EE5110: Special Topics in Automation & Control—Autonomous Systems

Motion planning (I)

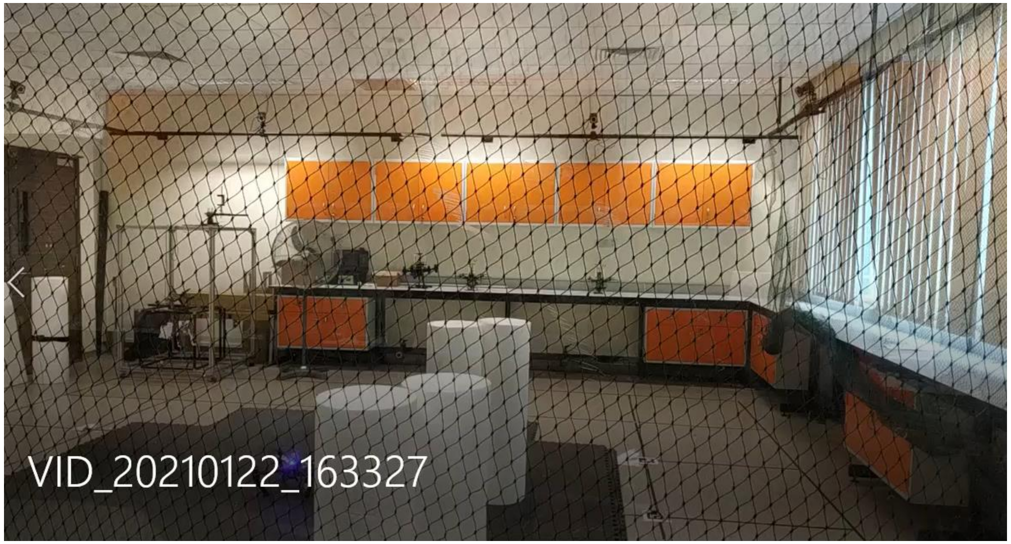
Sunan Huang

Temasek Laboratories

National University of Singapore

— Aug. 2020 —

Temasek Lab., NUS



Overview

- This talk will provide a broad overview of all components related to motion planning of autonomous systems.
- Later, two challenging topics will be discussed, i.e.,

 - A* search (next session)

- -Trajectory generation (after next session)

Learning Objectives

- Understand the motion planning concepts and developments in autonomous systems.
- Learning and mastering motion planning algorithms

Topics To Be Covered

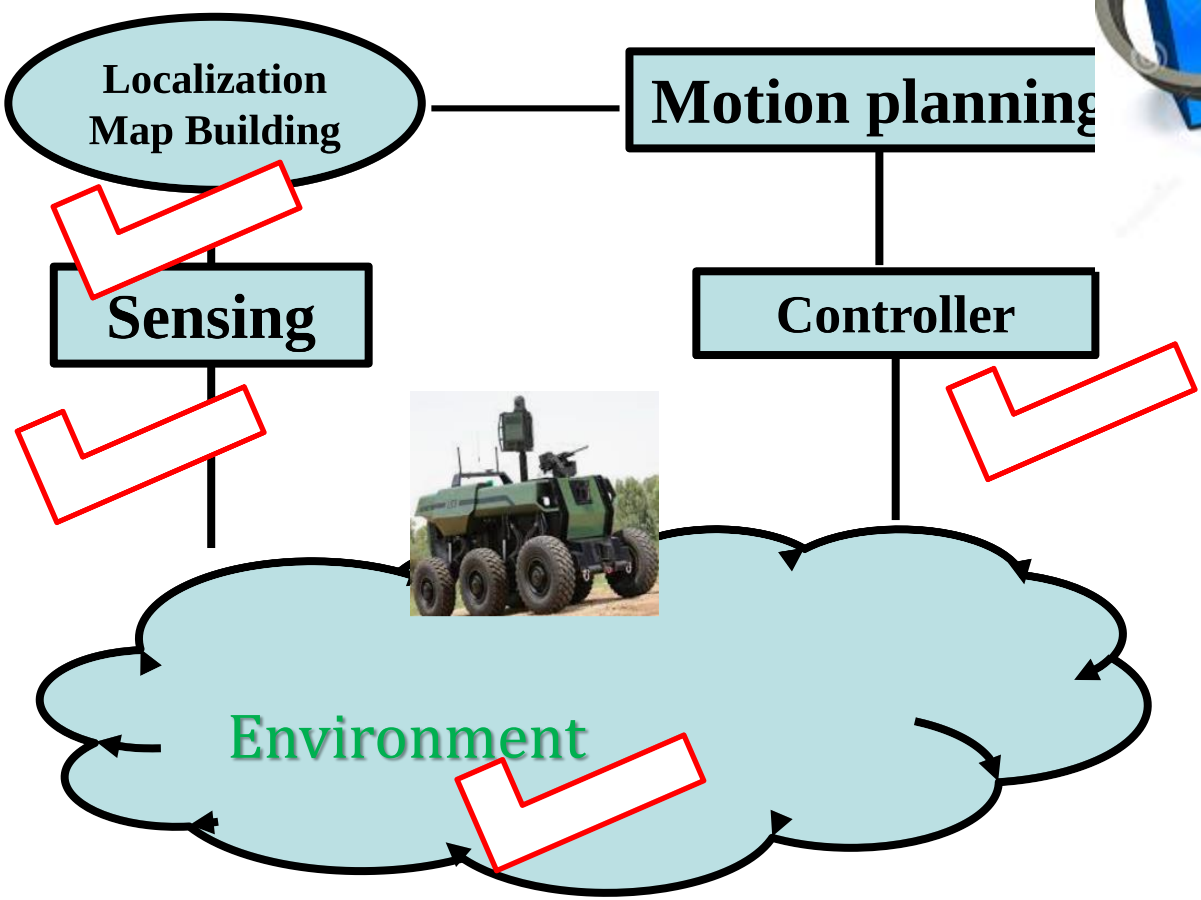
1. Motion planning concept
2. Historical development of motion planning in autonomous systems and overview of motion planning architectures
3. Several typical algorithms of motion planning

Grading

EE5110/EE6110/EE5062(6 questions)

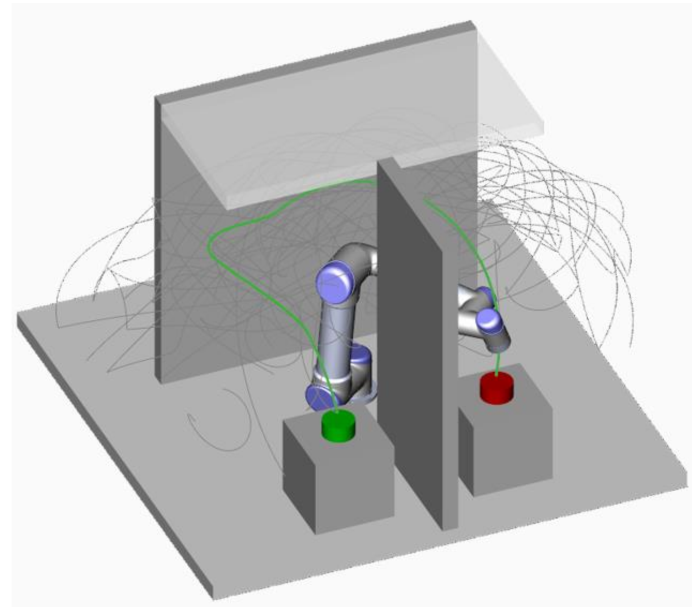
- **One report (homework) regarding motion planning concept (40%)**
- **One report (homework) regarding A* search (30%)**
- **One report (homework) regarding trajectory generation (30%)**
- **Only for EE6110,you have one option ---research project (programming)**

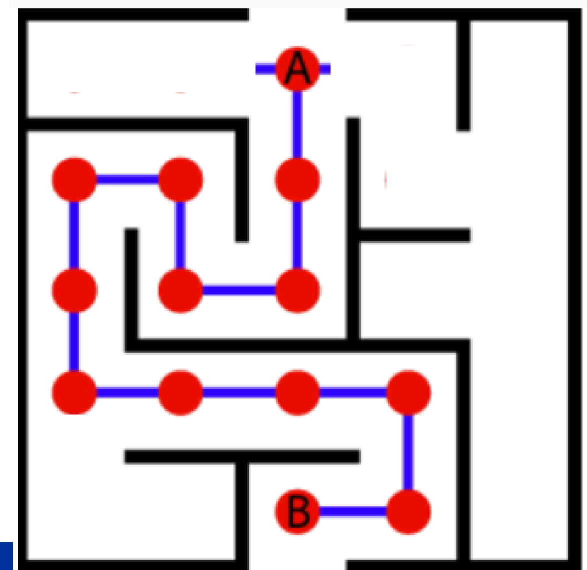
What we have learnt from Dr.Wang in autonomous systems



NUS National University of Singapore

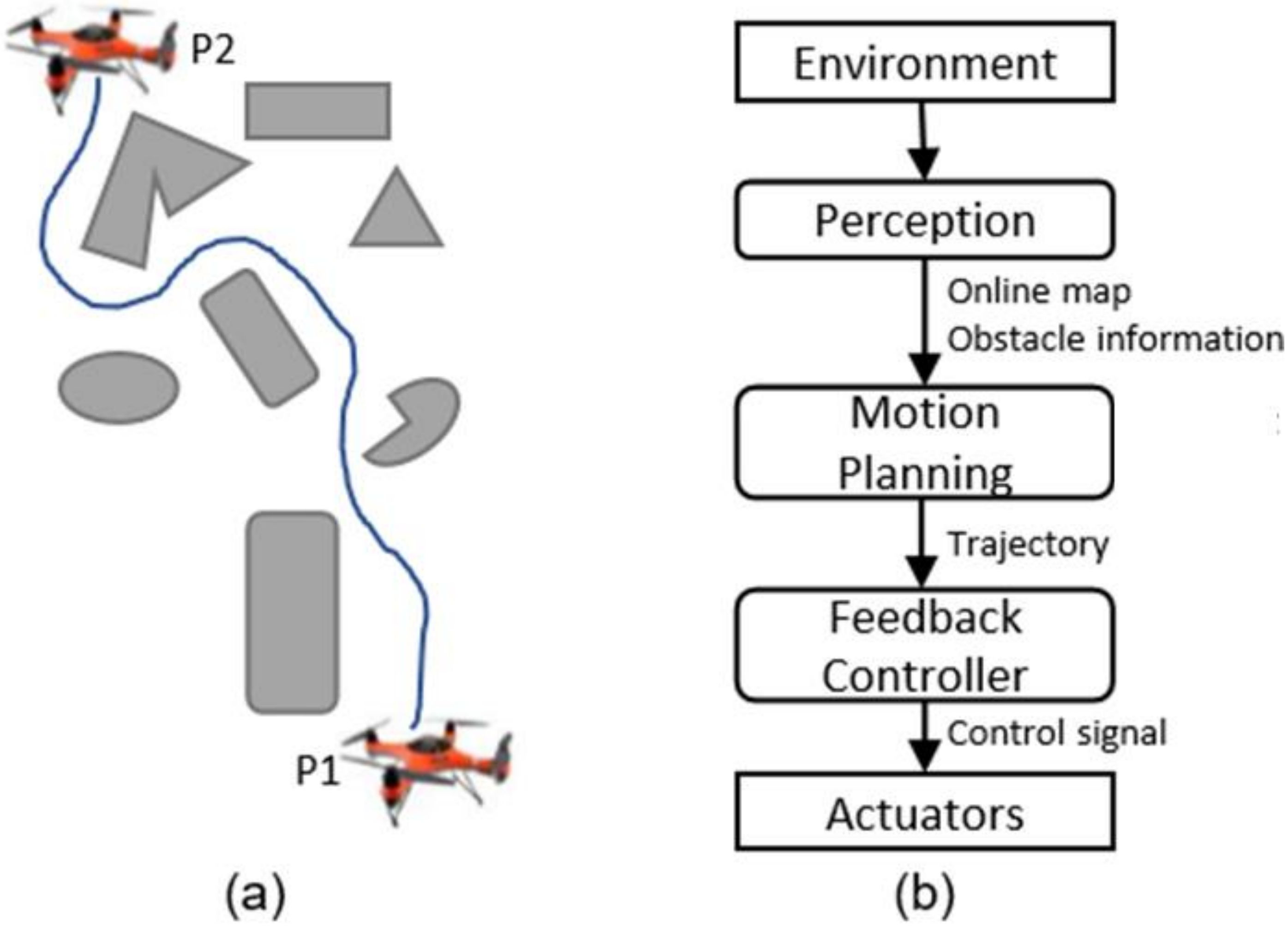
Is motion planning simple?





Where are we going?
How do we go anywhere?

1. The concept of autonomous system



One example: Use GPS for motion control

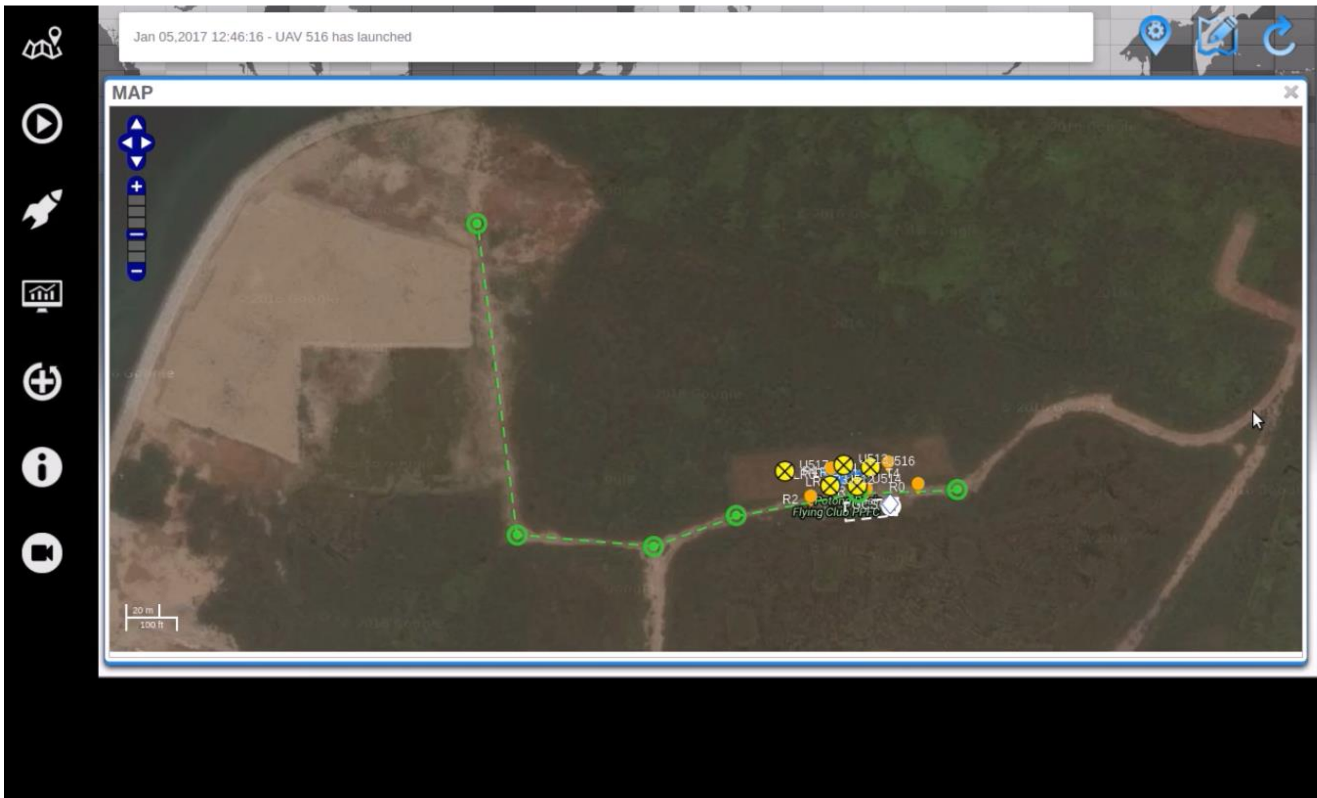
NUS
National University of Singapore

Motion planning

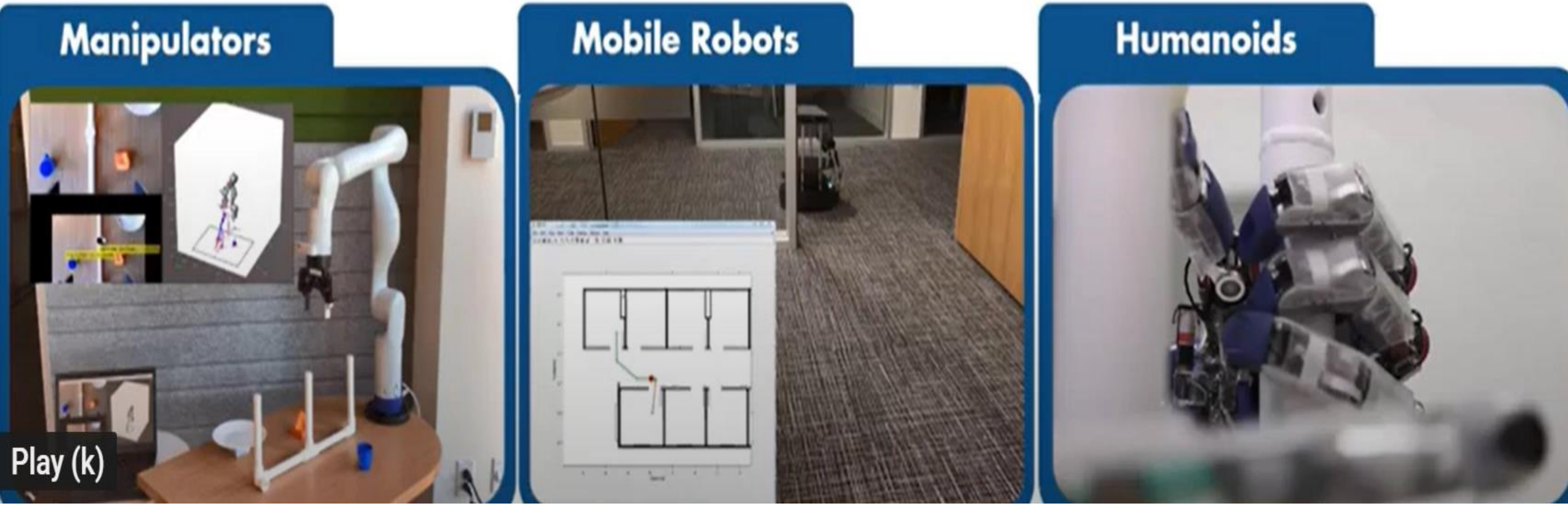
Why is motion planning important in autonomous systems?

