CMSC828T
Vision, Planning And Control In Aerial Robotics

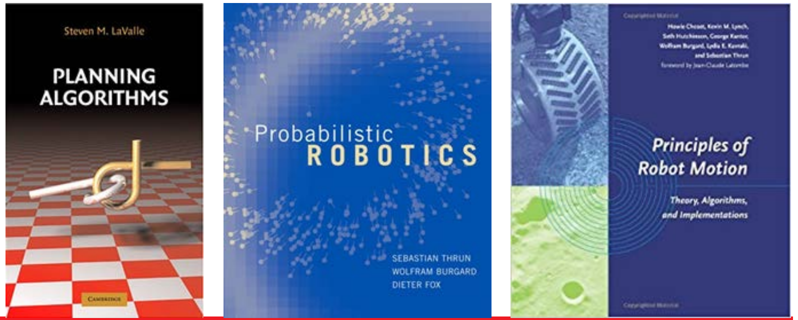
MOTION PLANNING

Topics Covered

- Topic Introductions
  - Necessity of planning
  - Challenges in planning
- Motion Planning Concepts
  - Geometric Modelling
  - Half space representations
  - Configuration Spaces
  - C-obstacle, C-free
- Planning in Discrete Spaces
  - Graph Search
  - DFS, BFS
  - Dijkstra's Algorithm
  - A* Algorithm
- Planning in Continuous Spaces
  - PRM
  - RRT

Course Bibliography

- Main Text
  - **LaValle, Steven M.** *Planning Algorithms.* Cambridge University Press, 2006. http://planning.cs.uiuc.edu/
- Reference Texts
  - **Thrun, Sebastian, et al.** *Probabilistic Robotics. Vol. 1.* Cambridge, MIT press, 2005.
  - **Choset, Howie, et al.** *Principles of Robot Motion: Theory, Algorithms, and Implementations*, MIT press, 2005.



COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

Planning for Robotics

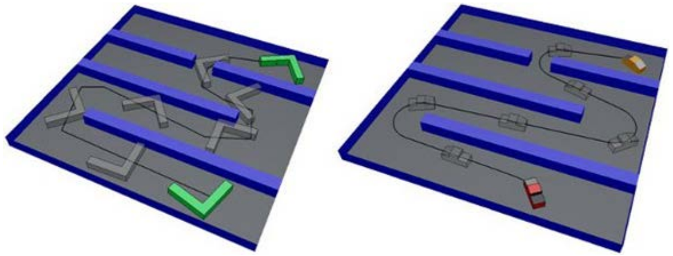
- How did we get here?
  - Was in **Location A** a while ago.
  - Decided to go to class (good job!)
  - Found out the classroom, a.k.a. **Location B**
  - ***Planned*** the trip from **A** to **B**
  - There were **obstacles** along the way, avoided them (successfully, I hope)
  - Probably used a **map** (new students, welcome)

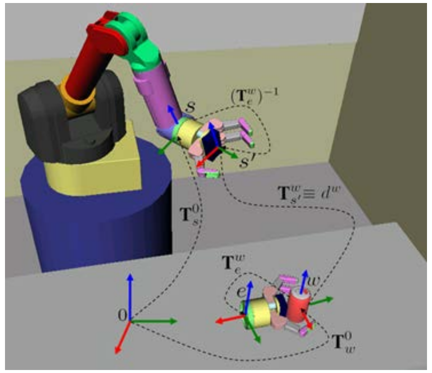
Planning for Robotics

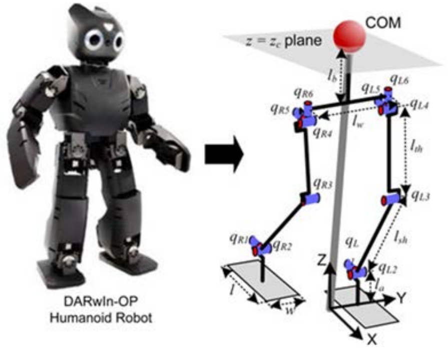
- Planning, then, is selecting a certain sequence of actions to meet a certain goal. The previous example talks about *Path Planning* but planning is obviously not limited only to that.
  - The sequence of actions must:
    - Be hierarchical
    - Take contingency into account
    - Ensure cost-reward tradeoff is done optimally

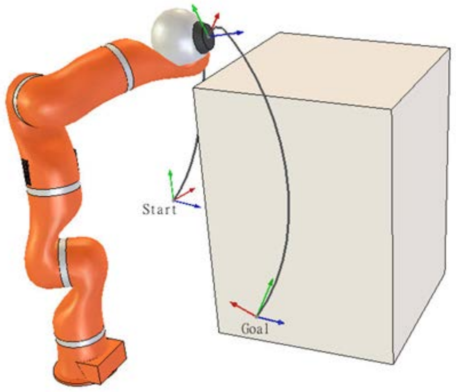
Planning for Robotics

- As mentioned previously, planning can (and must) be done for several applications, including but not limited to:
  - Task planning
  - Path planning
  - Trajectory planning
  - Foot placement planning
  - Grasp planning
  - and many, many more...



