metrics

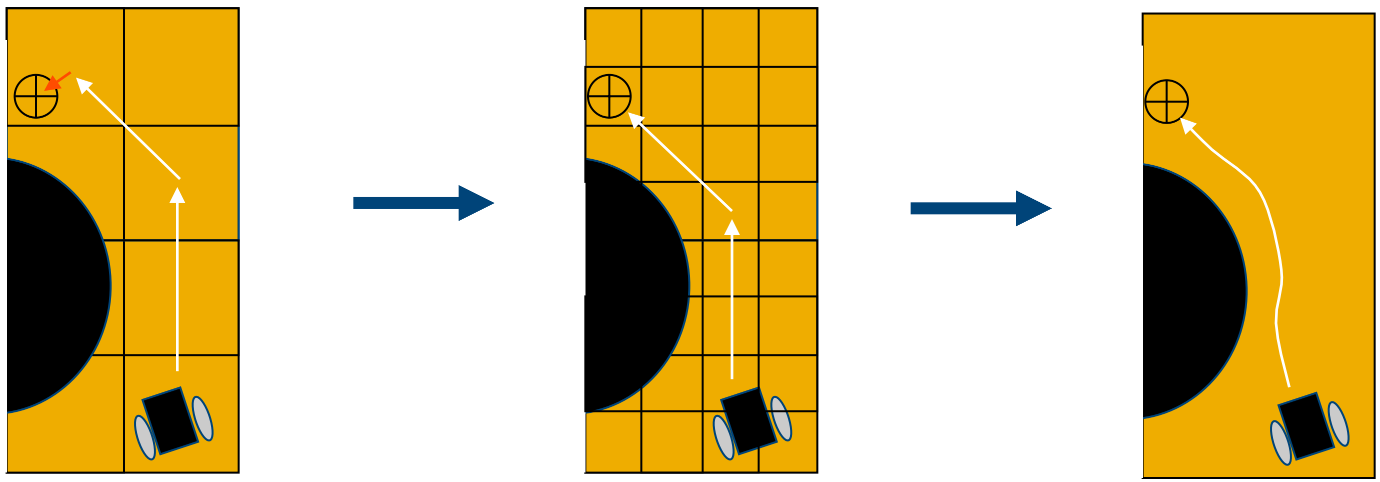
2. Completeness

- A **probabilistically complete** planner returns a path with high probability if a path exists. It may not terminate if no path exists.
  - E.g. $P(failure) \rightarrow 0$ as $time \rightarrow \infty$
- Weaker form of completeness, but usually faster.