Some videos…

1.One UAV leader and four UAV followers do a formation, performing motion planning and collision avoidance . We have one mobile ground control station (GCS).



2.Robotic system



https://www.youtube.com/watch?v=BwNw_-KXjbM

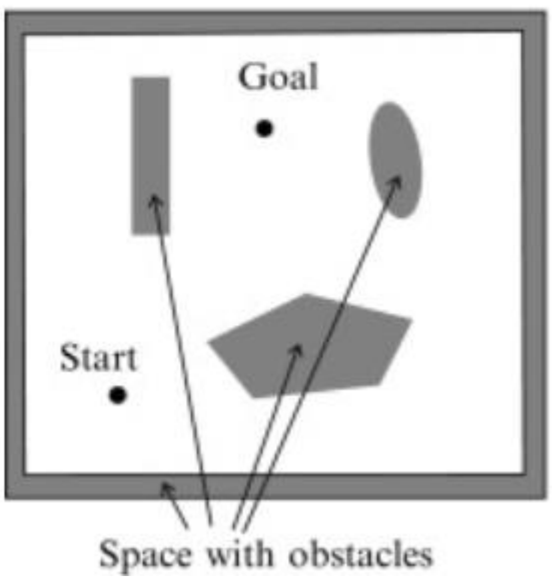
Motion planning is fundamental in autonomous control of an unmanned system.

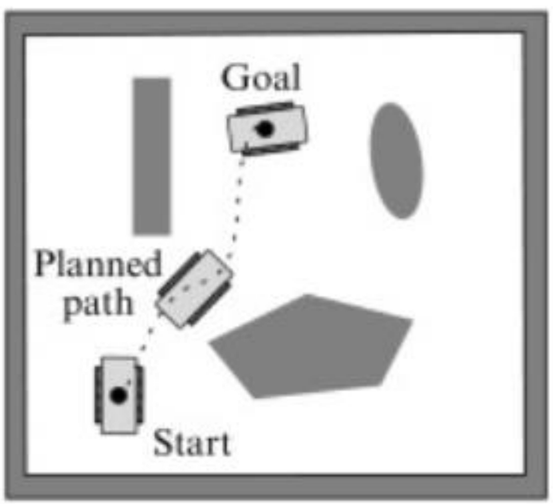
2.Evolution of motion planning

Year	Works and development
2000's	Motion planning of multi vehicles teams in complex environments
1990's	Local motion planning of multi vehicles
1985's	Path planning with considering uncertain dynamics and dynamical environments
1970's	Classical path planning. Mainly focused on on dealing with static obstacles

1970's Classical path planning

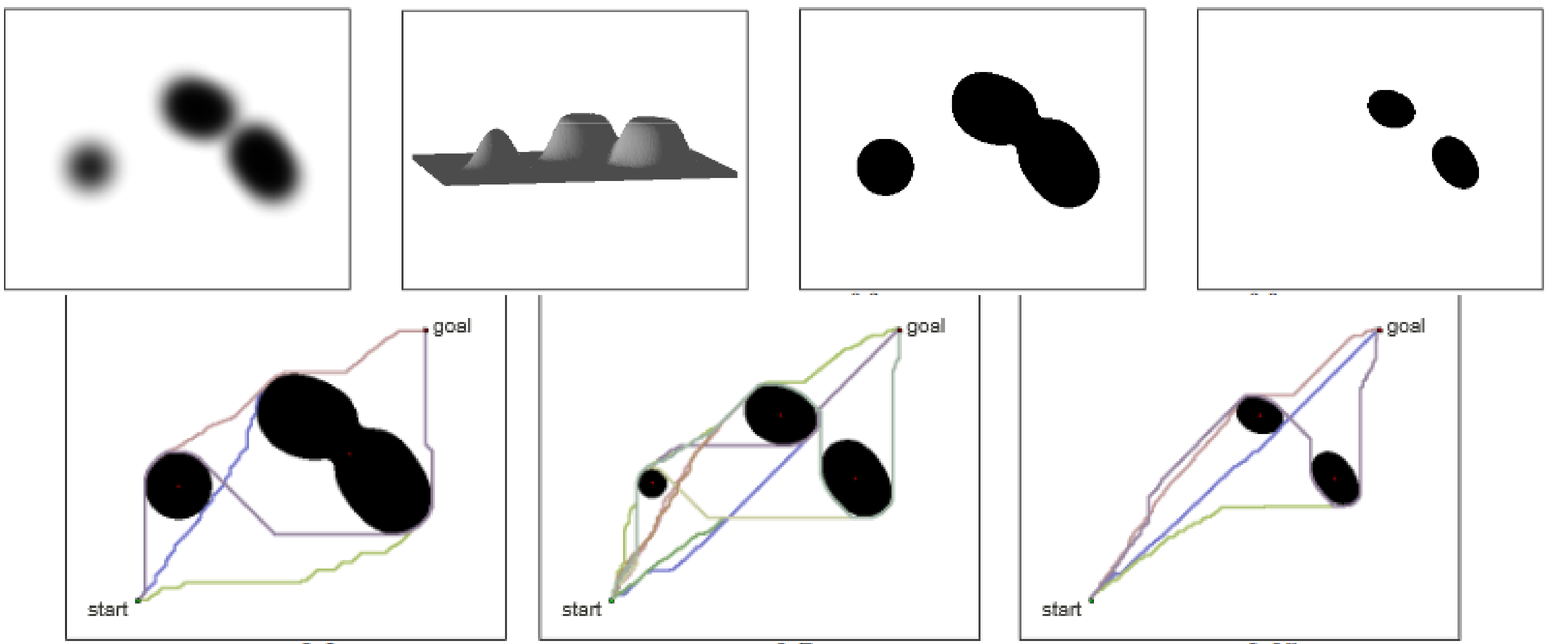
Path planning for vehicle in known environments typically requires a decomposition of the vehicle's *free space*, i.e., that part of the environment in which the vehicle does not collide with an obstacle. It is decomposed into a number of areas within which the vehicle can move freely without danger of collision and which are connected if the vehicle can execute a move between these areas without collision.





1985's Path planning in uncertain environment

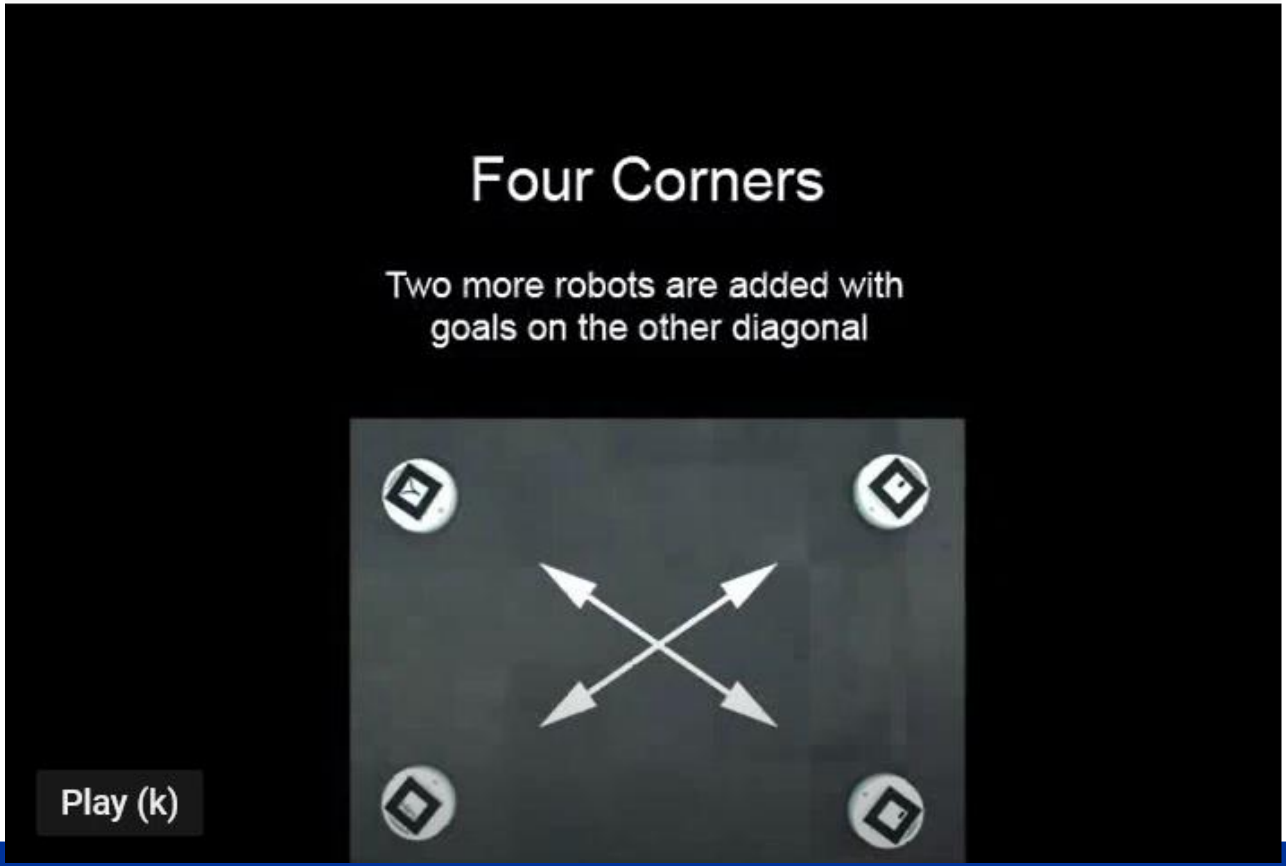
Uncertainty naturally arises in unknown environments, in the presence of process and/or observation noise, and unknown dynamics of the environment. For vehicle, the most common representation of the environment is an occupancy grid in which each cell is assigned a value of probability of occupancy.



(a) $U^{0.2}$.   (b) $U^{0.7}$.   (c) $U^{0.95}$.

1990's Local planning of multi-vehicles

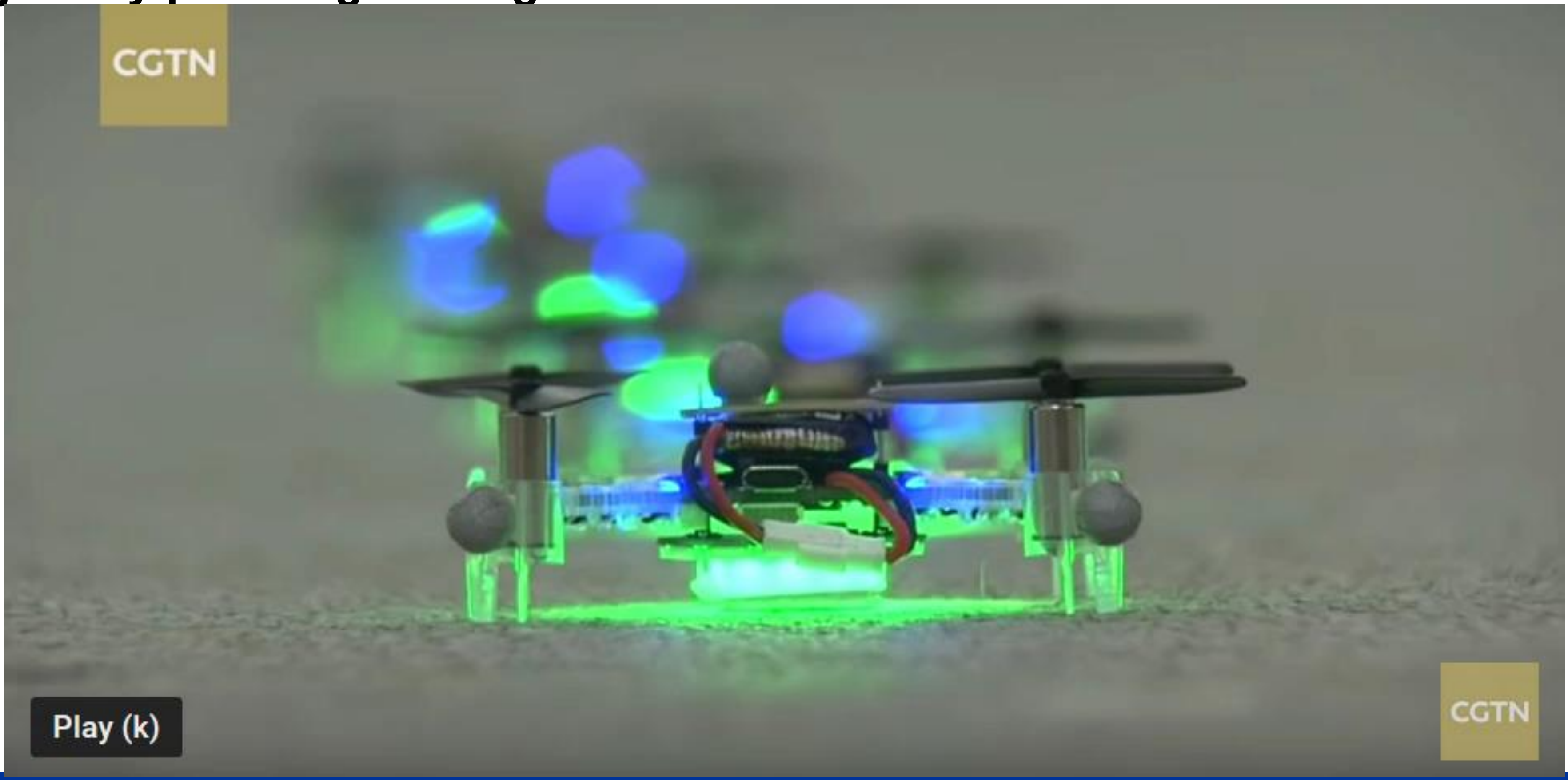
Motion planning in a dynamical environment. It involves the computation of a collision-free path from start to goal for each vehicle without colliding with other vehicles.