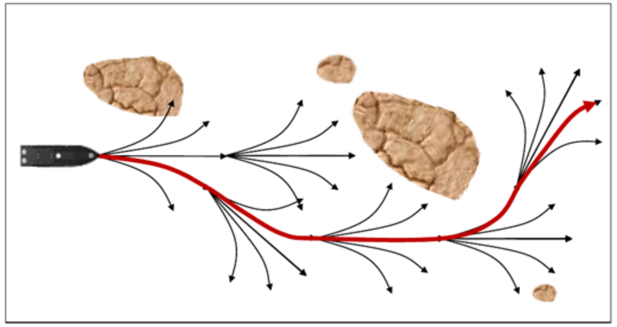


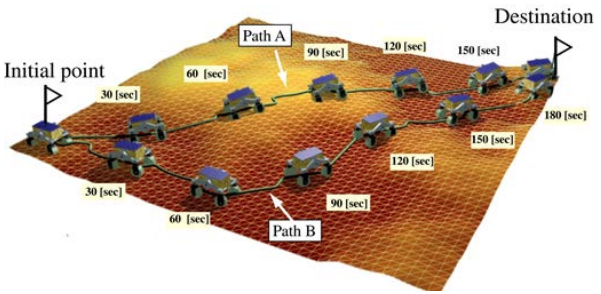


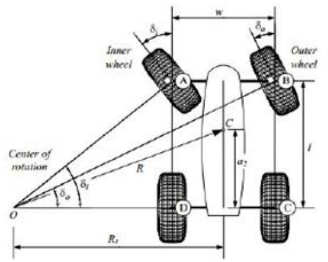


Challenges in Planning for Robotics

- Not an easy task, to create and execute a robot plan.
- Execution time + length optimization of collision-free path or trajectory
- Search space complexity
- Terrain interaction complexity
- Dynamic obstacles, unpredictable environment
- Robot dynamics, high-speed robots

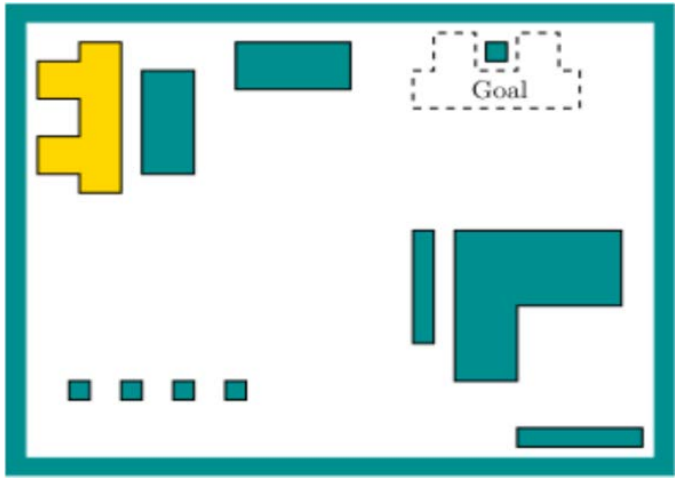


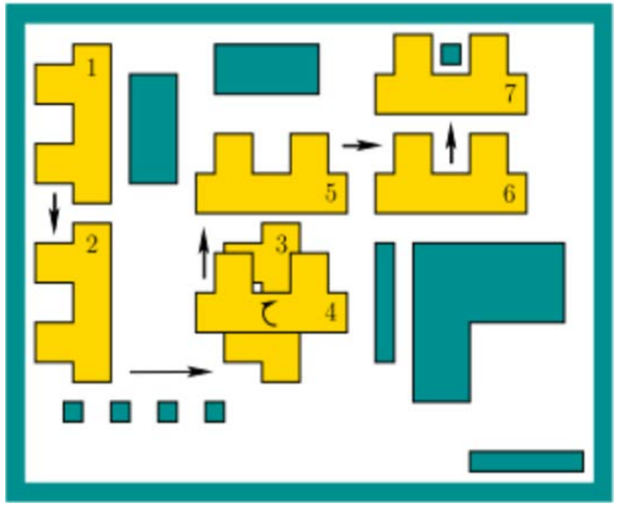




Geometric Modelling

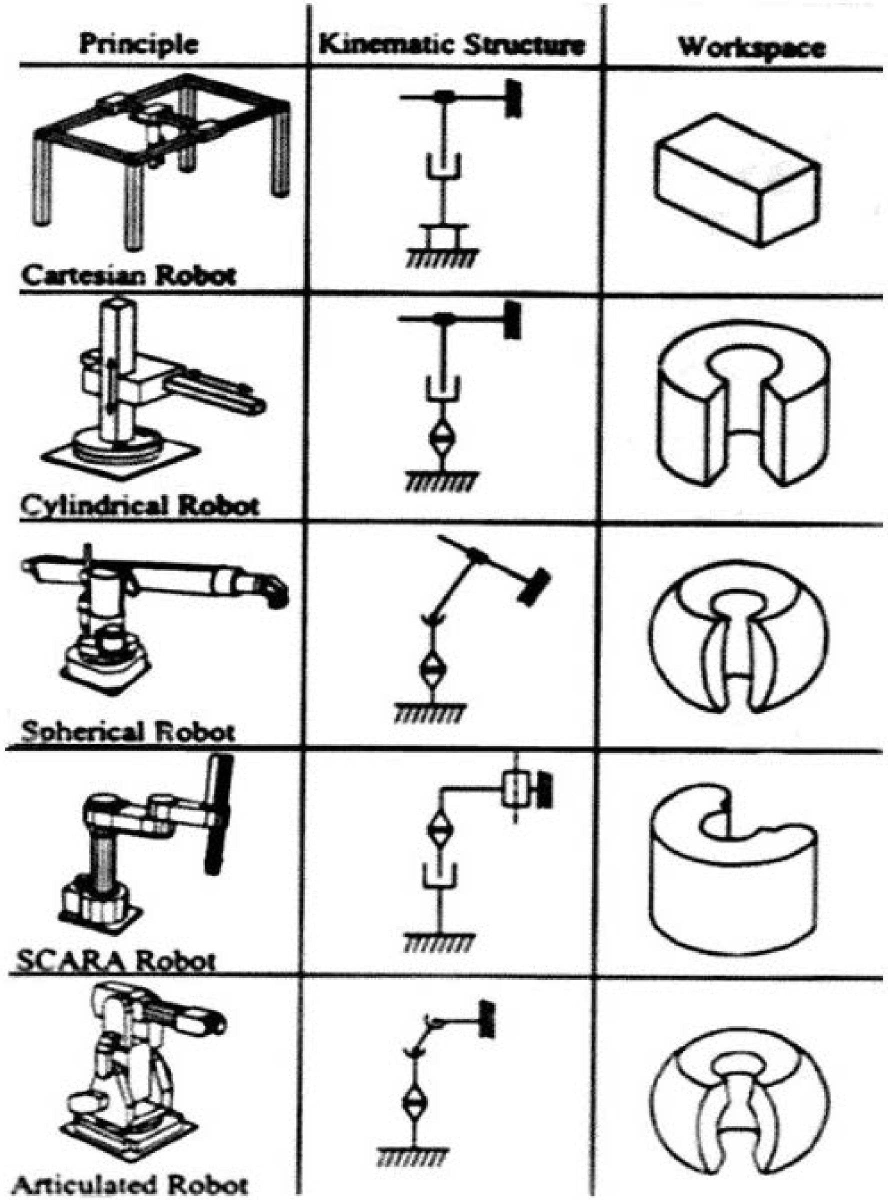
- Why geometric modelling?
  - Need to produce motion plan of robot's collision-free motions to the goal
  - Identify locations/points where robot can go without collision, capture motion of robot within environment
- Accurate model of robot and obstacles required to create plan