Metrics

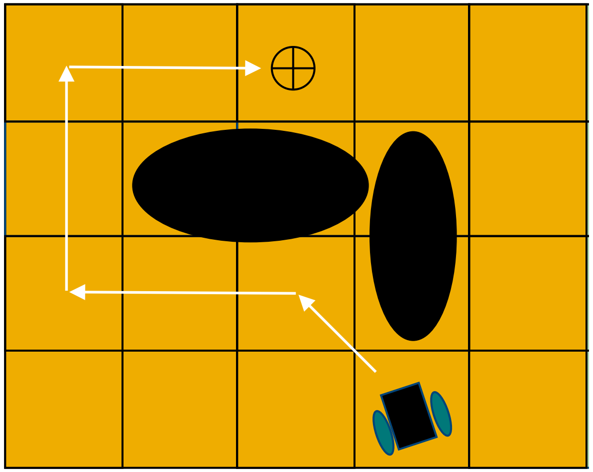
3. Optimality

- Resolution of Discretization can lead to sub-optimal solutions

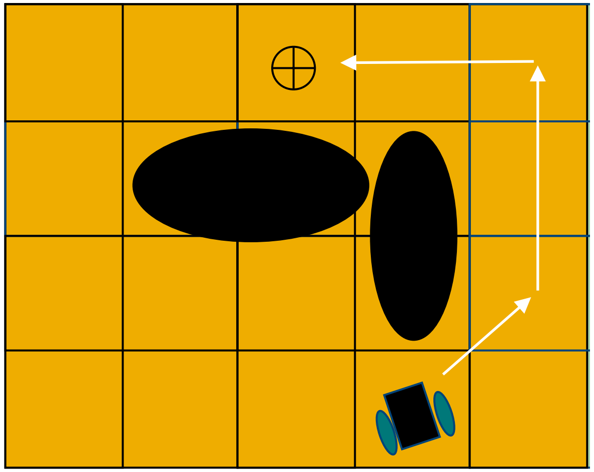


Metrics

3. Optimality
   - Some algorithms will only guarantee sub-optimal solutions (e.g. Greedy Search).



Sub-Optimal

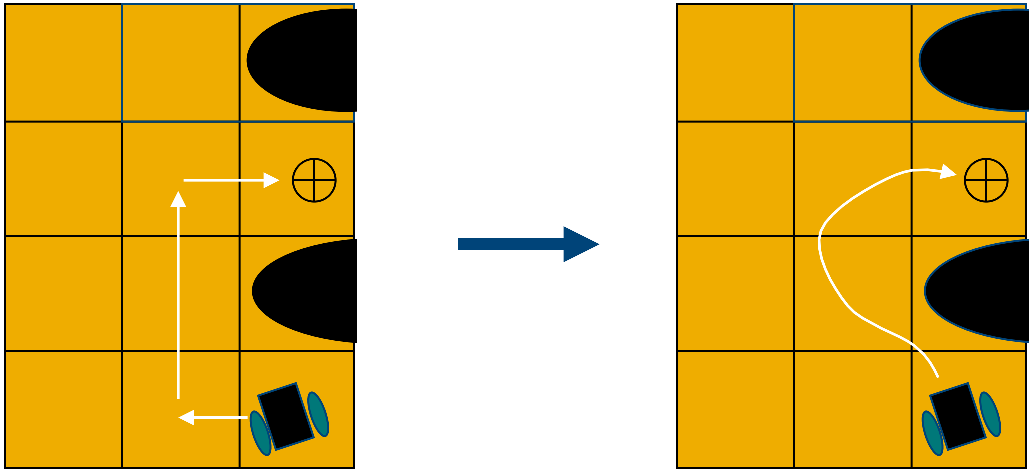


R. Optimal

Metrics

4. Feasibility of Solutions

- Not all planners take into account the exact model of the robot or environment.
- E.g. Non-differential drive robot



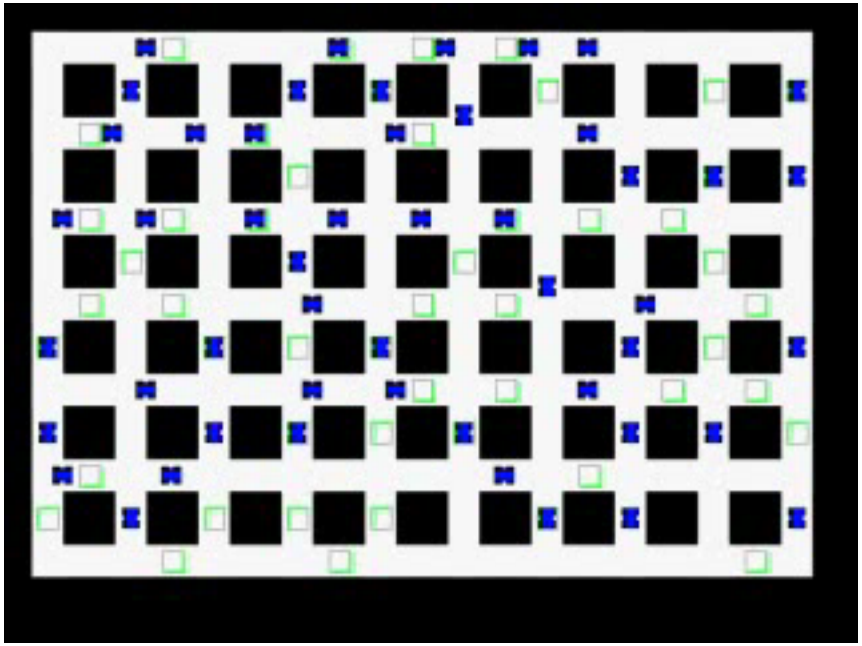
Infeasible　　Feasible

Metrics

- We are left with…
  - Theoretical algorithms
    - Strive for completeness and minimal worst-case complexity
    - Difficult to implement
  - Heuristic algorithms
    - Strive for efficiency in common situations
    - Use simplifying assumptions
    - Weaker completeness
    - Exponential algorithms that work in practice

Motion Planning: Searching the Configuration Space

- Example: Multi Robot MP



Introduction to Motion Planning

1. MP Overview
2. The Configuration Space
3. General Approach to MP
4. Metrics
5. PRMs
6. Single Query PRMs

Probabilistic Road Maps

- Definition:
  - A probabilistic road map is a discrete representation of a continuous configuration space generated by randomly sampling the free configurations of the C-space and connecting those points into a graph.

Probabilistic Road Maps

- Goal of PRMs:
  - **Quickly** generate a **small** roadmap of the Free Space F that has good **coverage** and **connectivity**

Probabilistic Road Maps

- PRMS have proven to useful in mapping free spaces that are difficult to model, or have many degrees of freedom.
  - This can facilitate fast planning for these situations

- Trade-off
  - PRMs often sacrifice **completeness** for **speed**

Probabilistic Road Maps

Moving Objects, Kindel

Probabilistic Road Maps

- Two Main Strategies:

1. Multi-Query:
   - Generate a single roadmap of *F* which can be used many times.
2. Single-Query:
   - Use a new roadmap to characterize the subspace of F which is relevant to the search problem.

Introduction to Motion Planning

1. MP Overview
2. The Configuration Space
3. General Approach to MP
4. Metrics
5. PRMs
6. Single Query PRMs

Motion Planning: Probabilistic Road Maps

- Single-Query PRMs (a.k.a. Rapidly Exploring Random Trees - RRTs)
  - Try to only sample a subspace of F that is relevant to the problem.
  - Probabilistically complete assuming C is *expansive* [Hsu et. al. 2000].
  - Very fast for many applications (allow for on-the-fly planning).