2000's Motion planning of multi-UAV teams

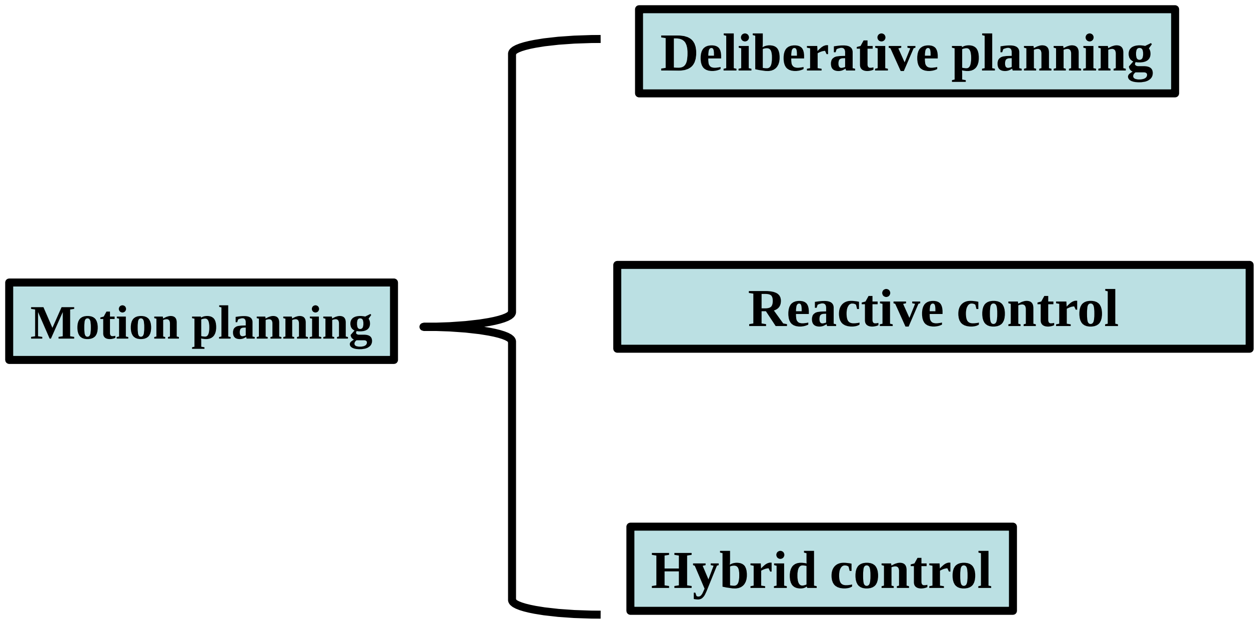
This planning focus on particular mechanisms for coordinating

among multi-UAVs without collision. Most of the works proposed a safe trajectory planning for large UAV teams.

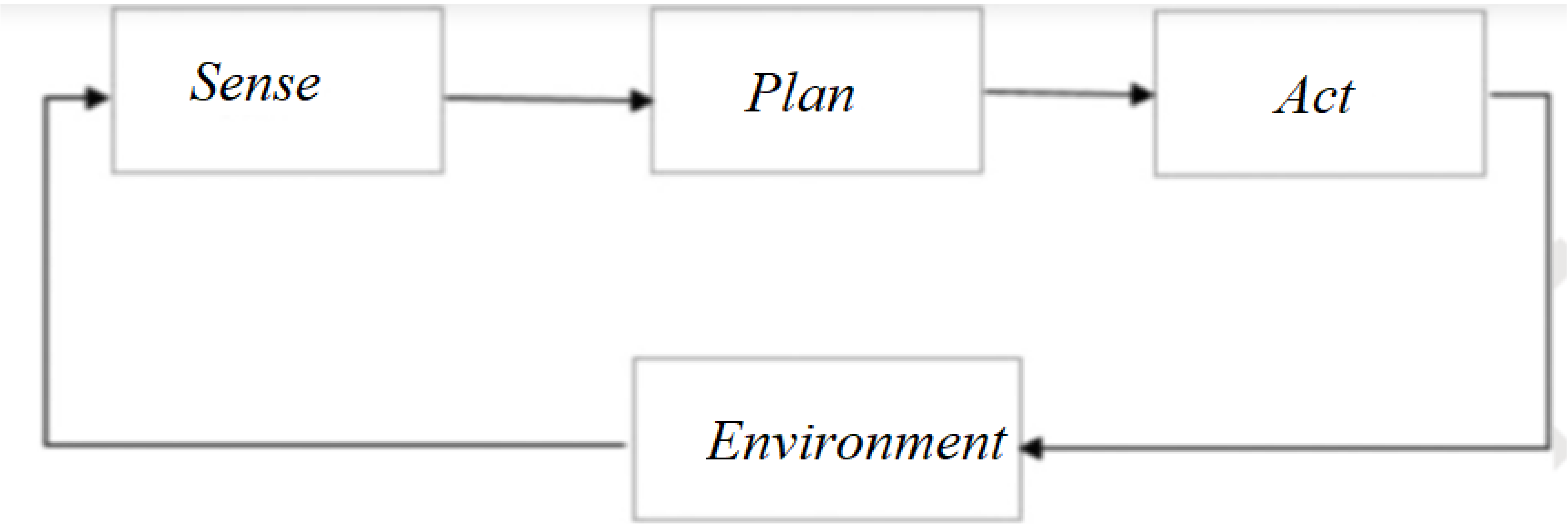


3. Overview of motion planning architectures

The motion planning architecture defines how the works of generating actions from perceiving the world around us are organized. Basically, motion planning can be grouped into three classes: deliberative planning architecture, reactive architecture and hybrid architecture.



3.1 Deliberative planning architecture

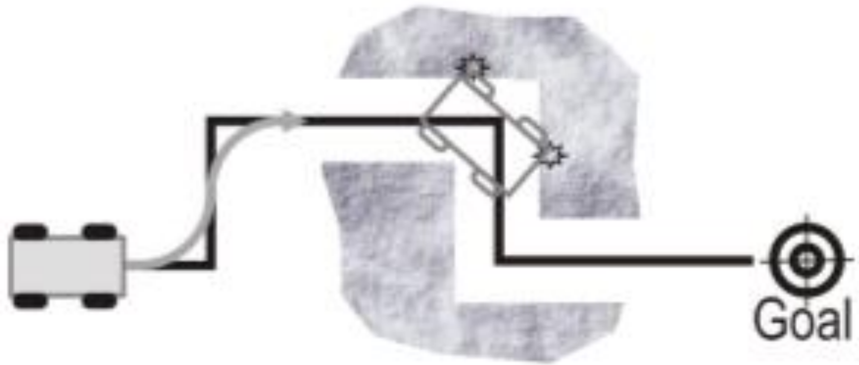


The deliberative methods

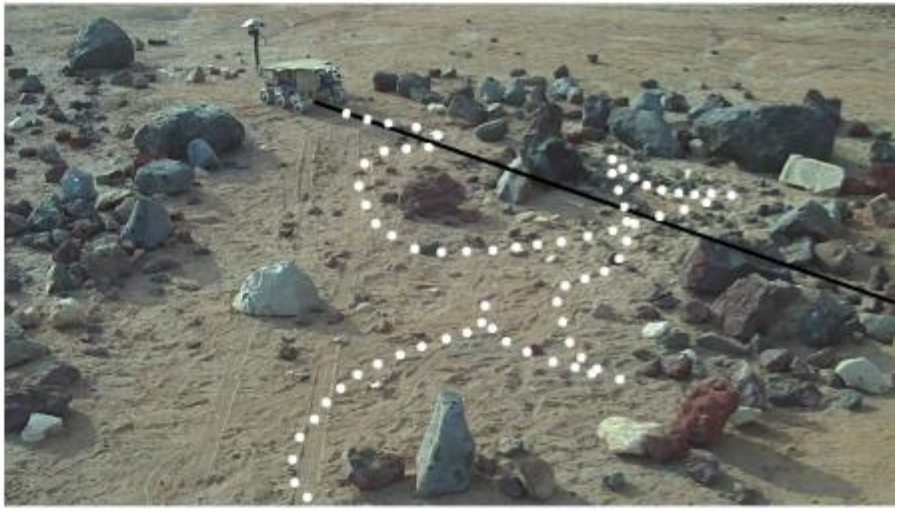
- **Deliberative control relies on reasoning on state information**
- Compute the different possibilities and all combinations of them
- Take a long time to think and produce a result. The results are near optimal.

One example:

FIDO rover navigates autonomously using the motion planner among dense obstacles in the Mars Yard at the California Institute of Technology



A simple example of a motion planning problem, where a car-like robot that attempts to follow the infeasible path (black line) experiences a failure.



Pros

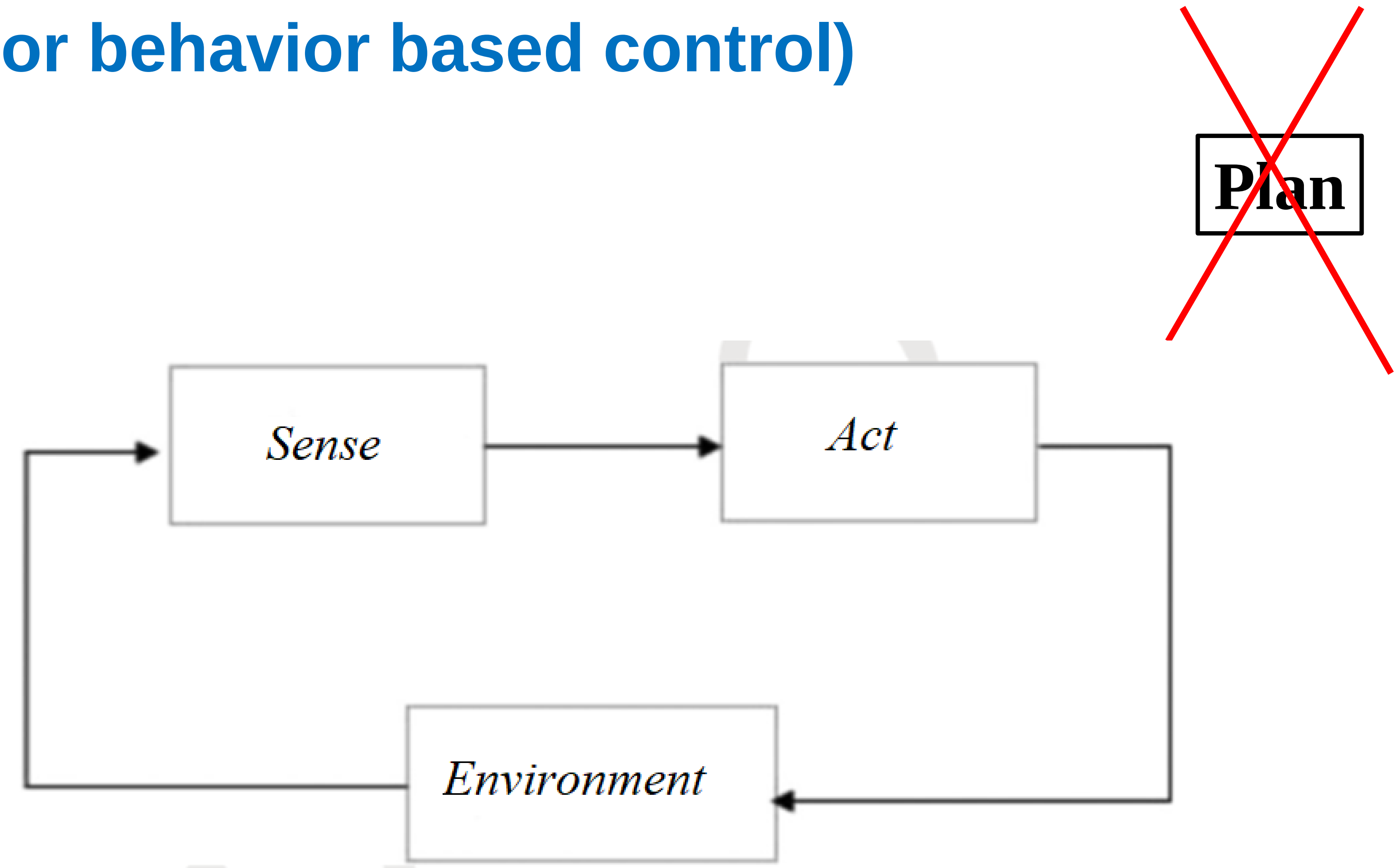
- Can maintain an effective goal-directed behavior despite the path perturbations caused by reacting to changes in the environment such as the motions of dynamic obstacles.
- Optimal solution

Cons

- Expensive computational load

3.2 Reactive control architecture

(or behavior based control)



Reactive control:

- Direct coupling of perception and control action.
- Aim is timely vehicle response in dynamic and unstructured worlds.
- Typically in the context of motor behaviors.

One example: reactive control

Pros:

- It is shown to effectively produce robust performance in complex and dynamic domains

Cons:

- Cannot select best behaviors to accomplish a task

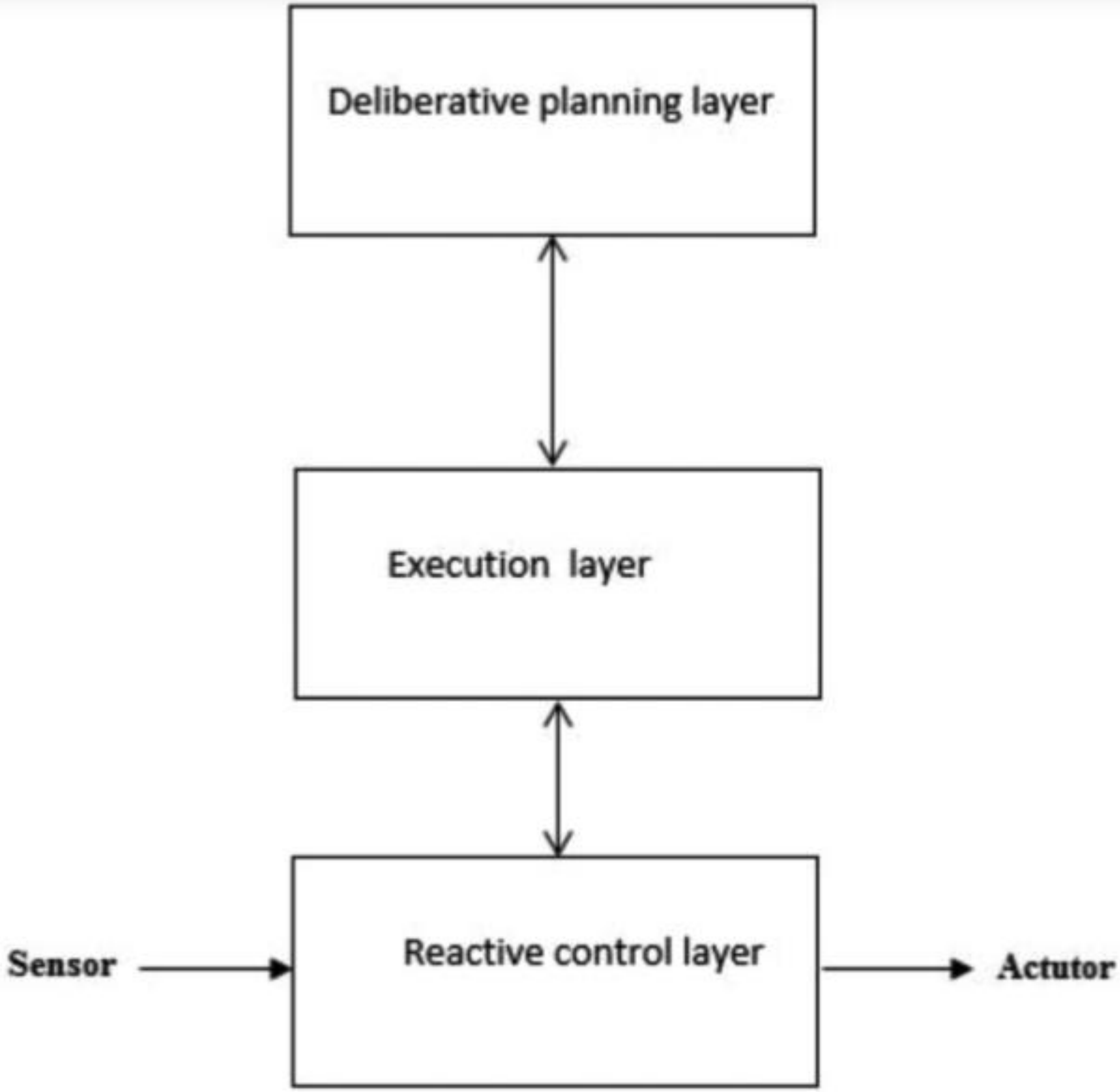
NUS National University of Singapore

Comparison between deliberative planning and reactive control

Deliberative	Reactive
Speed of response	
Predictive capabilities, completeness of world models	
Slow response High-level intelligence	Real-time response Low-level intelligence Simple (analogue) computation

http://www.informatik.uni-leipzig.de/~der/Vorlesungen/ROBOTIK/Deliberative%20vs_%20reactive%20control.htm