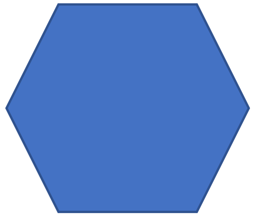
Geometric Modelling

- Robot $A(q) \subseteq W$ and obstacles $O \subset$ W usually positioned in a workspace $W = \mathbb{R}^2$ or $W = \mathbb{R}^3$
- Here, q is the configuration of the robot and, along with obstacles, are closed subsets of W
- These geometric representations must be computationally efficient without loss in detail.

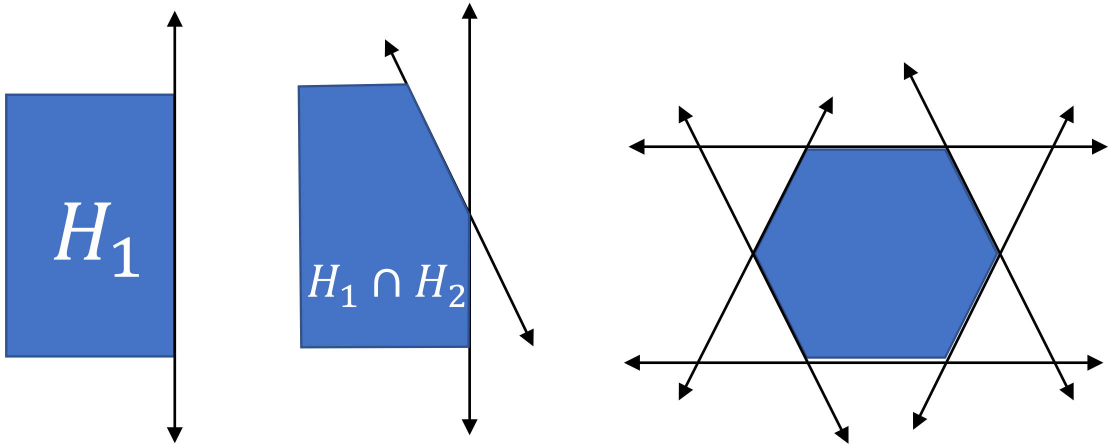


Polygonal Models: Half-space Representation

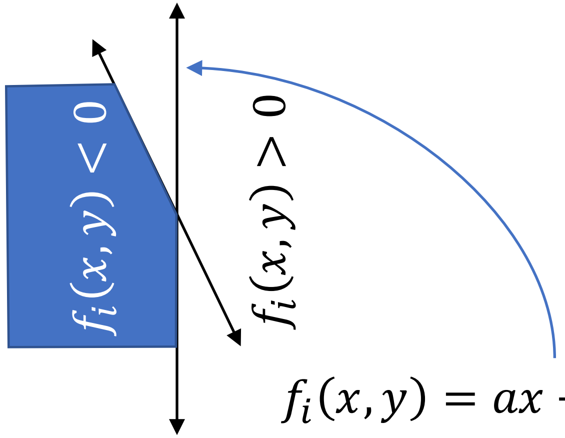
- Solid representation of convex polygon O as Boolean combination of half-plane primitives
- $H_i = \{(x, y) \in W \mid f_i(x, y) \leq 0\}$