Motion Planning: Probabilistic Road Maps

- Two approaches:

1. Single Directional:
   - Grow a milestone tree from start configuration until the tree reaches the goal configuration

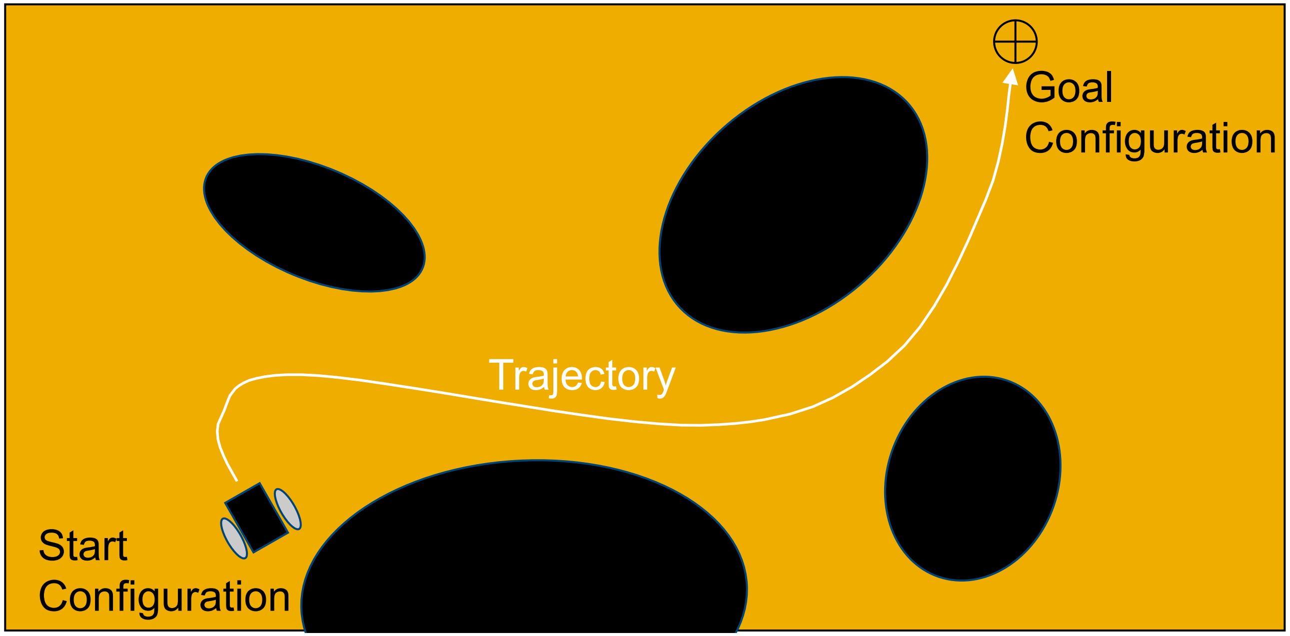
2. Bi-Directional:
   - Grow two trees, one from the start configuration and one from the goal configuration, until the two trees meet.
   - Can't consider time in the configuration space