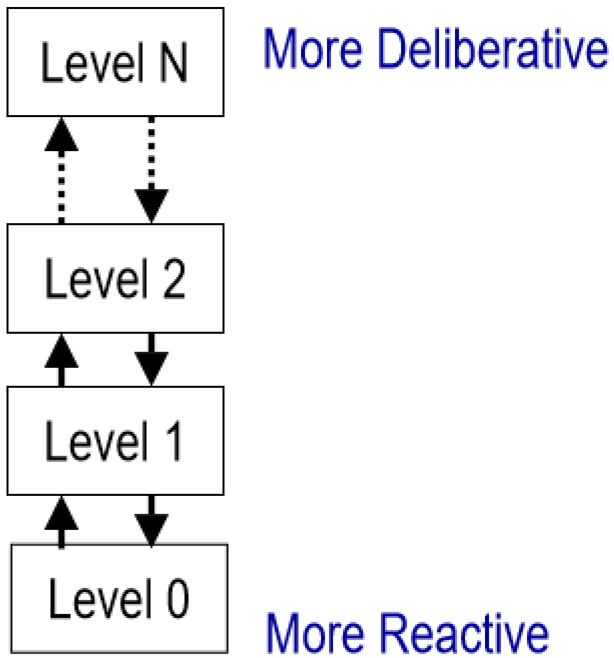
3.2 Hybrid architecture



Three ways for hybrid structure

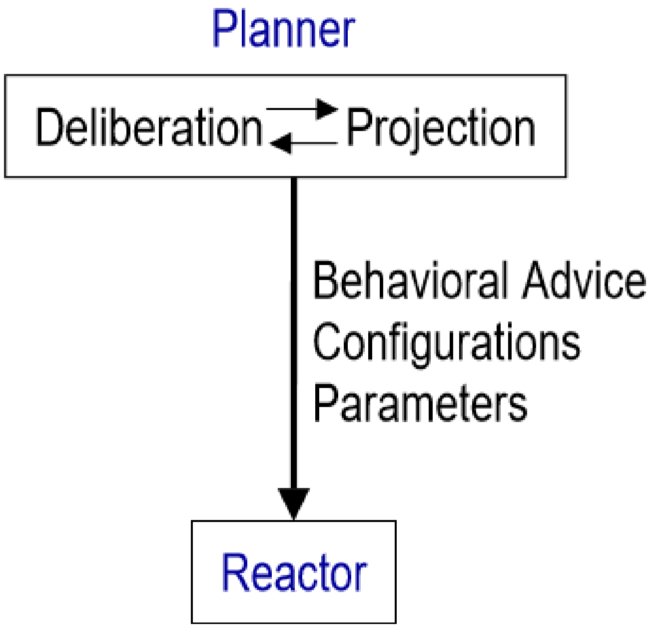
1.Hierarchical integration:

- **Deliberative planning and reactive execution are involved with different activities, time scales, and spatial scope**
- **Planning vs. reacting depends on situation at hand**



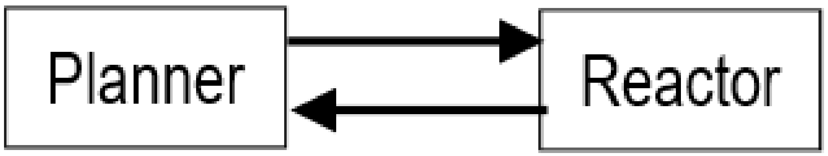
2.Planning to guide reaction:

- **Planning configures and sets parameters for the reactive control system**
- **Execution occurs only due to reactive system**
- **Planning occurs prior to and concurrent to reactive system**

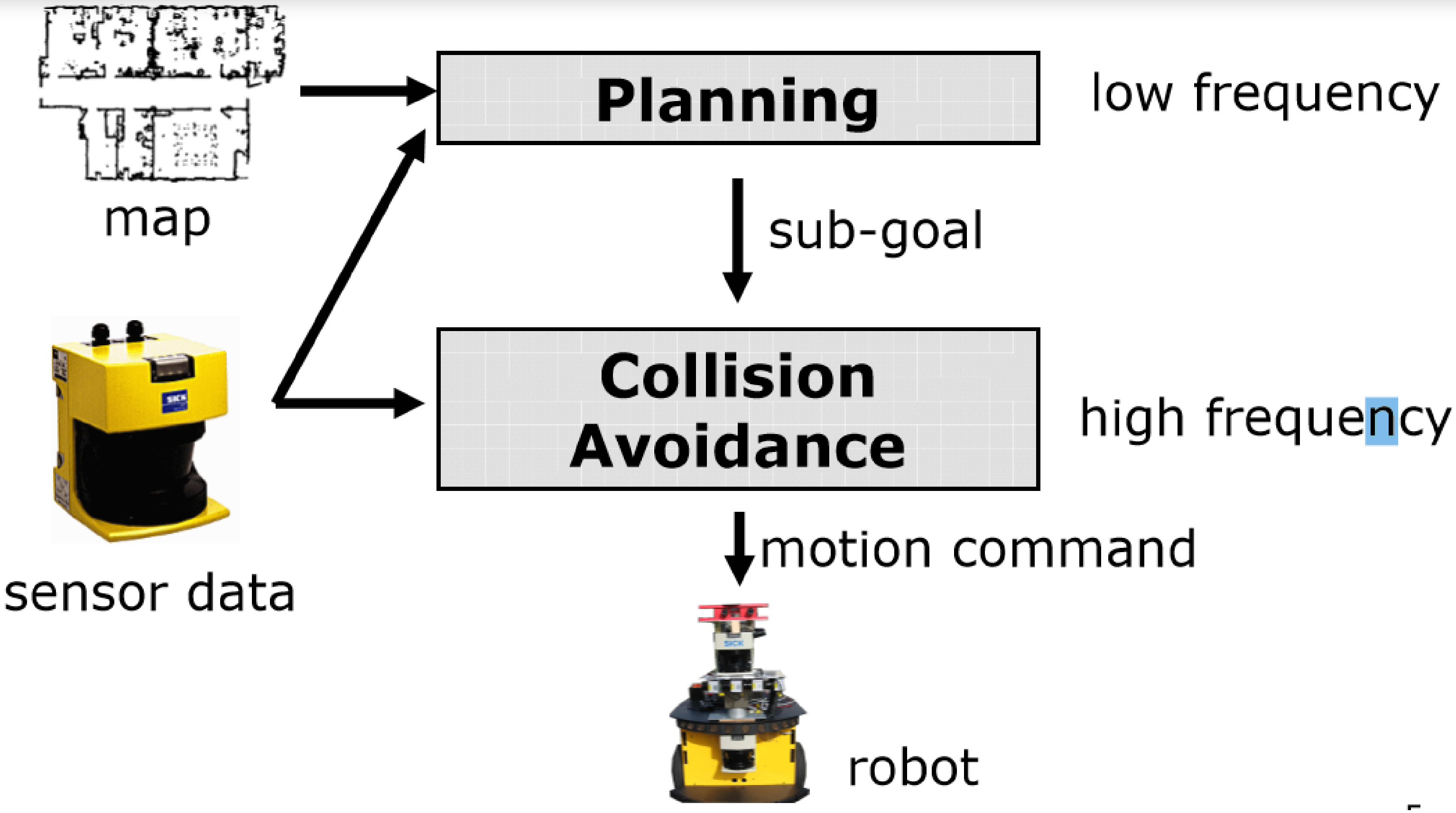


3.Coupled planning-reacting:

- **Planning and reacting are concurrent activities, each guiding the other**



An example of hybrid system

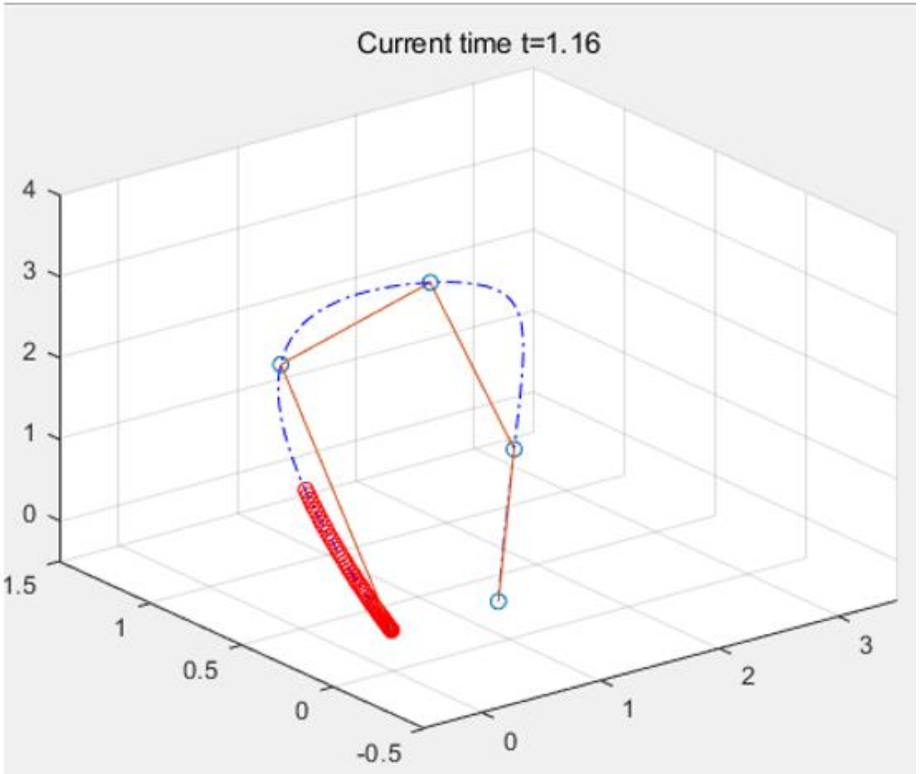


4.Fundamentals of Motion planning

The essential requirement of a motion planning task is to provide effective guidance for an autonomous system to reach the destination without any collision.

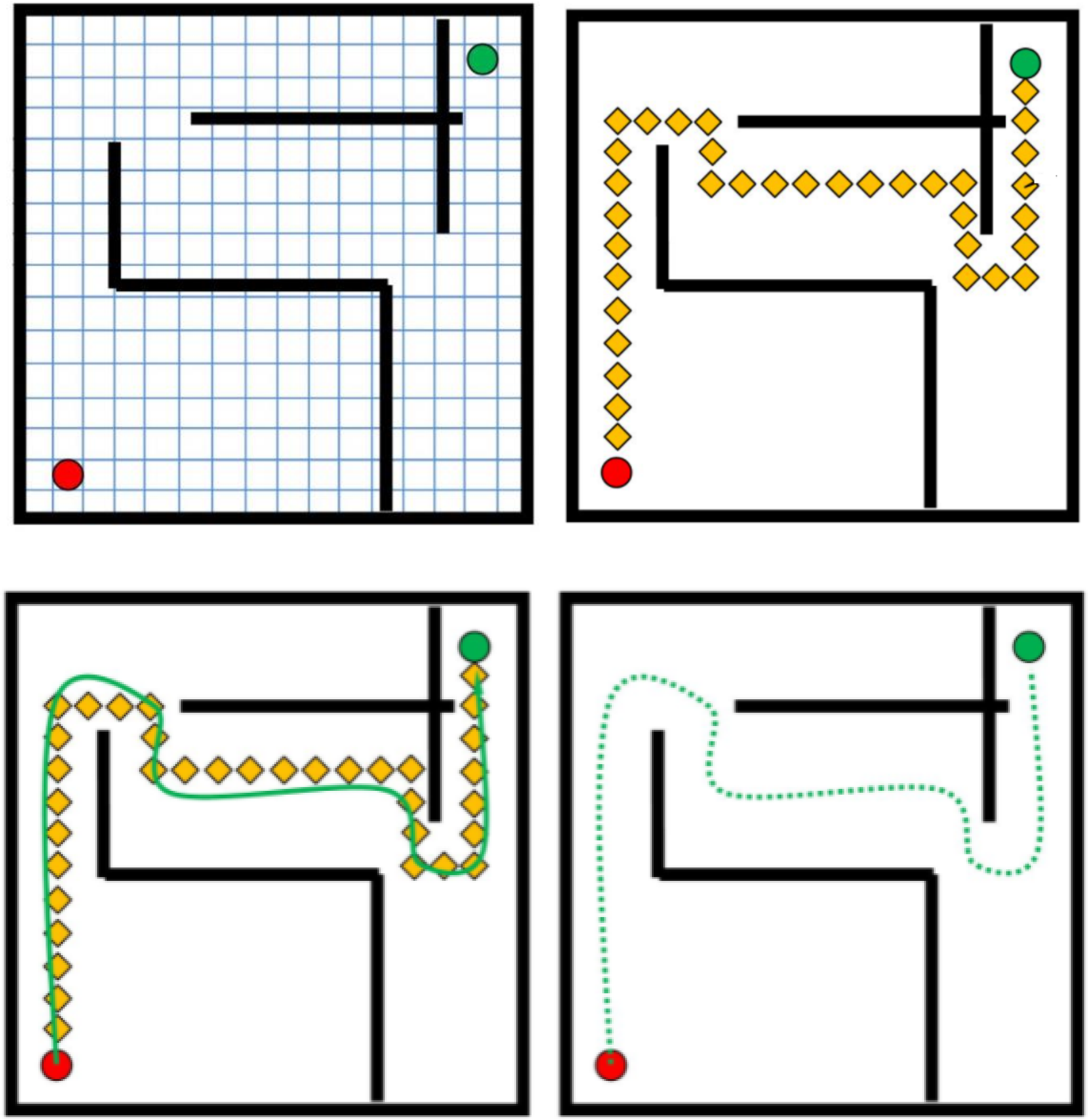
Motion planning involves

- C-Space
- Path planning
- Trajectory planning.



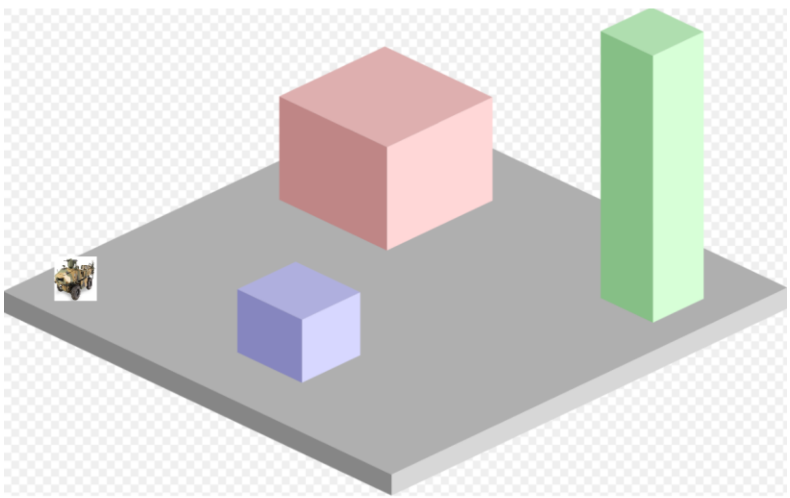
NUS National University of Singapore

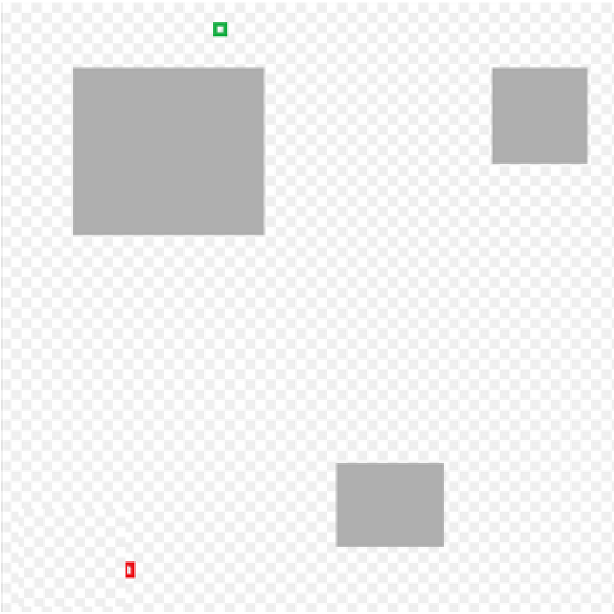
One example



Work space and configuration space (C-space)

Motion planning problem needs to connect a start configuration S and a goal configuration G, while avoiding collision with known obstacles. The vehicle and obstacle geometry are described in a 2D or 3D **workspace**, while the motion is represented as a path in configuration space (C-space).





White = C_{free}, gray = C_{obs}.