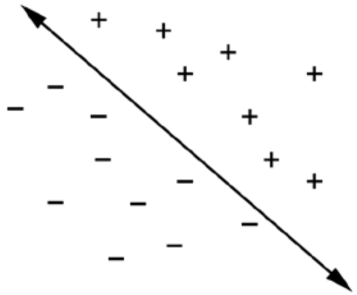


$$O = H_1 \cap H_2 \cap \cdots \cap H_n$$



$H_1 \cap H_2 \cap \cdots \cap H_n$



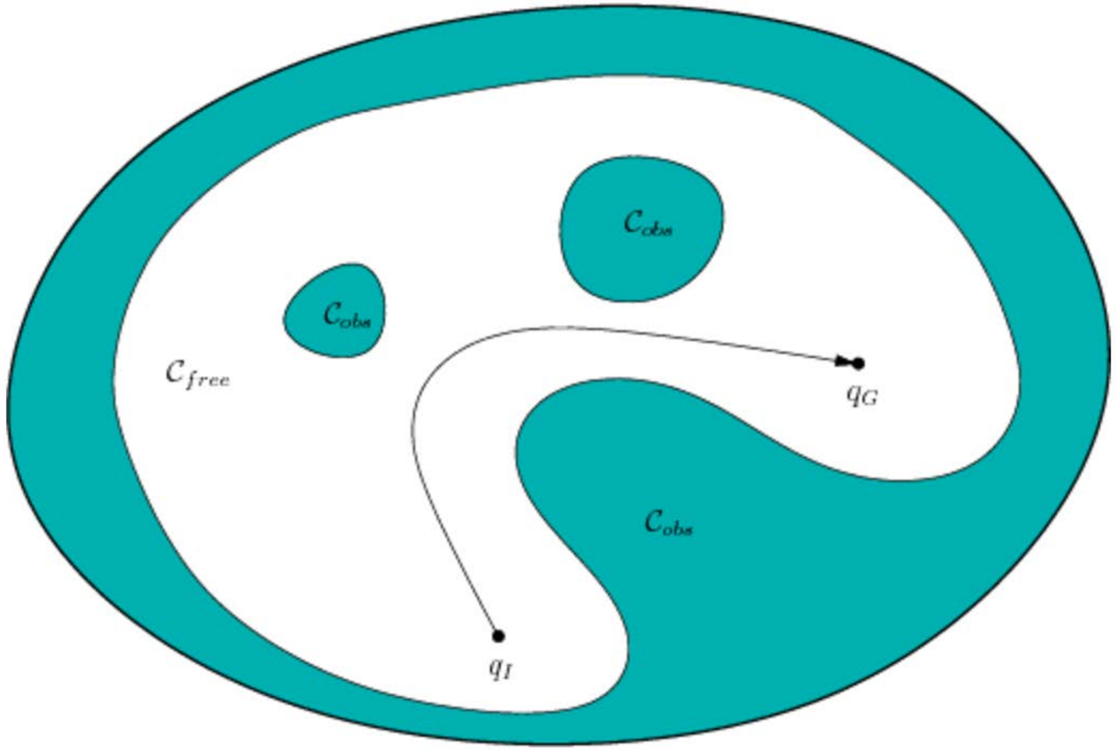


$f_i(x, y) = ax + by + c = 0$

Configuration Space

- A complete specification of the position of every point in the system - *configuration*
- The space of all such configurations is the **C-Space** or **Configuration Space**
- The basic task of motion planning is to find a path from q_I to q_G in C_{free}. The entire blob is