Single Query PRMs: Outline

1. Introduction
2. Algorithm Overview
3. Sampling strategies

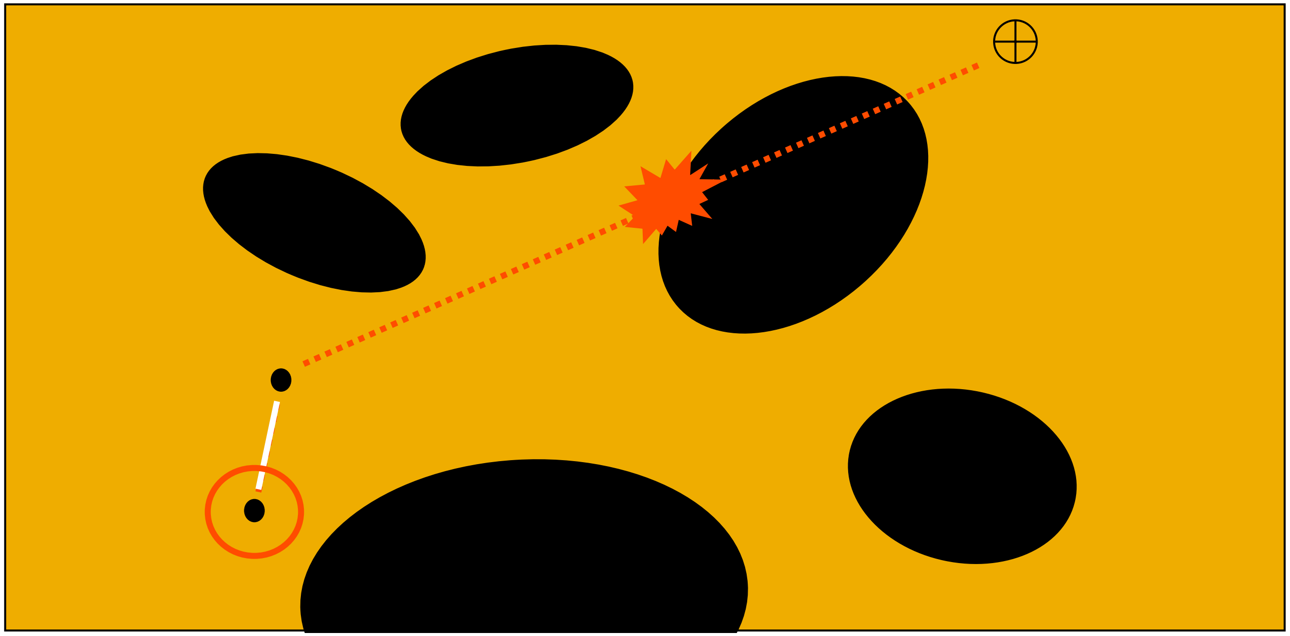
MP Overview

- Example:



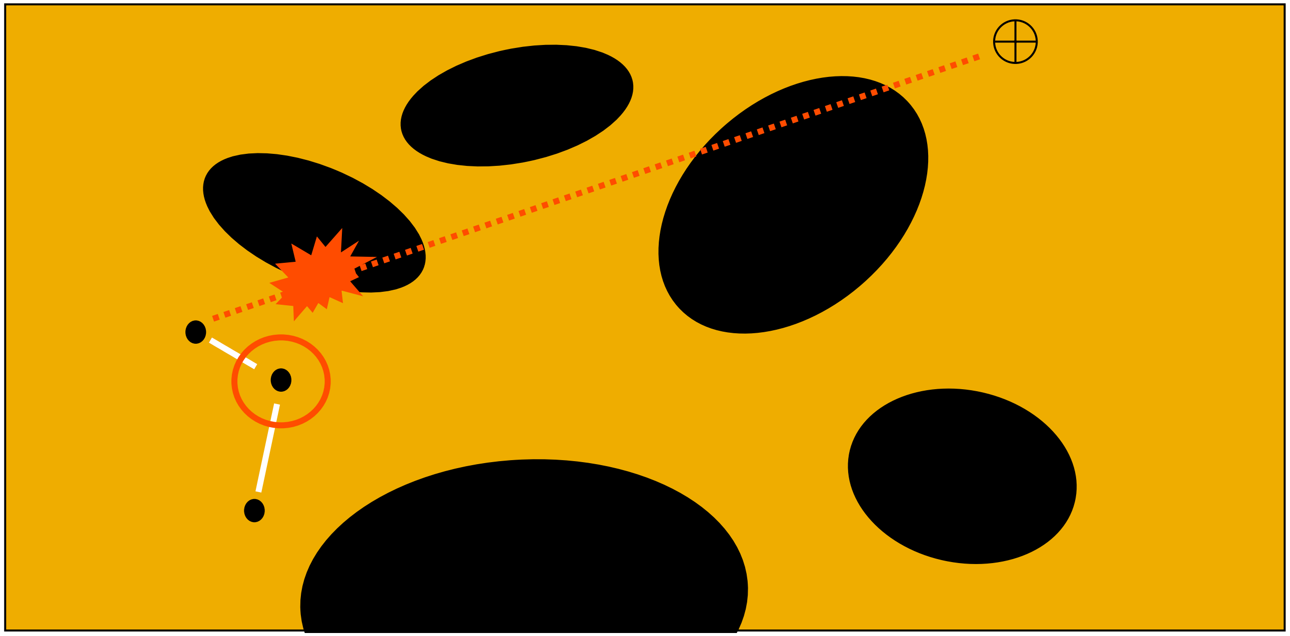
Motion Planning

- Example: Iteration 1



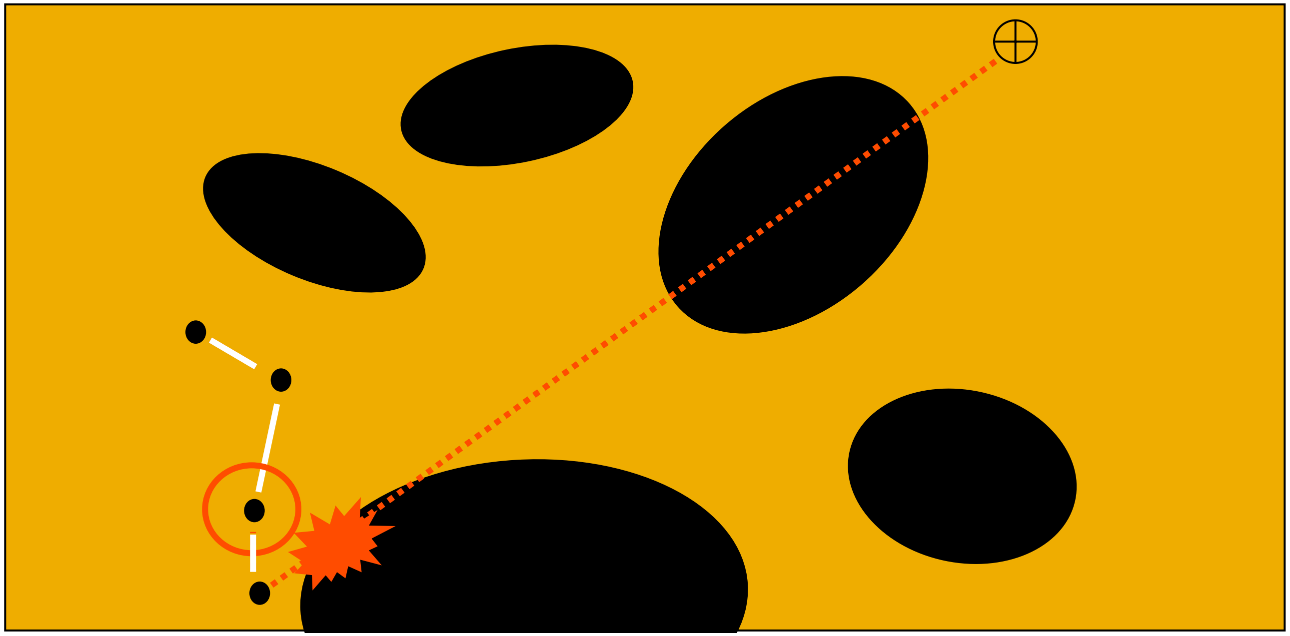
Motion Planning

- Example: Iteration 2



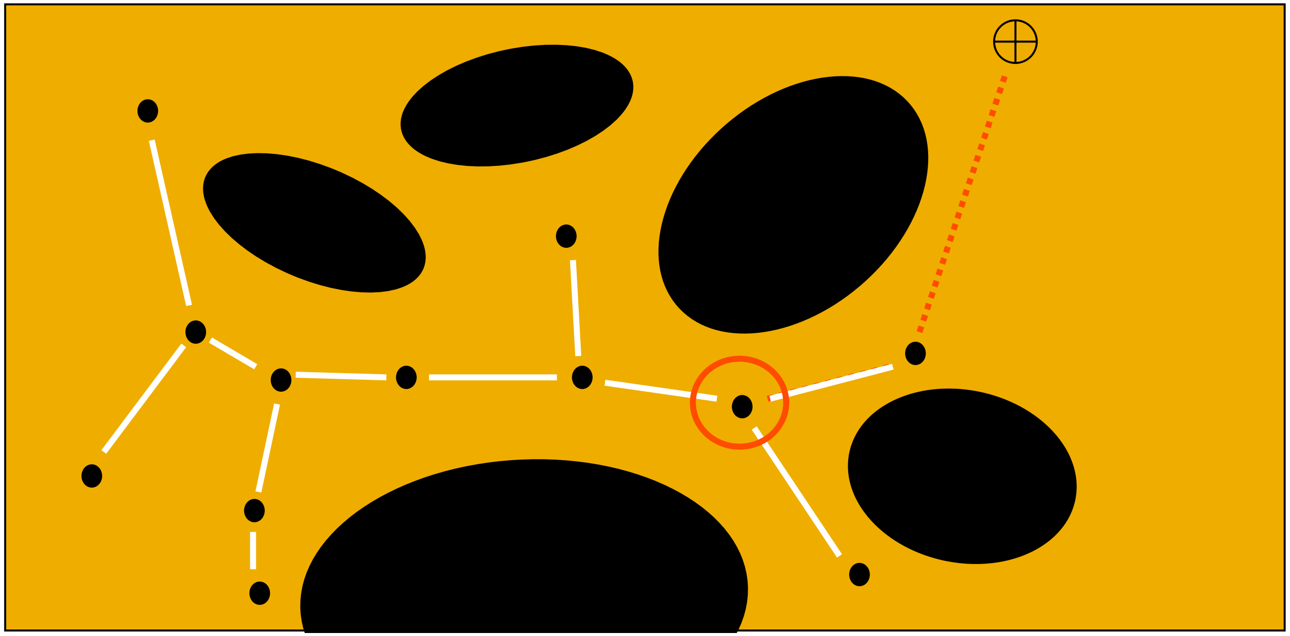
Motion Planning

- Example: Iteration 3



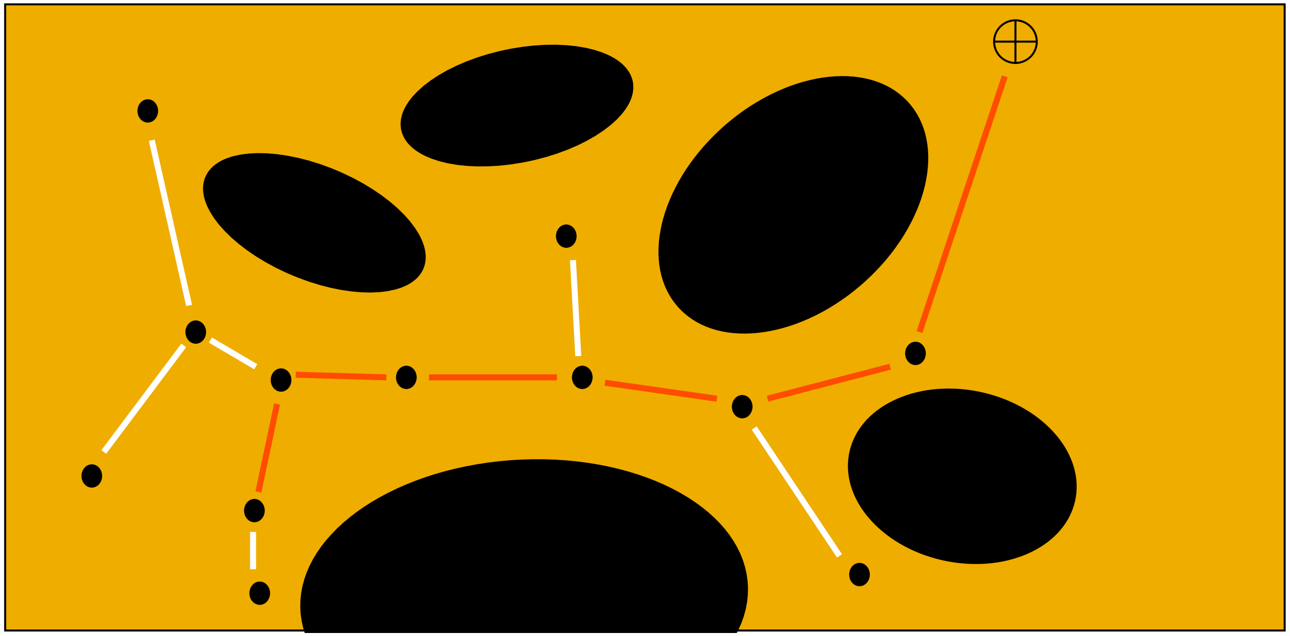
Motion Planning

- Example: Iteration 11



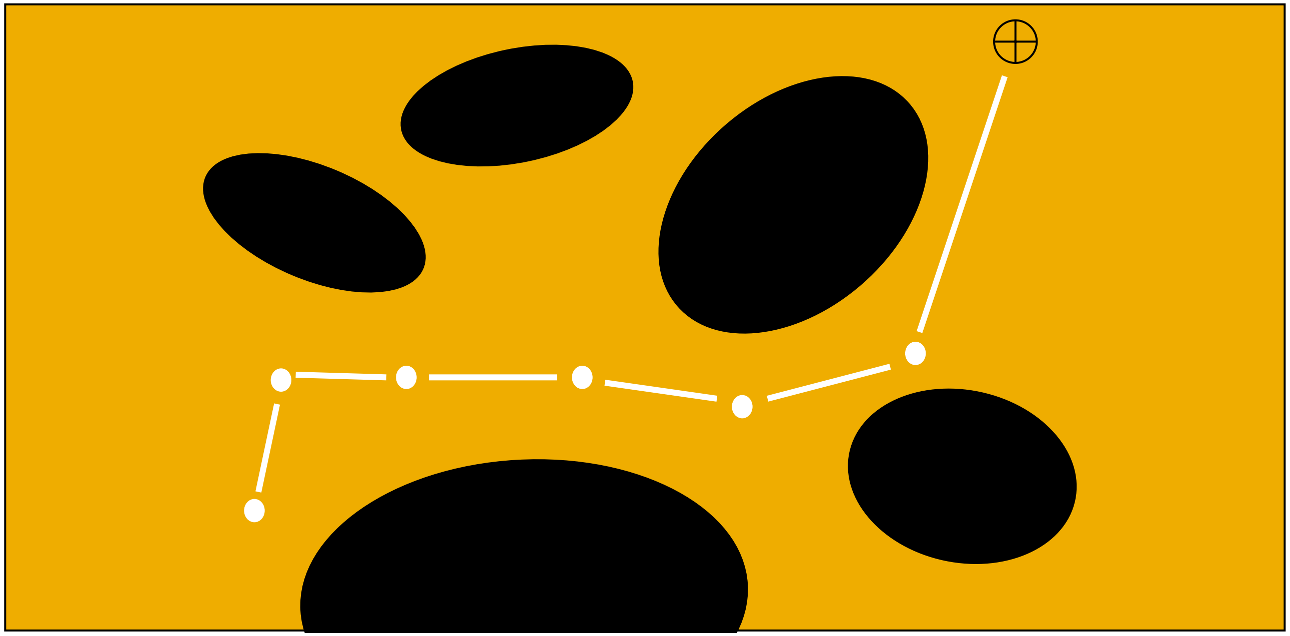
Motion Planning

- Example: Construct Path



Motion Planning

- Example: Construct Path



Probabilistic Road Maps: Learning Phase

- Nomenclature

$R = (\mathbf{N}, \mathbf{E})$	RoadMap
$\mathbf{N}$	Set of Nodes
$\mathbf{E}$	Set of edges
c	Configuration
e	edge

Motion Planning: Probabilistic Road Maps

- Algorithm
  1. Add start configuration c_{start} to $R(\mathbf{N}, \mathbf{E})$
  2. Loop
  3. Randomly Select New Node *c* to expand
  4. Randomly Generate new Node c' from c
  5. If edge e from c to c' is collision-free
  6. Add (c', e) to R
  7. If c' belongs to endgame region, return path
  8. Return if stopping criteria is met

Single Query PRMs: Outline

1. Introduction
2. Algorithm Overview
3. Sampling strategies
   - Node Selection (step 3)
   - Node Generation (step 4)
   - Endgame Region (step 7)

Motion Planning: PRM Node Selection

- One could pick the next node for expansion by picking from all nodes in the roadmap with equal probability.