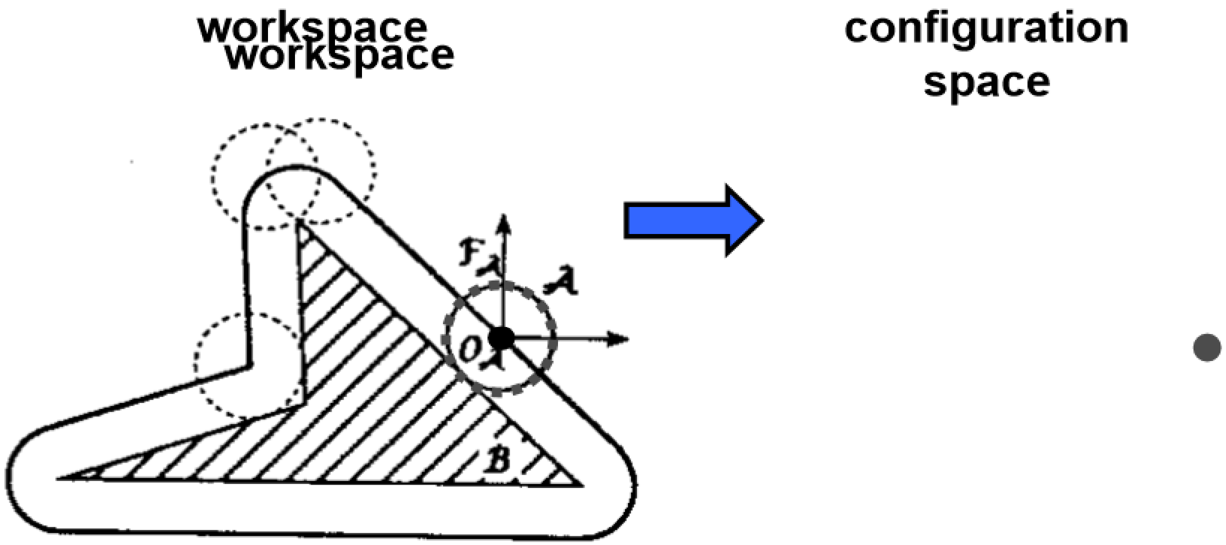
C-space: C = $C\_free \cup C\_obstacle$

- **Robot is referred to a point**

- $C_{Obstacle}$ **is the set of all configurations in which the robot collides with an obstacle.**

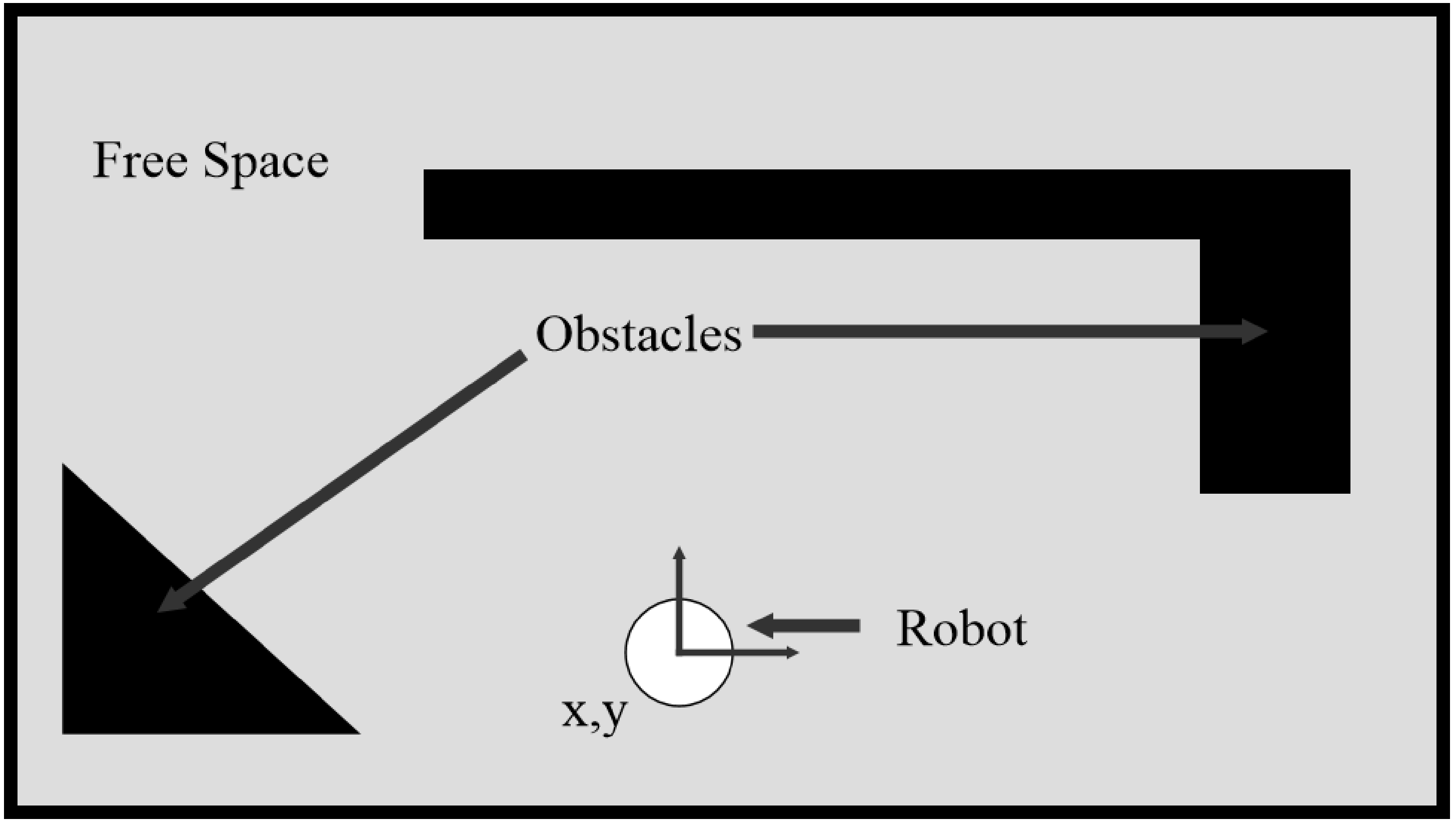
How do we compute *C_obstacle*?

For circular robot, expand all obstacles by the radius of the robot. We don't discuss other forms of robots here.

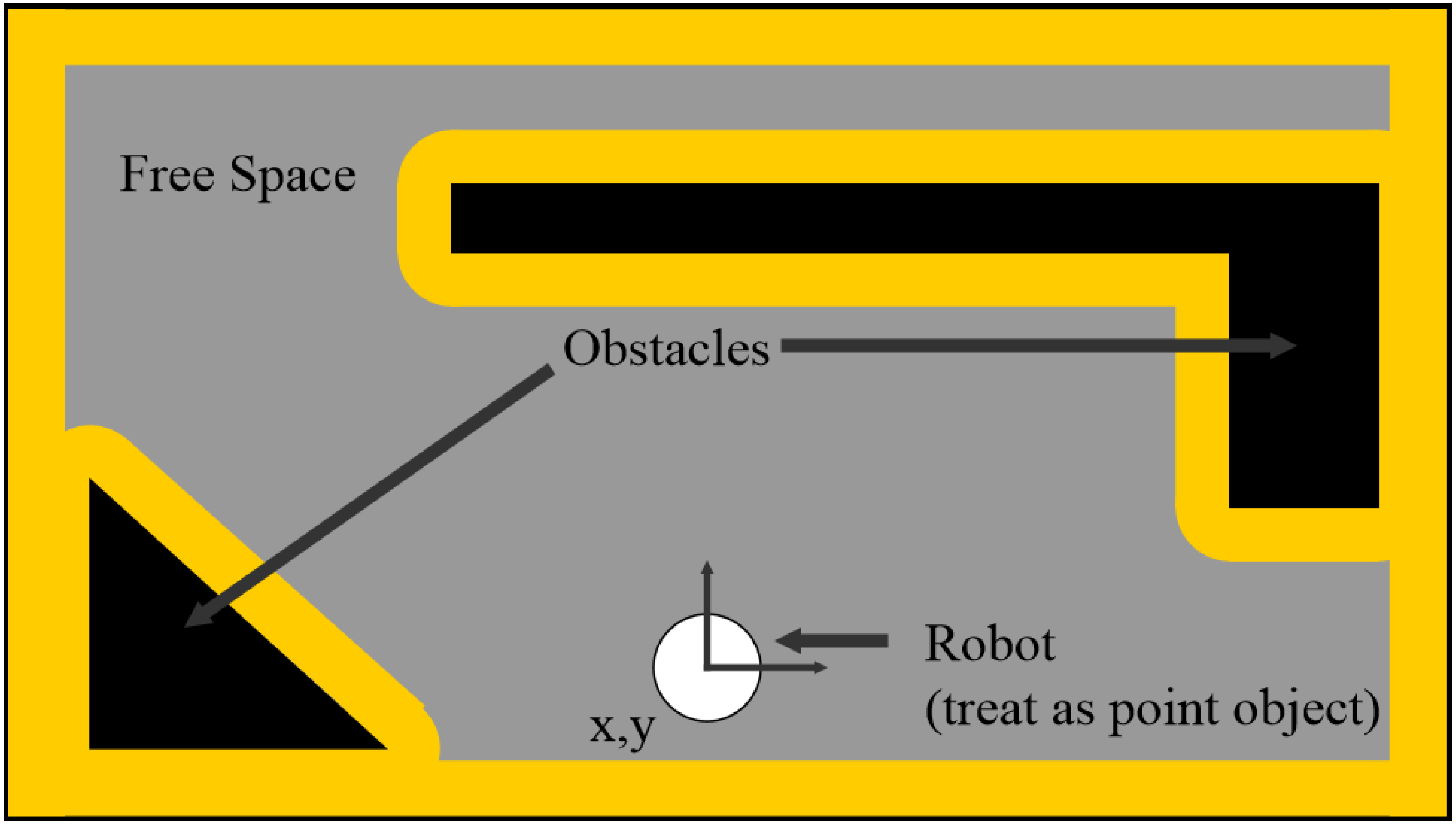


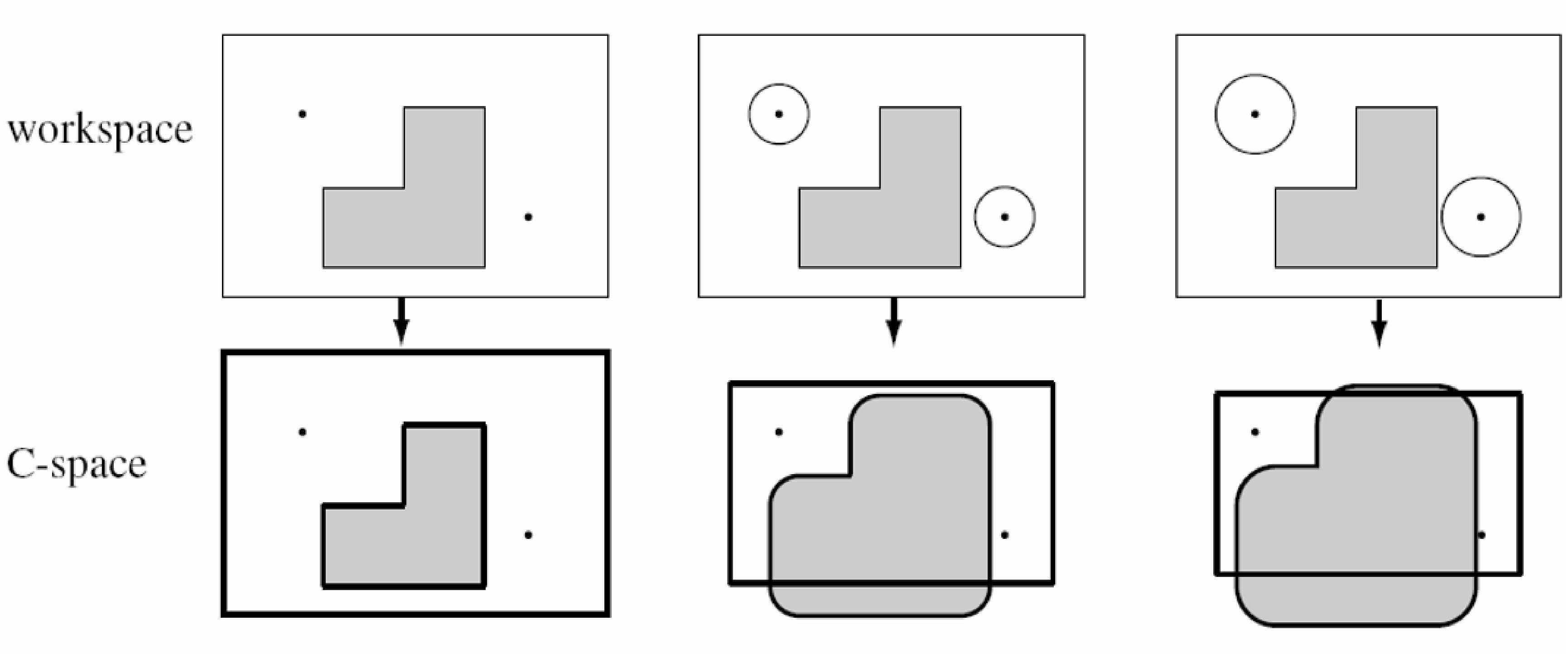
NUS National University of Singapore

Workspace

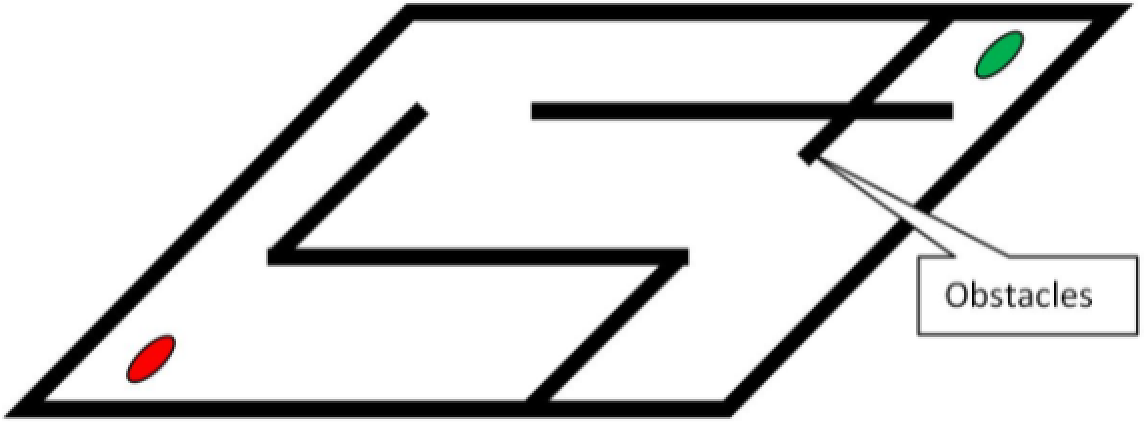


Configuration Space: Accommodate Robot Size




workspace
C-space

2D C-space (grid-based method):

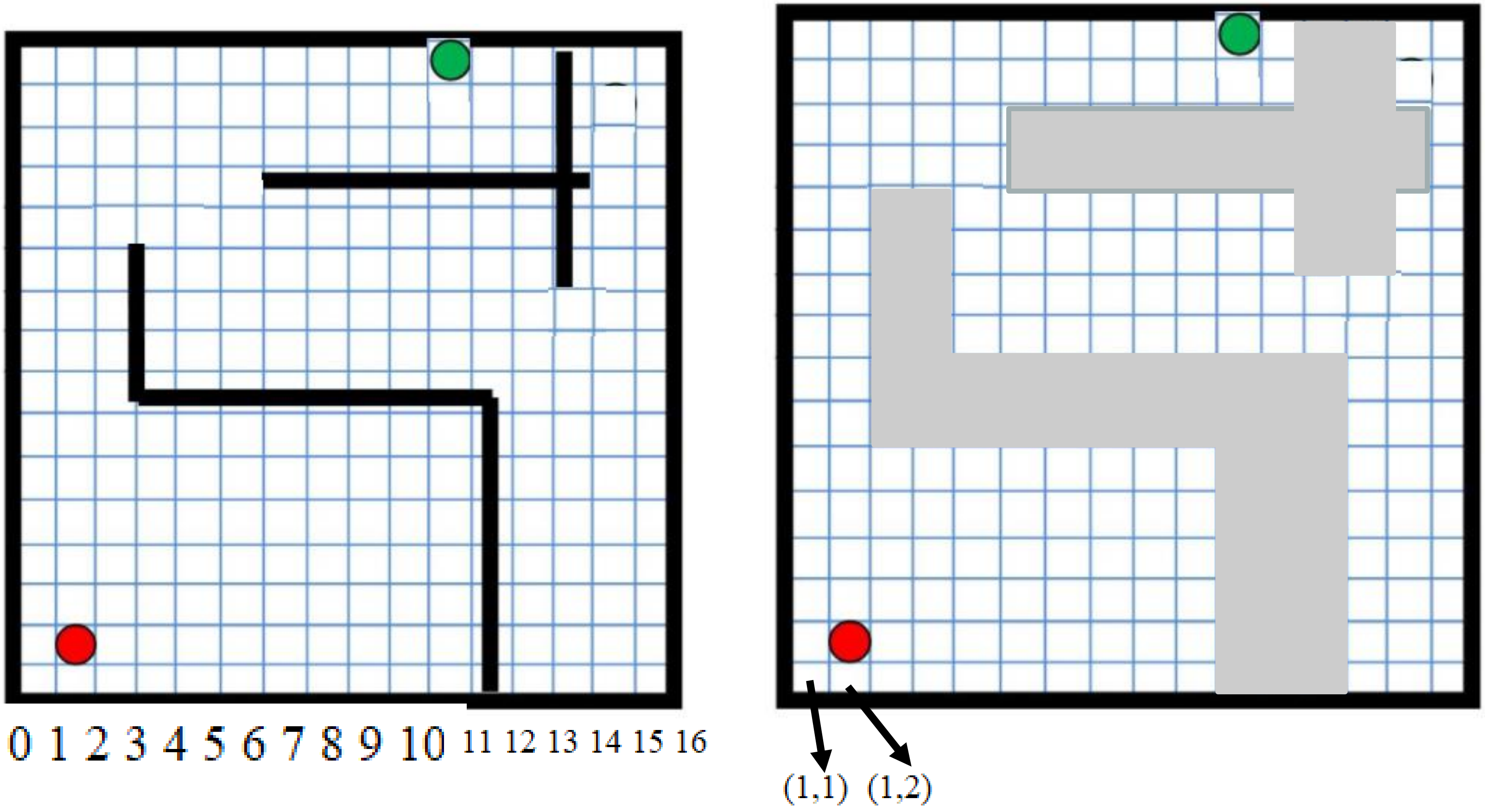


1.<u>Discretize the environment</u>, i.e. transform the continuous environment into a discrete one.