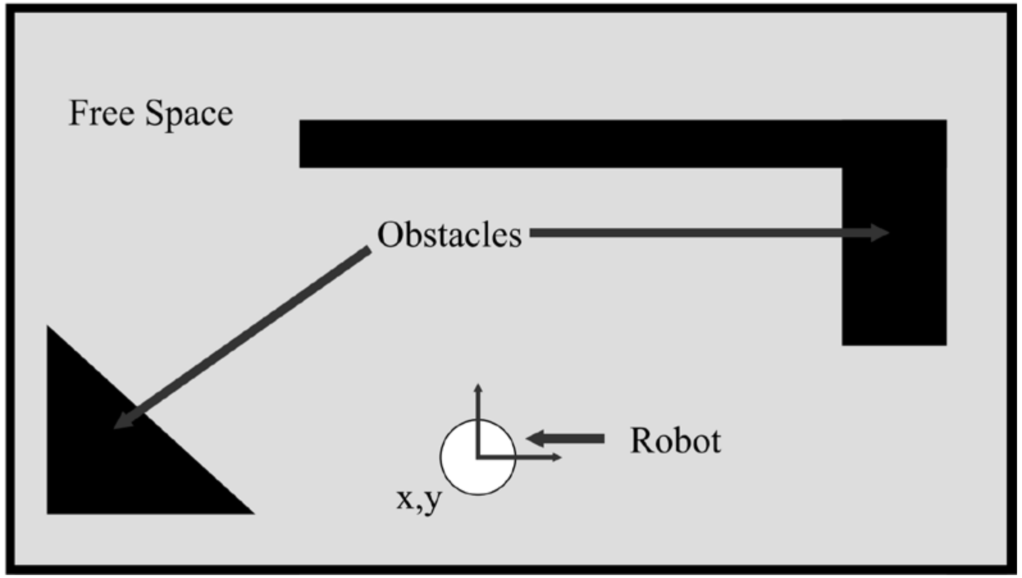
$$C = C_{free} \cup C_{obs}$$

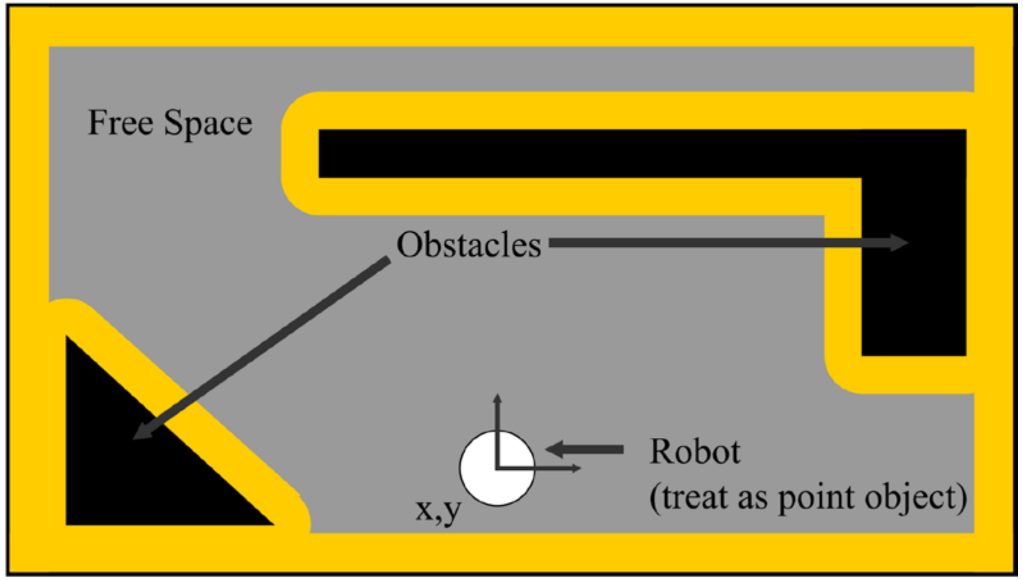


COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

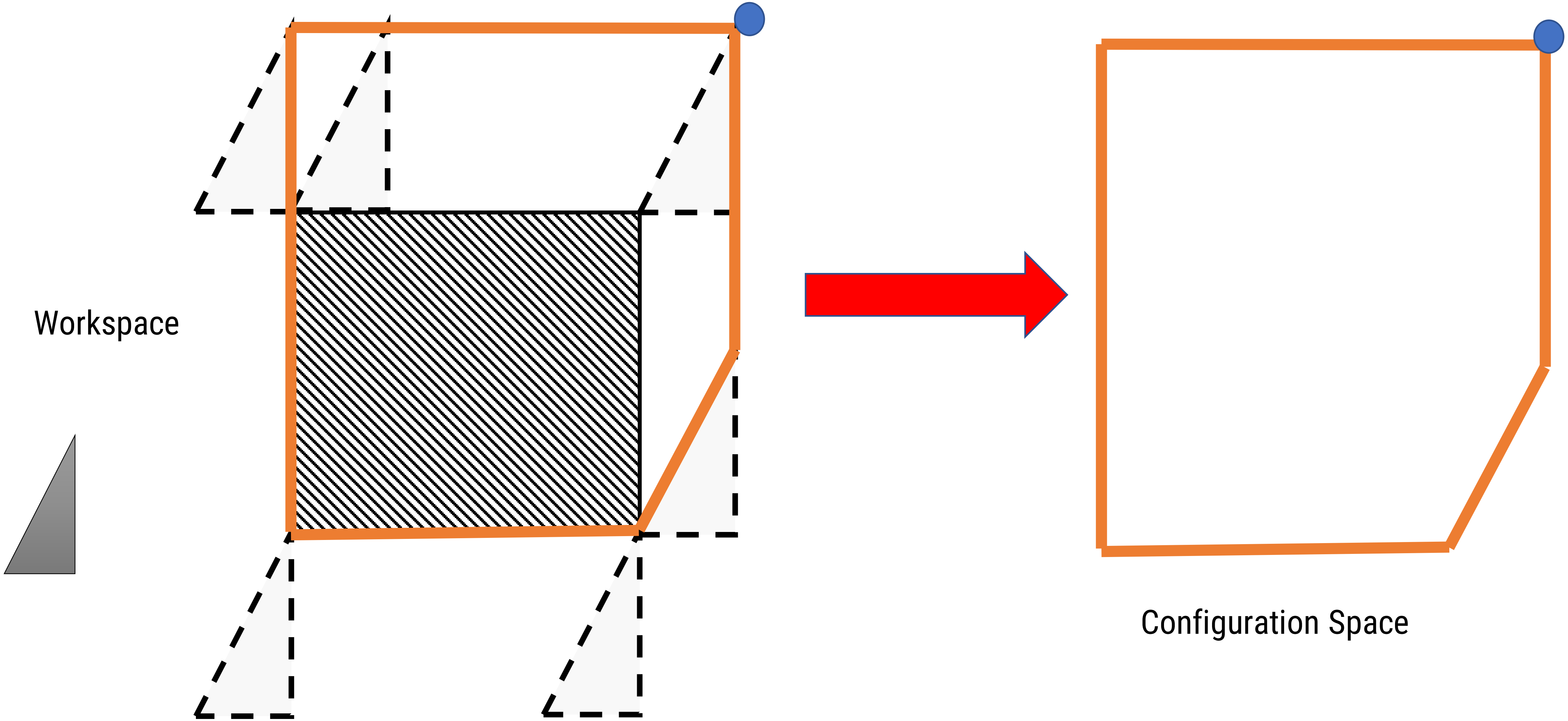
Configuration Space

- For a robot with k degrees of freedom, the **C-space** is a coordinate system with *one dimension per degree of freedom.*
- In **C-space**, a robot pose is simplified into a **POINT** and an obstacle is a **COMPLEX SHAPE**
- Thus, we transform the problem from Cartesian to Configuration space, solve it and bring it back.