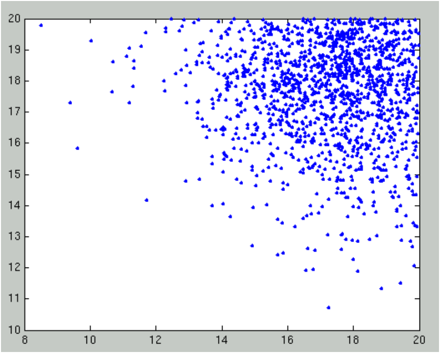
  - This is easy to implement, but leads to poor expansion ⟶ **Clustering**

Motion Planning: PRM Node Selection

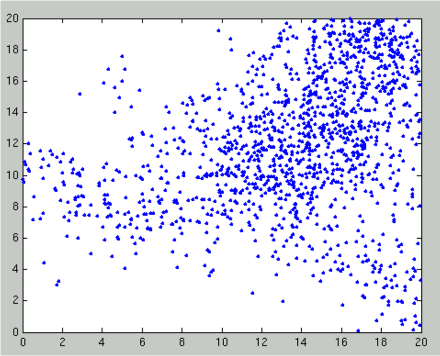
- Cont'
  - Method is to weight the random selection of nodes to expand, this can greatly affect the roadmap coverage of the configuration space.
  - Want to pick nodes with probability proportional to the inverse of node density.

Motion Planning: PRM Node Selection

- Example:
  - Presented is a 2DOF configuration space where the initial node in the roadmap is located in the upper right corner.
  - After X iterations, the roadmap produced from an unweighted expansion has limited coverage.



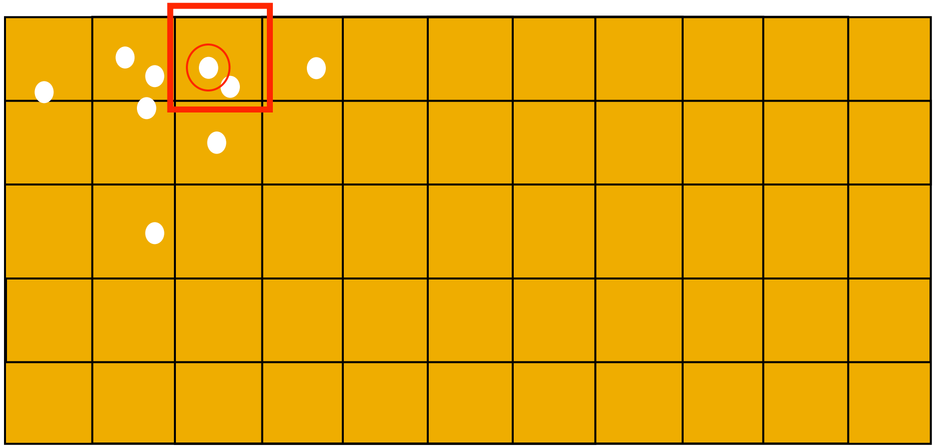
Unweighted



Weighted

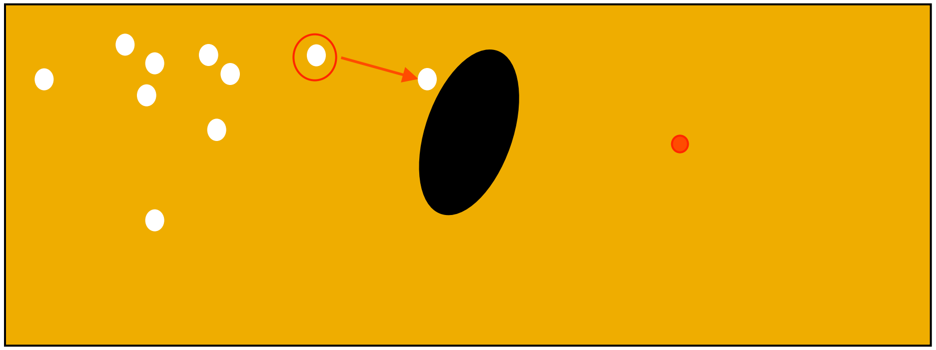
Motion Planning: PRM Node Selection Technique 1

- The workspace was divided up into cells to form a grid [Kindel 2000].
  - Algorithm:
    1. Randomly pick an occupied cell from the grid.
    2. Randomly pick a milestone in that cell.