2.Matrix *Mmxn* *(16x16).*

3.For each element, M(i,j)=0 if empty; 1 otherwise.

4.Expand obstacles with robot's radius=0.5



Obstacles are expressed in C-space

- $$C_{obstacle1} = \{M(11,1), M(12,1), M(13,1),$$

$$M(11,2), M(12,2), M(13,2),$$

$$M(11,3), M(12,3), M(13,3),$$

$$M(11,4), M(12,4), M(13,4),$$

$$M(11,5), M(12,5), M(13,5),$$

$$M(11,6), M(12,6), M(13,6),$$

$$M(11,7), M(12,7), M(13,7),$$

$$M(11,8), M(12,8), M(13,8),$$

$$M(3,7), M(4,7), M(5,7), M(6,7), M(7,7), M(8,7), M(9,7), M(10,7),$$

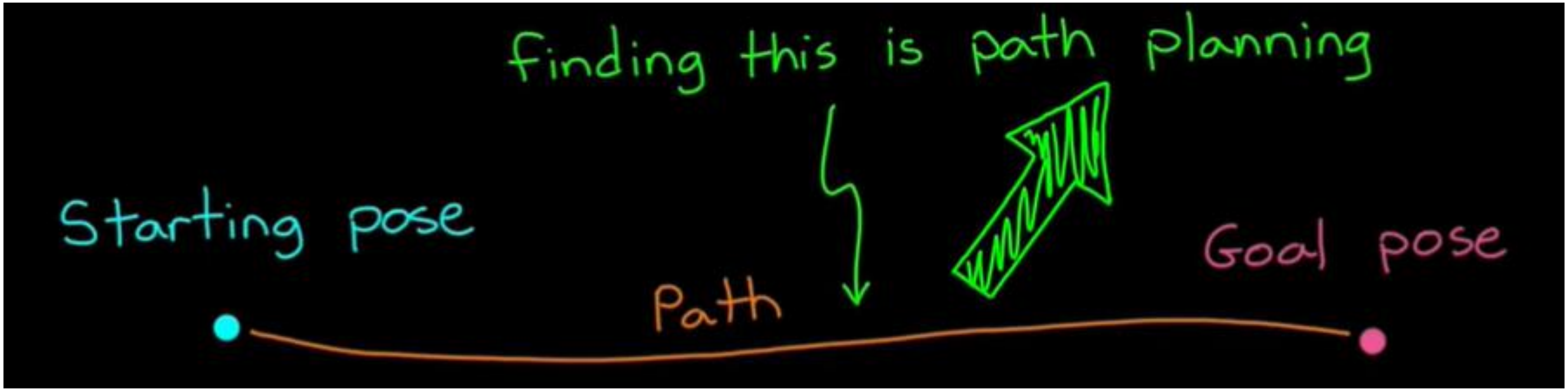
$$M(3,8), M(4,8), M(5,8), M(6,8), M(7,8), M(8,8), M(9,8), M(10,8),$$

$$M(3,9), M(4,9), M(3,10), M(4,10), M(3,11), M(4,11),$$

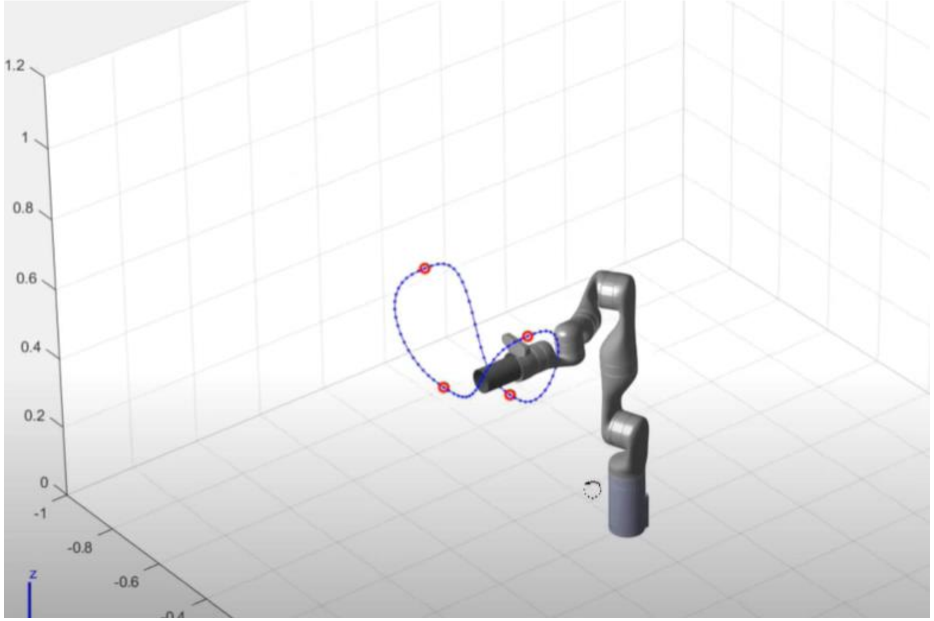
$$M(3,12), M(4,12)\}$$

Can you use this way to express Free Space in C-space?

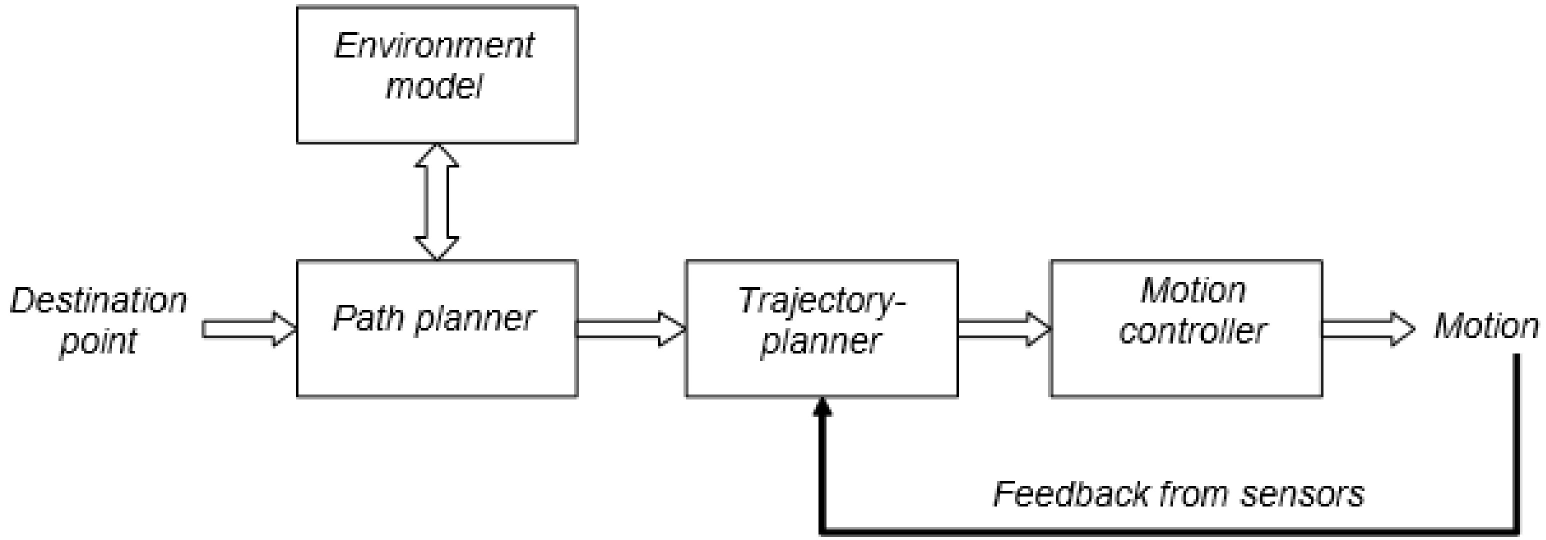
Path planning aims to find a purely geometric path connecting two states (or configurations in robotic terminology) by focusing only on collision avoidance while ignoring the vehicle's dynamics, the motion duration, and other constraints like the physically limited actuation.



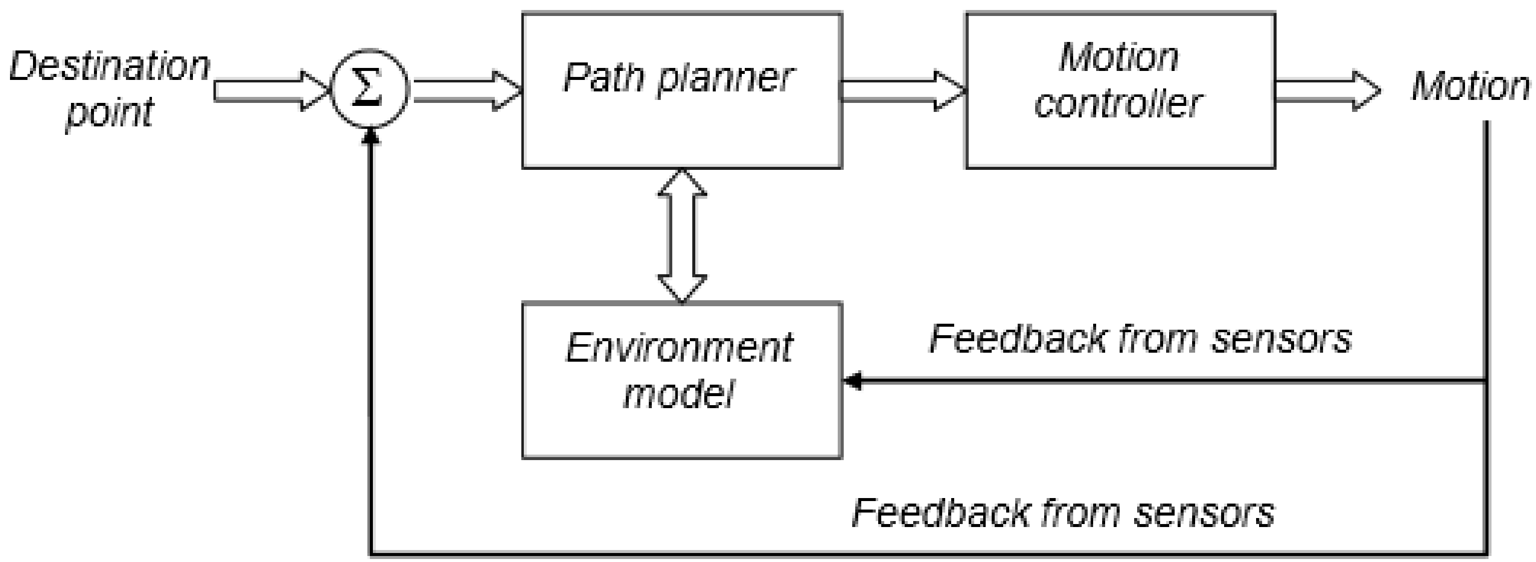
Trajectory planning must take the vehicle's dynamics into consideration, called differential constraints, as well as optionally more constraints on velocity, acceleration, and control inputs. Besides, it is a common practice to guide trajectory planning with a custom performance index.



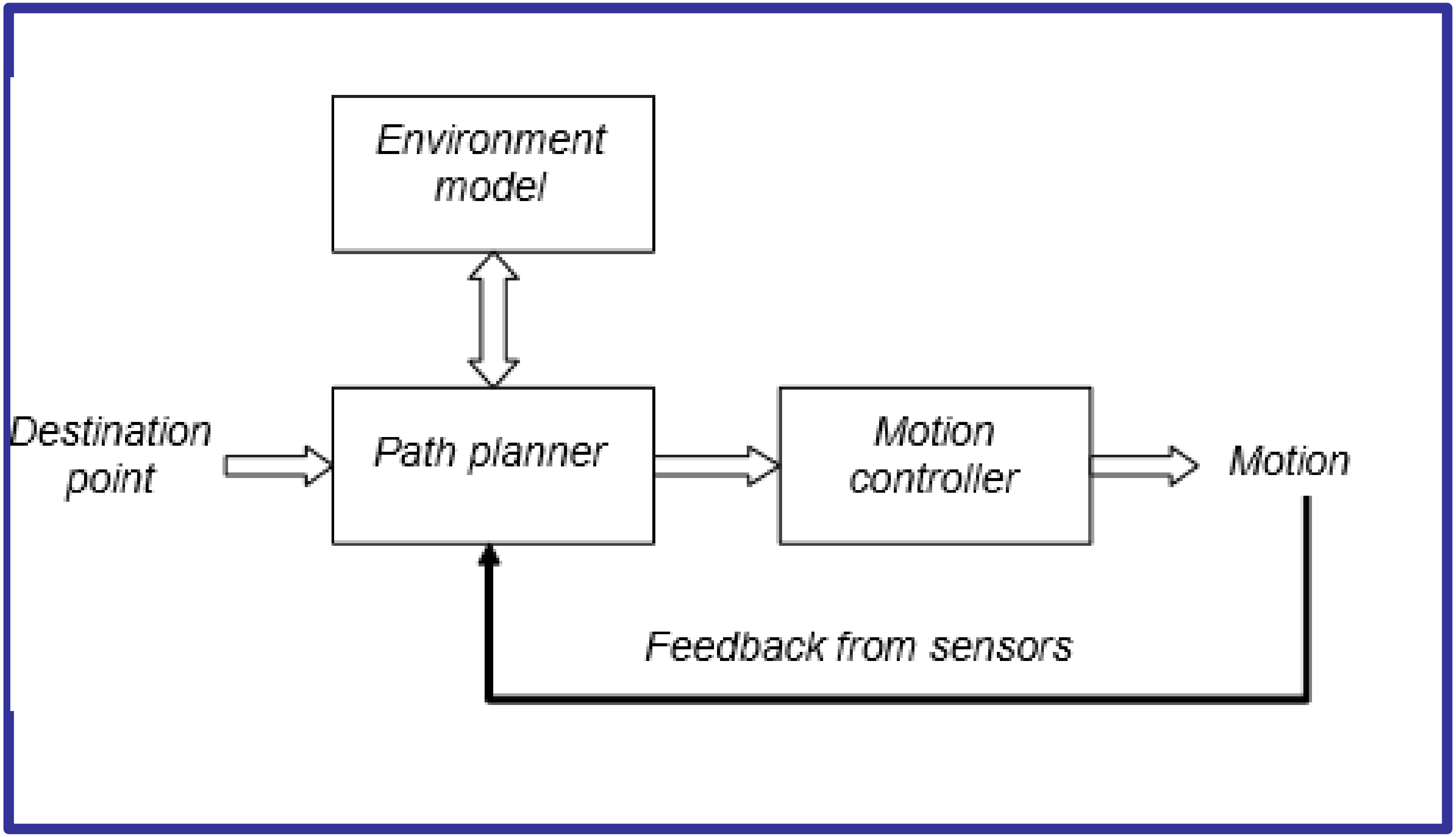
How do we combine both in autonomous systems?