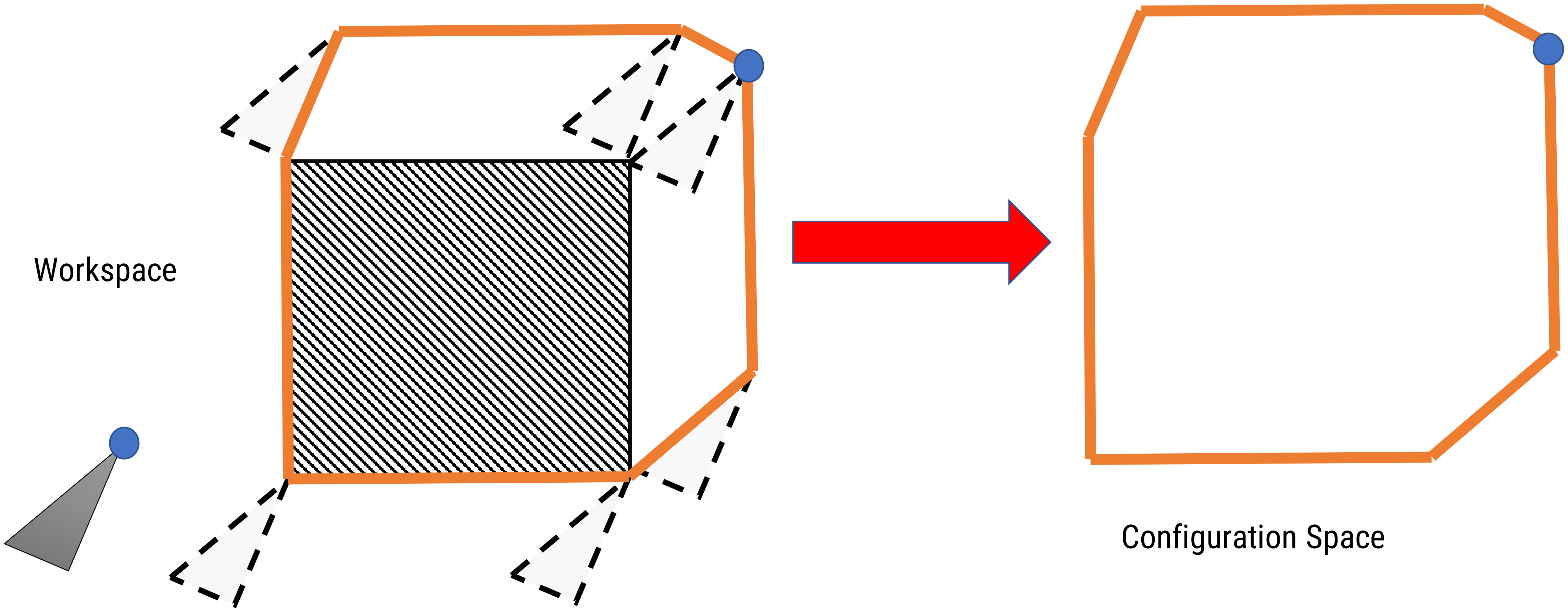




Rigid robot translating in 2-D Workspace



Robot translating + rotating in 2-D workspace

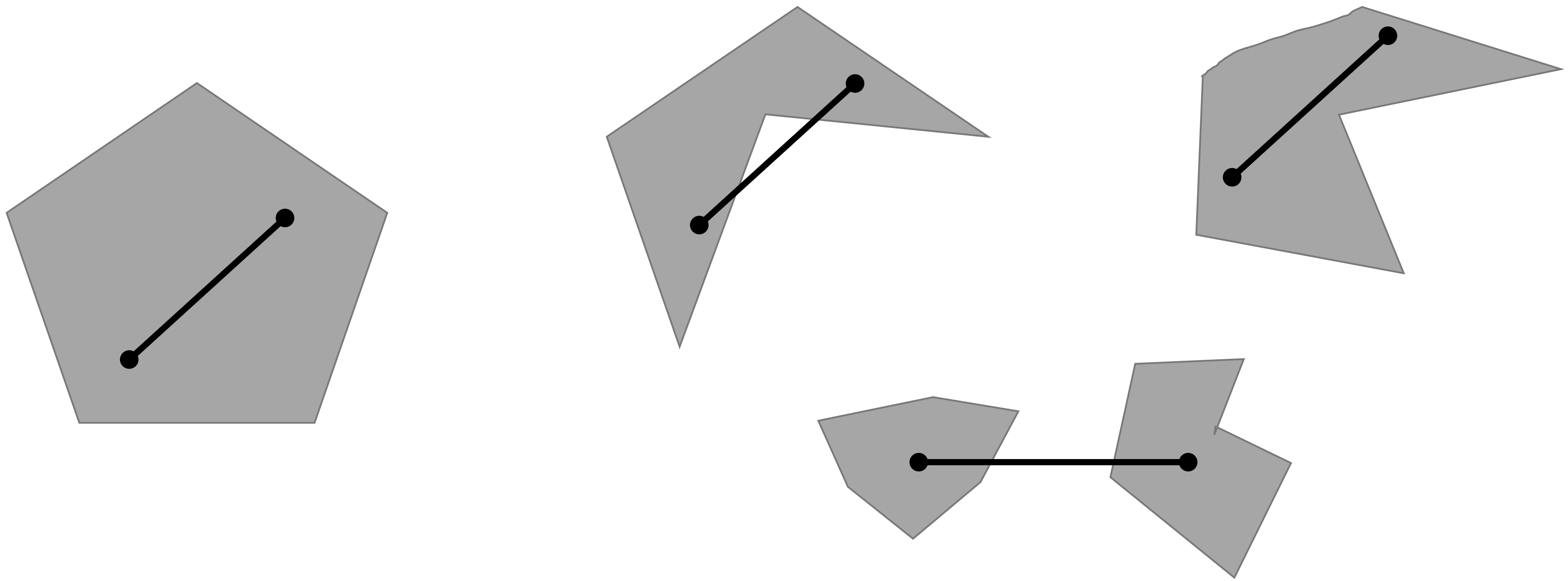


Minkowski Sum and Difference

- The **Minkowski Sum** of two sets A and B is denoted by
$$A \oplus \mathrm{B} = \{\mathrm{a} + \mathrm{b} |\ \mathrm{a} \in A, b \in B\}$$
- The **Minkowski Difference** of two sets A and B is denoted by
$$A \ominus \mathrm{B} = \{\mathrm{a} - \mathrm{b} |\ \mathrm{a} \in A, b \in B\}$$
- The **Minkowski Sum of Convex Polygons** of two convex polygons, P and Q having m and n vertices, respectively, is a convex polygon $P \oplus Q$ having $m + n$ vertices.