Motion Planning: PRM Node Selection Technique 2

- Commonly used in Rapidly exploring Random Trees (RRTs) [Lavalle]
  - Algorithm:
    1. Randomly pick configuration c_{rand} from C.
    2. Find node c from R that is closest to node c_{rand}
    3. Expand from c in the direction of c_{rand}



Single Query PRMs: Outline

1. Introduction
2. Algorithm Overview
3. Sampling strategies
   - Node Selection (step 3)
   - Node Generation (step 4)
   - Endgame Region (step 7)

Motion Planning: PRM Milestone Generation

- Use random control inputs to propagate robot from previous node c to new configuration c'
  - Algorithm:
    1. Randomly select controls u and Δt
    2. Use known dynamics/kinematics equation *f* of robot to generate new configuration
       $$c' = f(c, u, \Delta t)$$
    3. If path from c to c' is collision-free, then add c' to R

Motion Planning: PRM Milestone Generation

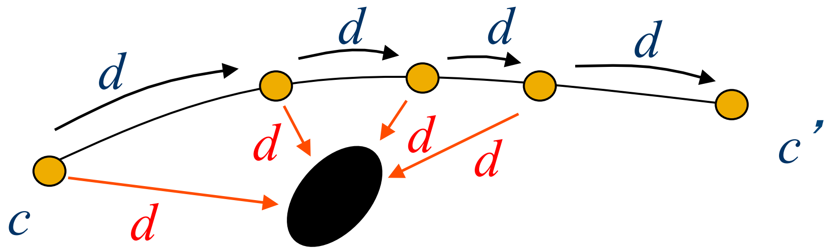
- Example: Differential drive robot
  1. Randomly select controls $\dot{\varphi}_{left}$, $\dot{\varphi}_{right}$ and Δt
  2. Propagate:
     1. Get Δs_{left} and Δs_{right}
     2. Calculate new state c' with:

$$c' = f(x, y, \theta, \Delta s_r, \Delta s_l) = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \frac{\Delta s_r + \Delta s_l}{2} \cos\left(\theta + \frac{\Delta s_r - \Delta s_l}{2b}\right) \\ \frac{\Delta s_r + \Delta s_l}{2} \sin\left(\theta + \frac{\Delta s_r - \Delta s_l}{2b}\right) \\ \frac{\Delta s_r - \Delta s_l}{b} \end{bmatrix}$$

     3. Use iterative search to check for collisions on path.

Motion Planning: PRM Milestone Generation

- Example: Differential drive robot (cont’)
  - Iterative Collision checking is simple but not always efficient:
  - Algorithm:
    1. Calculate distance d to nearest obstacle
    2. Propagate forward distance *d* along path from c to c'
    3. If d is too small, return **collision**
    4. If c reaches or surpasses c', return **collision-free**

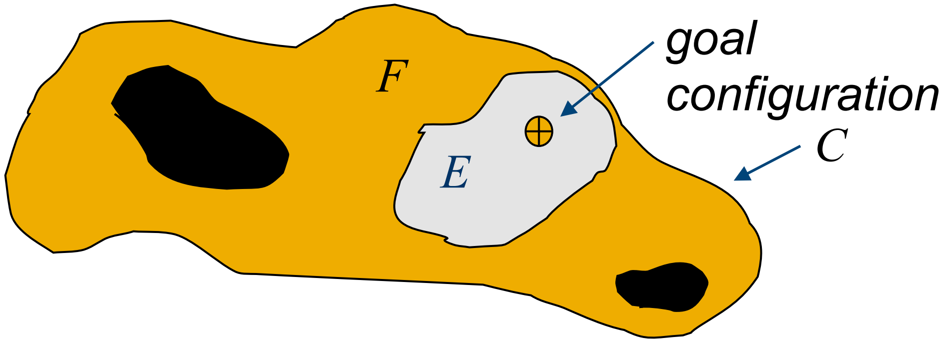


Single Query PRMs: Outline

1. Introduction
2. Algorithm Overview
3. Sampling strategies
   - Node Selection (step 3)
   - Node Generation (step 4)
   - Endgame Region (step 7)

Motion Planning: PRM Endgame Region

- We define the endgame region E, to be the set of configurations that have a **simple** connection to the goal configuration.
  - For each planning problem, we can define a unique method of making **simple** connections.
  - This method will inherently define E.



Motion Planning: PRM Endgame Region

- Given the complexity of most configuration spaces, it is very difficult to model E.
  - In practice, we develop a simple admissibility test to calculate if a configuration c' belongs to the E
  - At every iteration of the algorithm, this test is used to determine if newly generated configurations are connected to the goal configuration.

Motion Planning: PRM Endgame Region

- In defining E, we need two things for good performance:

1. The region E should be **large**: this increases the chance that a newly generated milestone will belong to E and provide us a solution.
2. The admissibility test to be as **fast** as possible. This test is conducted at every iteration of the algorithm and will greatly affect the algorithm running time.

Motion Planning: PRM Endgame Region

- Several endgame definitions exist:
  1. The set of all configurations within some radius r of the goal configuration

Motion Planning: PRM Endgame Region

- Several endgame definitions exist:
  1. The set of all configurations within some radius r of the goal configuration
  2. The set of all configurations that have "simple", collision-free connection with the goal configuration.
     - Example: Use circular arc for differential drive robots.