Hierarchical path planning



Dynamical path planning


Environment model
Destination point
Path planner
Motion controller
Motion
Feedback from sensors


DUKE ROBOTICS
Robot Motion Planning on a Chip
Sean Murray, Will Floyd-Jones, Ying Qi, Dan Sorin, George Konidaris

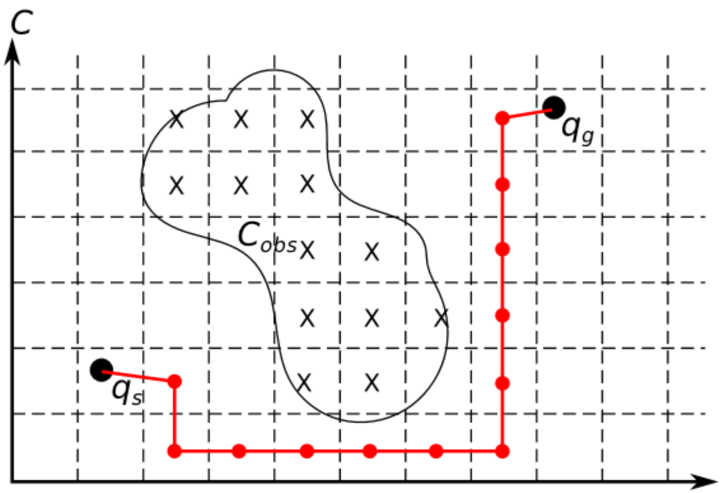
5. Typical motion planning algorithms

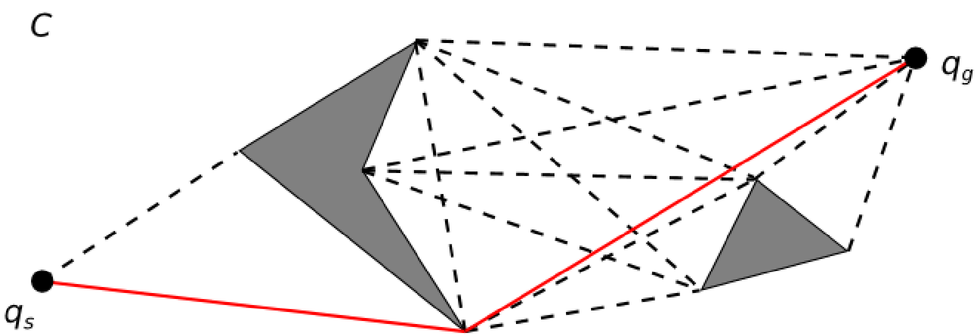
- Graph search
- Conflict detection and resolution approach
- Model predictive control
- Potential field function
- Geometric guidance

1.Graph search

- This method is to plan a safe path for a polyhedral object moving among known environment
- The algorithm requires the obstacles which are represented by polygons so that they can be handled by the graph-based optimization approach.
- The task is to plan a path which avoids all obstacle polygons. This can be accomplished by searching a path through a graph connecting vertices of the obstacle polygons

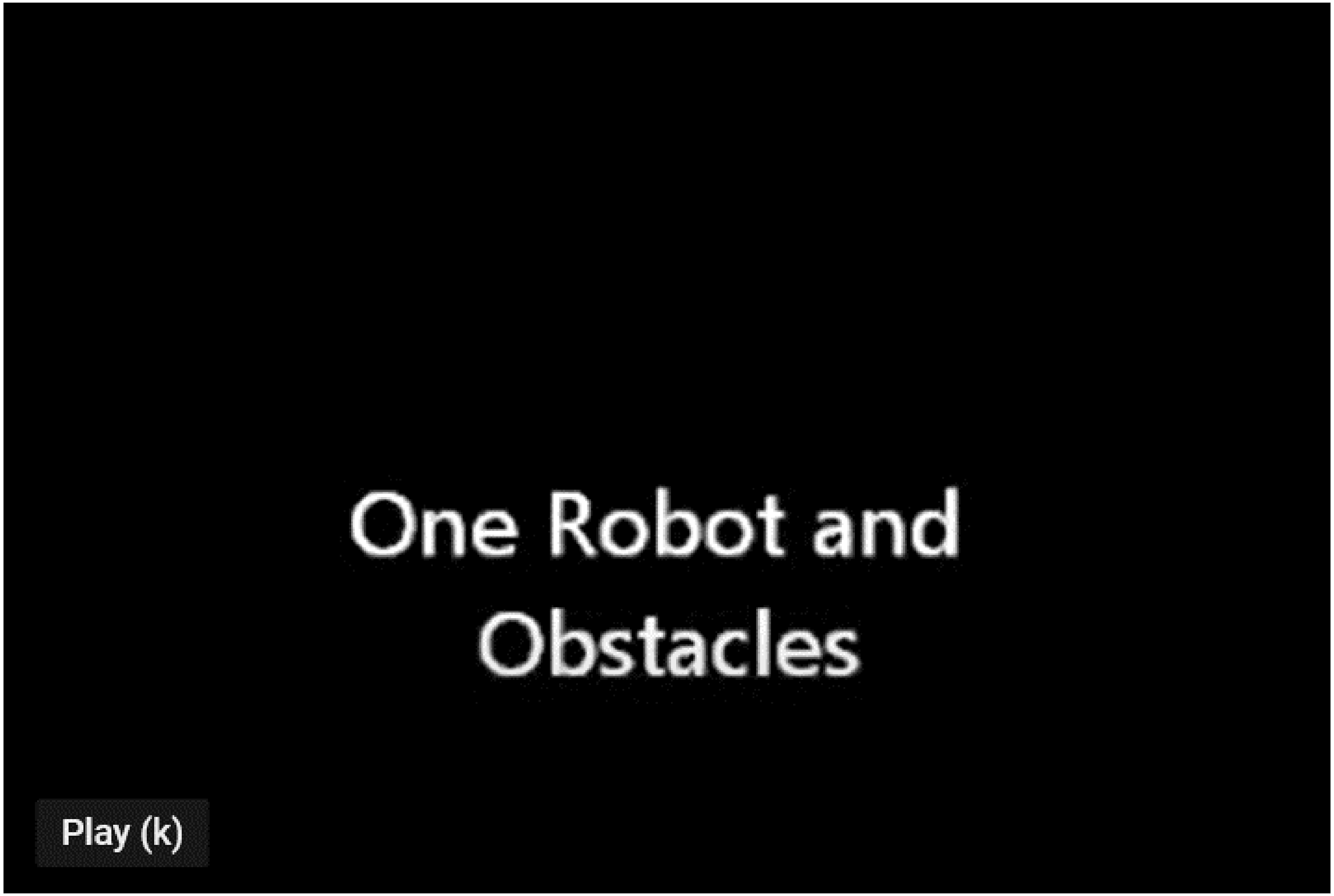
This technology includes A*, D*, visibility algorithm, Probabilistic Roadmap (PRM) , Rapidly Exploring Random Trees (RRT), ant swarm algorithm, and bug algorithm etc.





NUS National University of Singapore

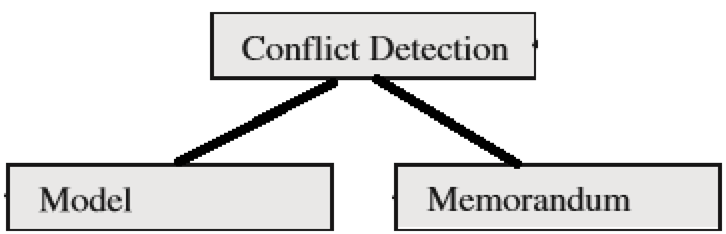
One example: D* search

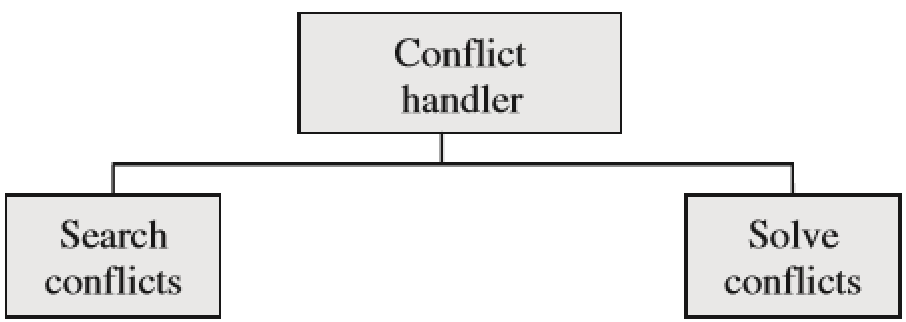


2.Conflict detection and resolution

The term conflict is defined as an event in which the time interval/distance/other parameters between two or multiple vehicle violate the normative separation.

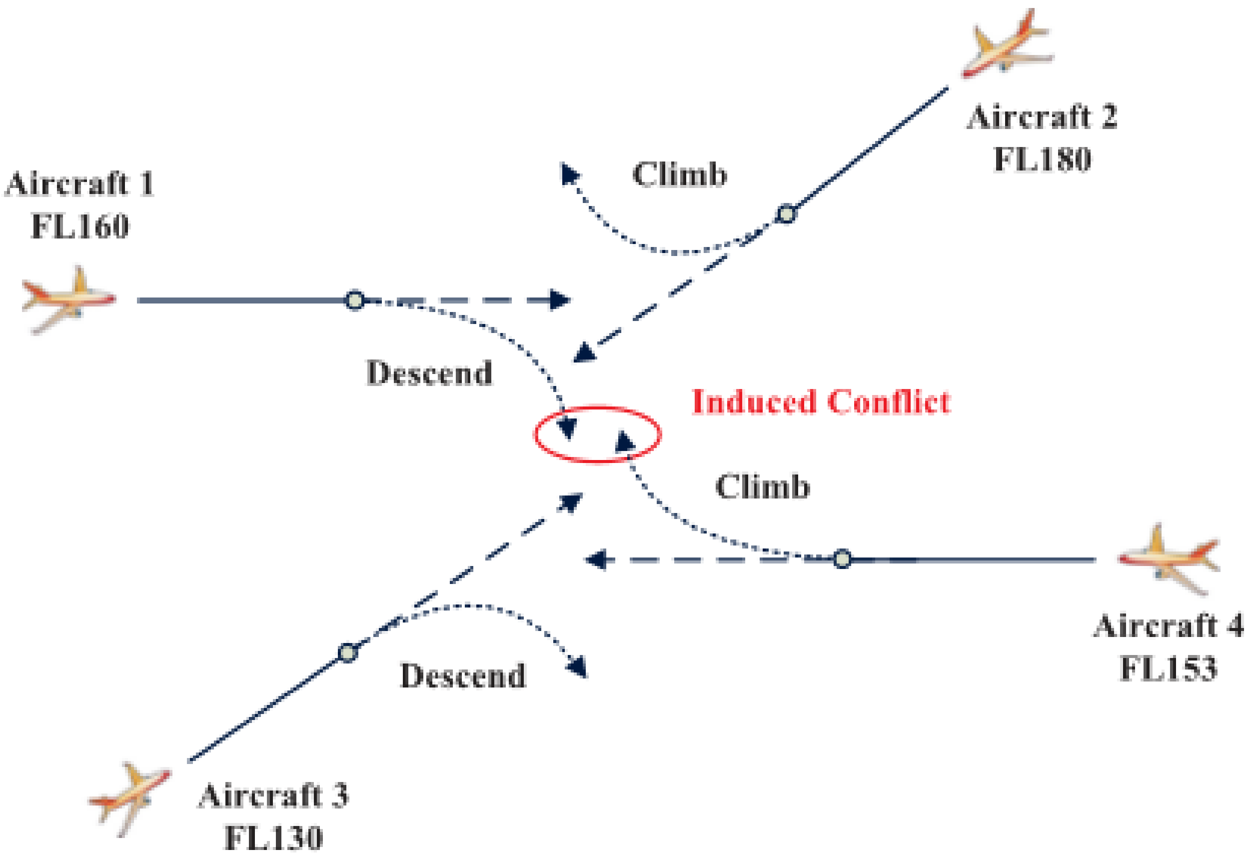
The term resolution is defined as a solver in which the conflict is solved by using a policy





https://www.researchgate.net/publication/336343228

Conflict detection and resolution



https://www.researchgate.net/publication/336343228

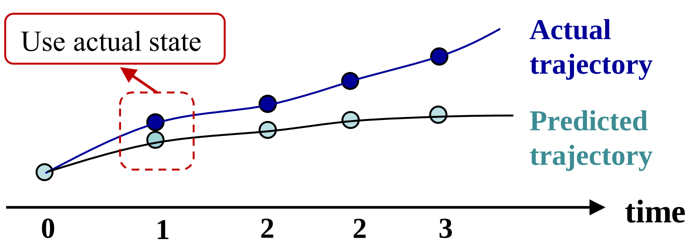
3. Model predictive control (MPC):

$$\min_{\boldsymbol{u}} \; J(\boldsymbol{u}) = l_f(x_N) + \sum_{t=0}^{N-1} l(x_t, u_t)$$

$$\begin{aligned} \text{s.t. } & x_0 = x_{\text{init}}, \\ & x_{t+1} = f(x_t, u_t), \; t \in [0, N-1] \\ & g(x_t, u_t) \le 0, \; t \in [0, N-1] \\ & g_f(x_N) \le 0 \end{aligned}$$

LQR control

- **Model-based control**



MPC :

- **Iterative, finite-horizon optimal control (similar to *replanning*)**
- **Model predicts T steps.**
- **Solve T-horizon optimal control at each time step, $T < N$**
- **Only the first step control is executed**

$$J(\boldsymbol{u}; t) = l_f(x_{t+T}) + \sum_{k=t}^{t+T-1} l(x_k, u_k)$$

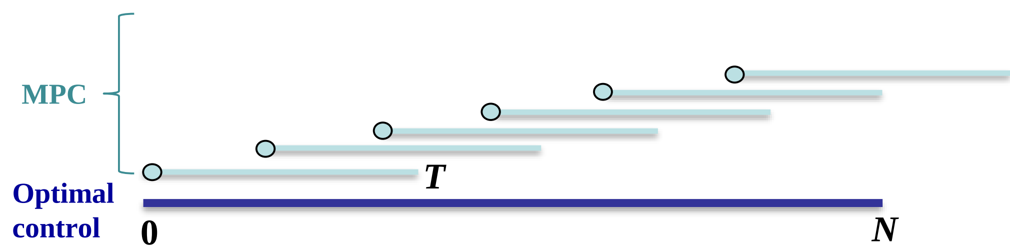
$t = 0, [0, T]: x_{init} = x(0), \quad J(\boldsymbol{u}; 0) \to [u_0, u_1, \cdots, u_{T-1}]$

$t = 1, [1, T+1]: x_{init} = x(1), \quad J(\boldsymbol{u}; 1) \to [u_1, u_2, \cdots, u_T]$

$t = 2, [2, T+2]: x_{init} = x(2), \quad J(\boldsymbol{u}; 2) \to [u_2, u_3, \cdots, u_{T+1}]$

$\vdots$

time



2017 IEEE Intelligent

Vehicles Symposium (IV)

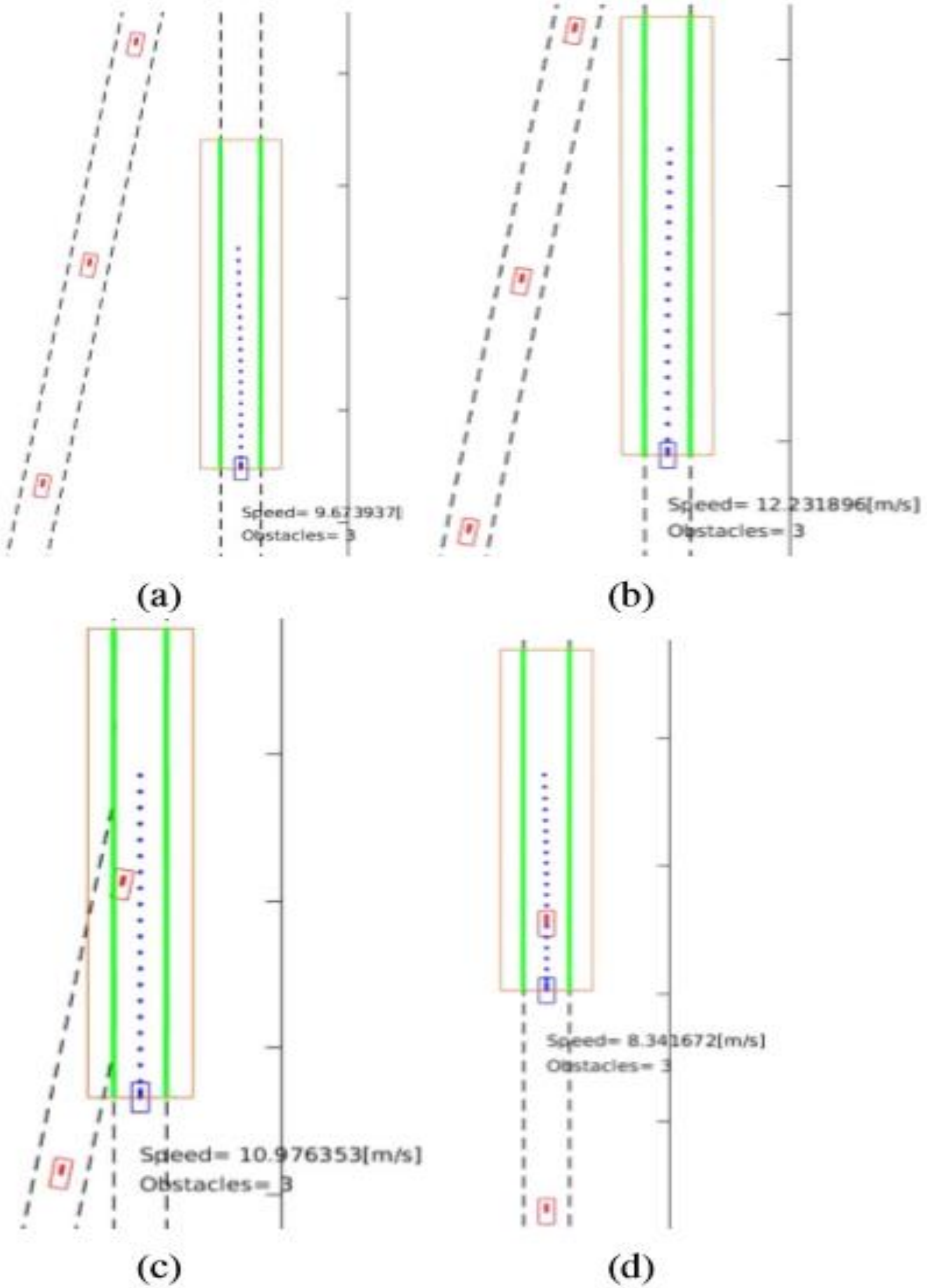
June 11-14, 2017,

Redondo Beach, CA, USA

Path Planning for Autonomous

Vehicles using

Model Predictive Control



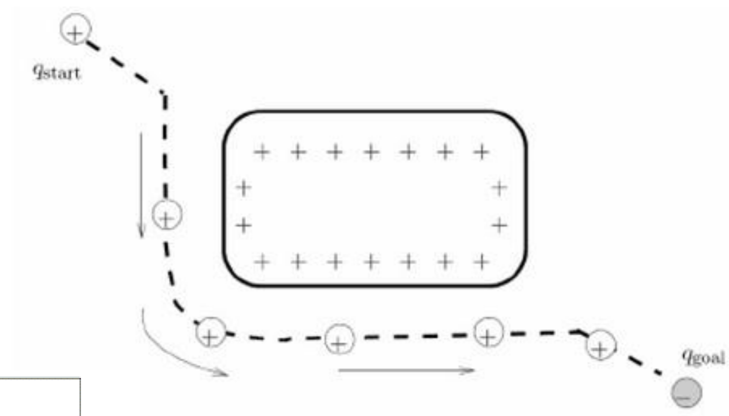
Potential Field methods use potential function U:

4. Potential function

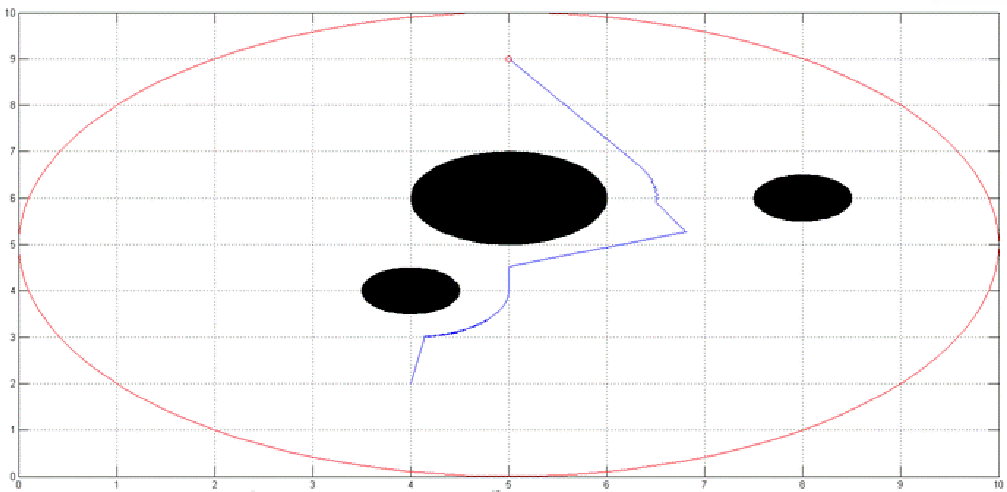
Potential Field methods use potential function U: $\mathbb{R}^m \rightarrow \mathbb{R}$.

Gradient of the U, $\nabla U(q)$ points in the direction that locally maximally increases U.

$$q_{\mathrm{next}} = q_{\mathrm{current}} - \alpha(i) * \nabla U(q)$$



Potential Field from the electrical charges idea.



- **Design potential function**
- **Suitable for simple form of obstacles**
- **It is difficult to handle 3D complex obstacles**

The potential field:

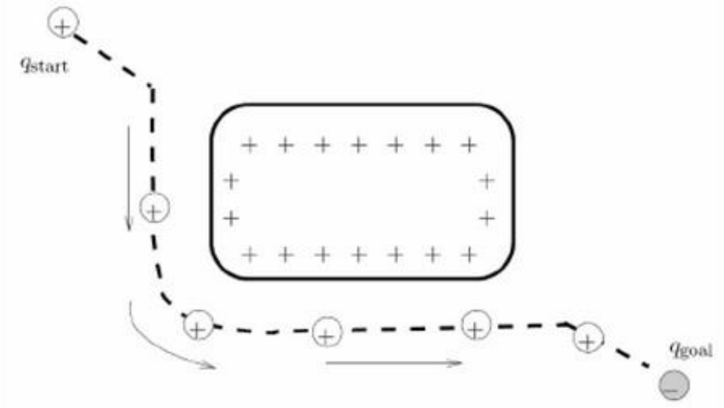
- **Attractive field of the goal**
- **Repulsive fields of the obstacles**

$$U = U_{att}(q) + U_{rep}(q)$$

Force is given by:

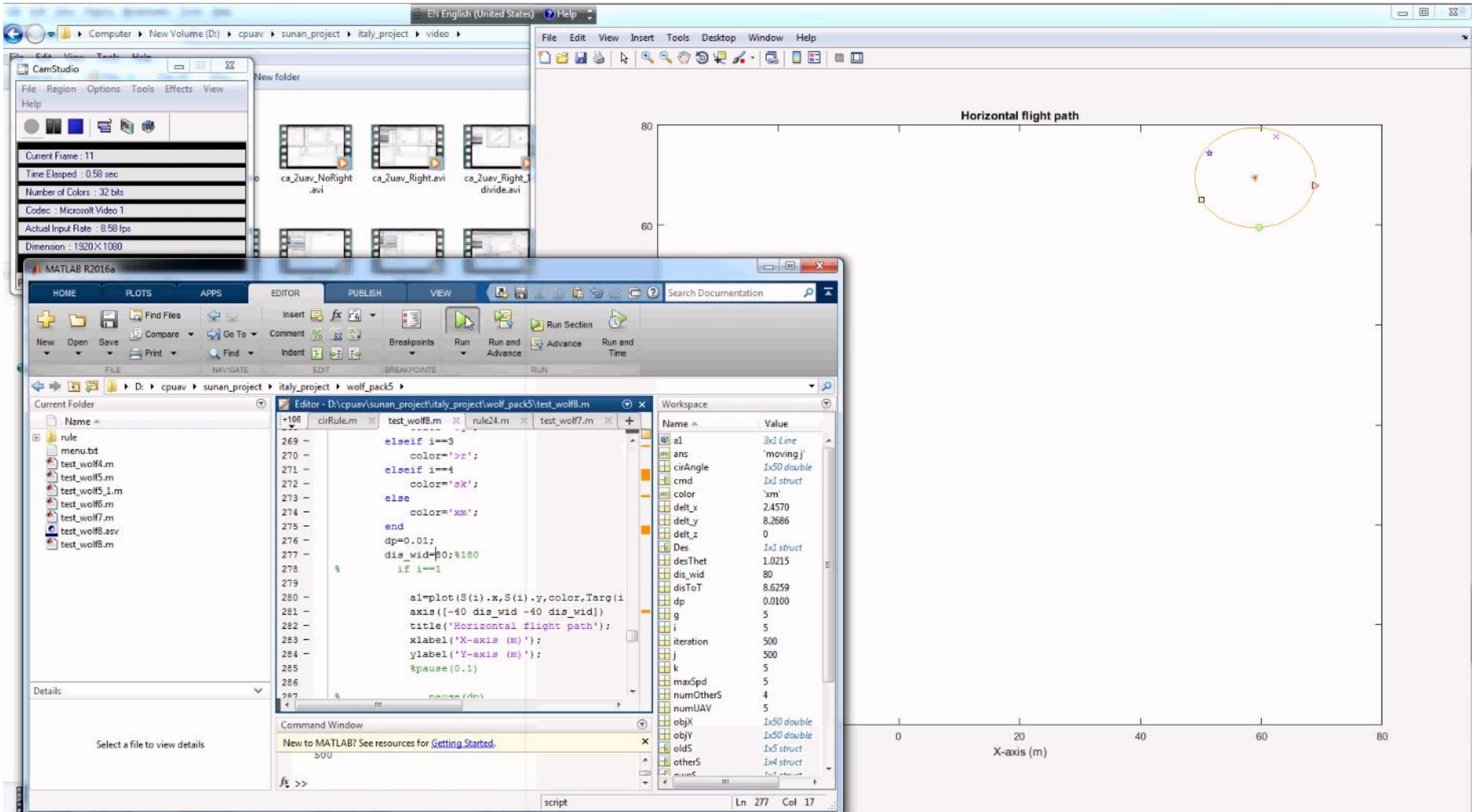
$$\boldsymbol{F = F_{att}(q) - F_{rep}(q)\nabla}$$

$$\boldsymbol{= \nabla U_{att}(q) - \nabla U_{rep}(q)}$$



Potential Field from the electrical charges idea.

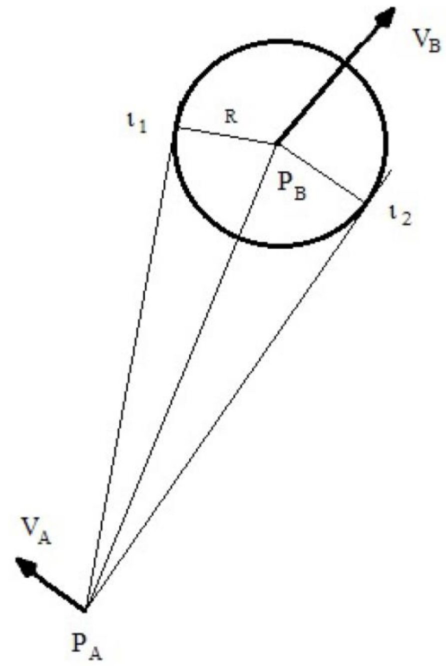
An application of potential function

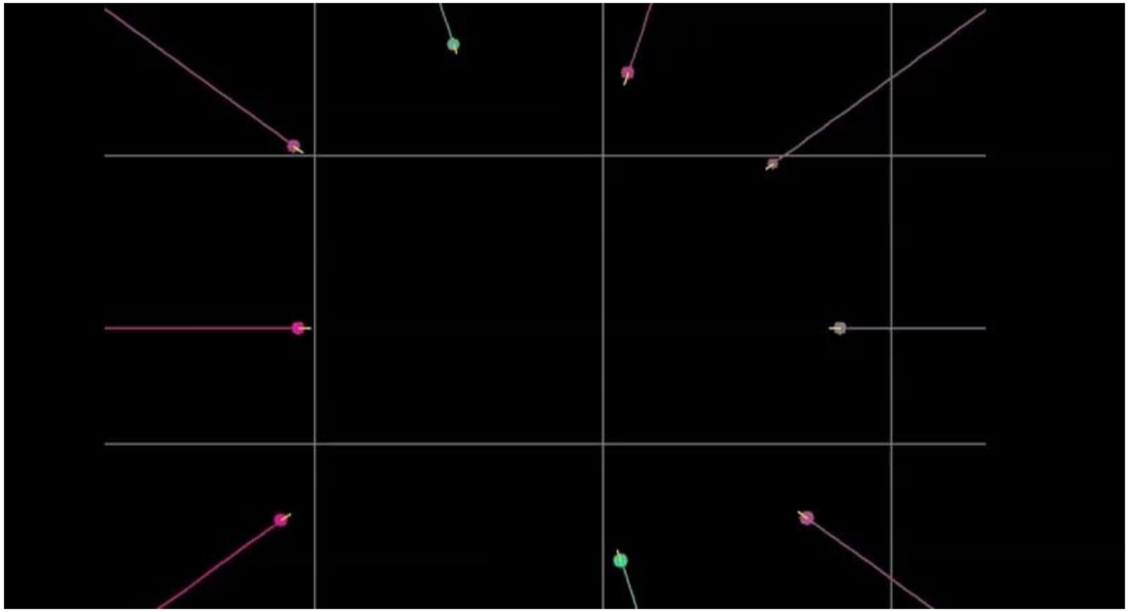


5. Geometric Guidance

(Velocity Obstacle Method)

Key idea: Collision Cone

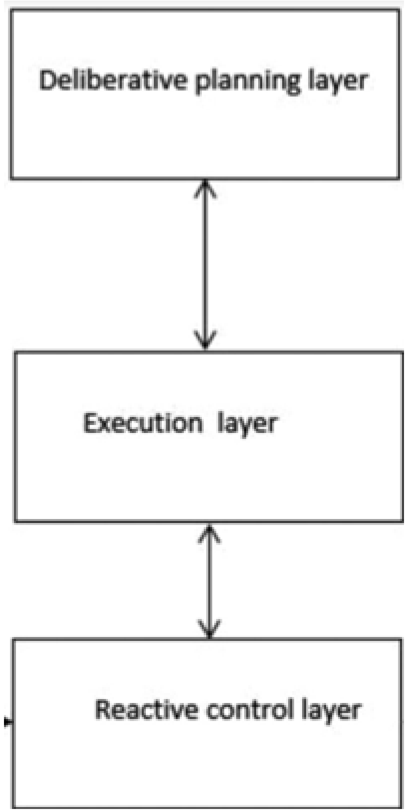




Because it is less computational complex, the velocity obstacle method is attractive in giving a fast solution in a dynamic environment

For each algorithm, you have to know that it is global search, or local search

?

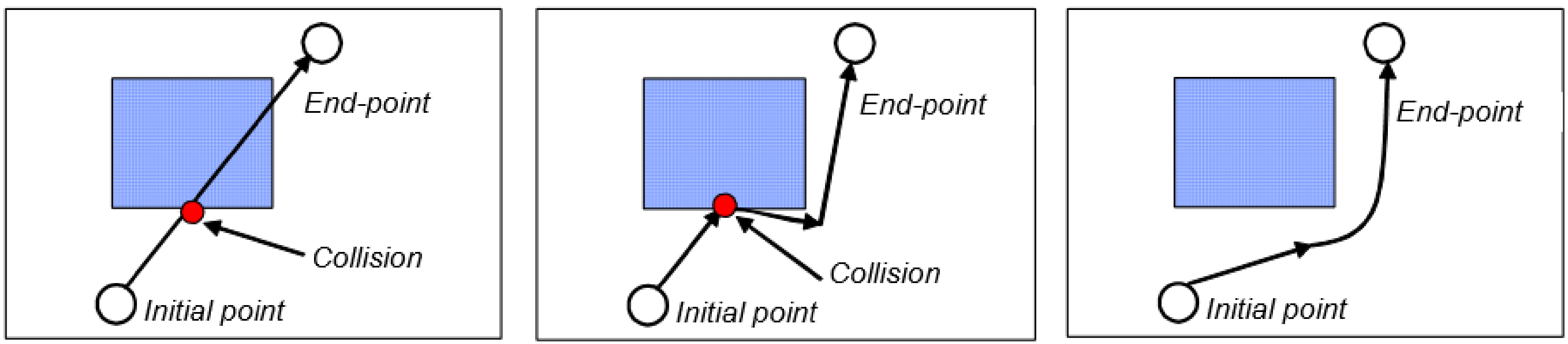


- **Global solution**
  - Graph search
  - Conflict detection and resolution
- **Local solution**
  - Model predictive control
  - Potential function
  - Geometric guidance

Summary

- Motion planning concept
  - C-space
  - Path planning
  - Trajectory planning

*C-space: grid, expansion and matrix expression



No collision avoidance

Local collision avoidance

Global collision avoidance

- **Motion planning architectures**
  - Deliberative planning
  - Reactive control
  - Hybrid control
- **Several typical algorithms**

Reference

- LaValle SM (2006) **Planning algorithms**. Cambridge university press
- Latombe J-C (2012) **Robot motion planning.** Springer Science & Business Media
- Hadi Jahanshahi and Naeimeh Najafizadeh Sari (2018) **Robot Path Planning Algorithms: A Review of Theory and Experiment**
- Sunan Huang, Rodney S.H.Teo, Kok Kiong Tan,

Collision avoidance of multi unmanned aerial vehicles: A review, Annual Reviews in Control 48, 147-164,2019

- **Path and trajectory planning, http://www.ene.ttu.ee/elektriajamid/oppeinfo/materjal/AAR0040/03_Robotics.pdf**

Thank you!

Recess time (15 mins)