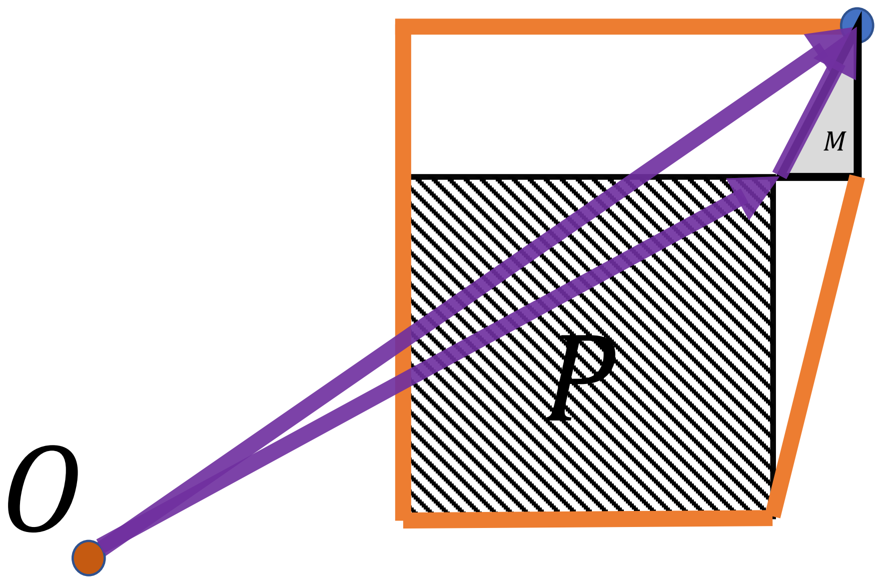
Convexity of Sets

A set S is convex *if and only if* every line segment connecting two points in S is contained within S



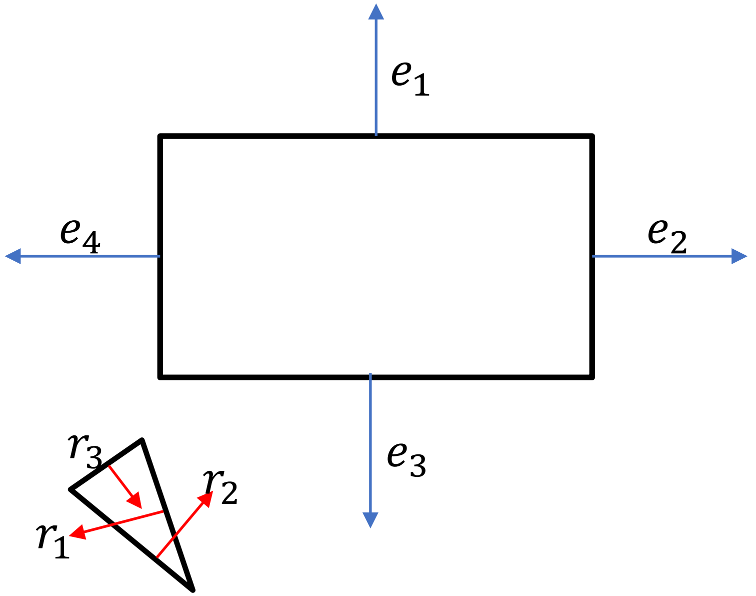
Obstacle in Workspace to C-Space

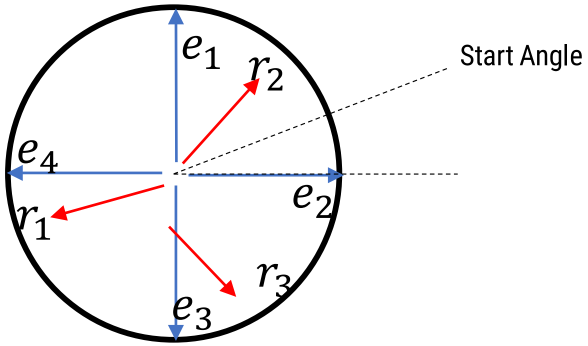
- If P is an obstacle in the workspace, and M is a moving object (the robot, say) then the C-Space obstacle corresponding to P is given by $P \ominus M$
- $\boldsymbol{QO_i} = \{\boldsymbol{q} \in \boldsymbol{Q} | \boldsymbol{R}(\boldsymbol{q}) \cap \boldsymbol{WO_i} \neq \emptyset\}$
  - Q is 2D configuration space
  - W is Workspace
  - R is robot



Star Algorithm

- Determines sliding contact of robot around obstacle, and order of sliding operations.
- Choose one direction, CW or CCW, and remain consistent.
- Fix robot normals to point inwards, obstacle normals pointing outwards.
- Pick start angle and proceed according to direction.





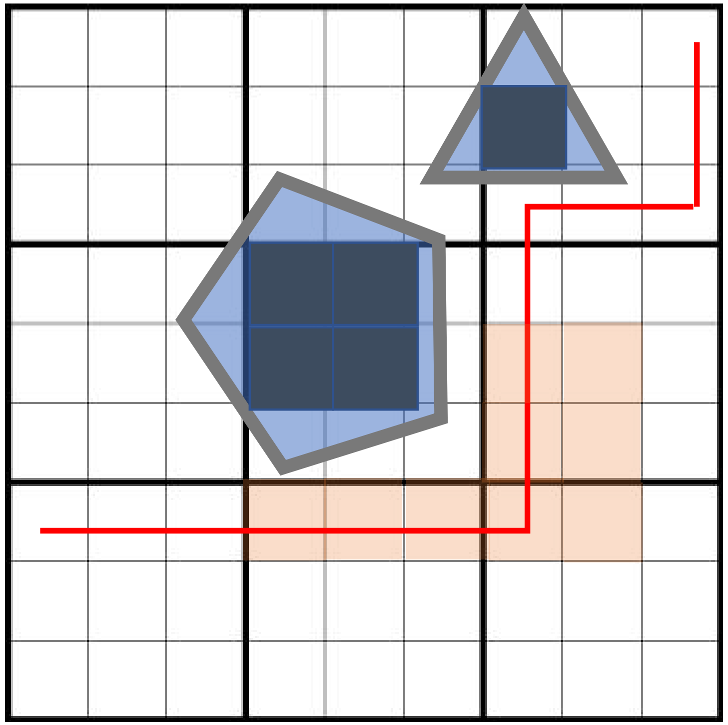
C-Free

- Successfully constructed obstacle representation in C-space, but none for free space.
- Remember, C-space dimensions increases very fast; 3-space corresponds to 6D for rigid body, further articulation adds even more dimensions.
- Obviously, cannot have explicit representation of free space for high degree of freedom robots (manipulators, for example)
- What is the solution?
- APRROXIMATE or RANDOMIZE

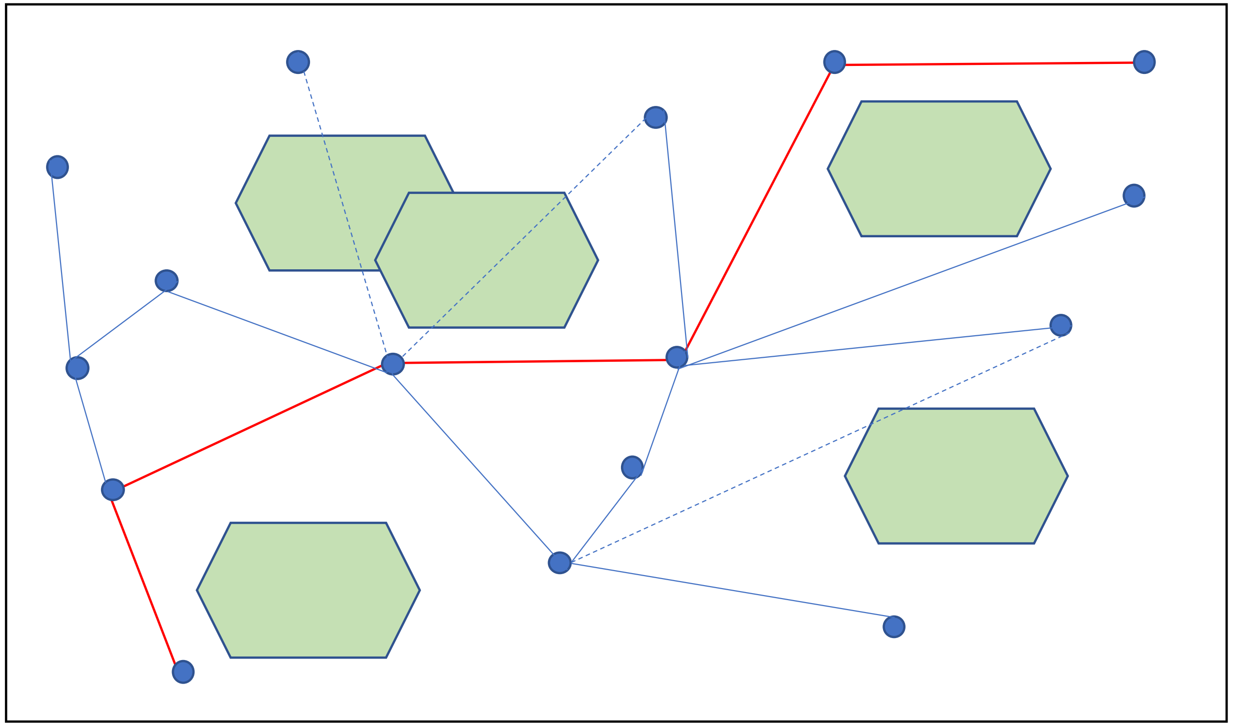
C-Free: APRROXIMATE or RANDOMIZE

- **Approximate** the free space using regular subdivision.
  - Convert to grid, maybe.
  - Pick only "complete" cells in grid.
- **Randomize** the robot in free space by sampling and checking pose in C-space
- **Tradeoff** between **performance** and **completeness**

C-Free: APPROXIMATE or RANDOMIZE



Cell Decomposition



Probabilistic Roadmaps

Completeness of Planning Algorithms

- **Complete**
  - If solution exists, finds it; else reports failure.
  - Geometry based planners
- **Semi-complete**
  - If solution exists, finds it; else may run forever.
- **Resolution Complete**
  - If solution exists, finds it; else reports solution not found for specified resolution
  - Grid- or Interval- based planners
- **Probabilistic Complete**
  - If solutions exists, probability that it will be found tends to 1, as the number of search iteration tends to infinity.
  - Sampling-based planners

References

- Steven M. Lavalle
  - http://planning.cs.uiuc.edu
  - http://msl.cs.uiuc.edu/~lavalle/icra12/
- Howie Choset with slides from G.D. Hager, Z. Dodds, and Dinesh Mocha
  - http://www.cs.cmu.edu/~motionplanning/lecture/lecture.html
- http://courses.csail.mit.edu/6.141/spring2010/pub/lectures/
- http://www.eng.utah.edu/~cs6370/Lectures/
- http://robotics.caltech.edu/~jwb/courses/ME132/handouts/
- Prof. Krishnanand Kaipa's lecture notes
  - ENPM808C – Planning for Autonomous Robots

References

- https://www2.cs.arizona.edu/classes/cs437/fall11/
- http://www.cs.wustl.edu/~pless/546/lectures/lecMinkSum.html