MEAM 620 – Part II
Introduction to Motion Planning

Peng Cheng

chpeng@seas.upenn.edu

Levine 403,GRASP Lab

Part II Objectives

- Overview of motion planning
- Introduction to some basic concepts and methods for motion planning problems
- Implementation of motion planning algorithms
- Some thoughts for the future research

Course Plan

- 7 lectures about basic motion planning by Peng Cheng
- 3 programming assignments
  - Discrete planning (data structure + search)
  - Collision checking between two polygons
  - Planning algorithms for 2D problems
  - Matlab, C++, java, …
- 7 lectures in extensions of basic motion planning problems by Savvas Loizou
- 1 course project

References and Links

- [LaVallle 05] Planning algorithms

  http://msl.cs.uiuc.edu/planning
- [Choset et. al. 05] Principles of robot motion
- [Latombe 91] Robot Motion Planning
- [Laumond 98] Robot motion planning and control

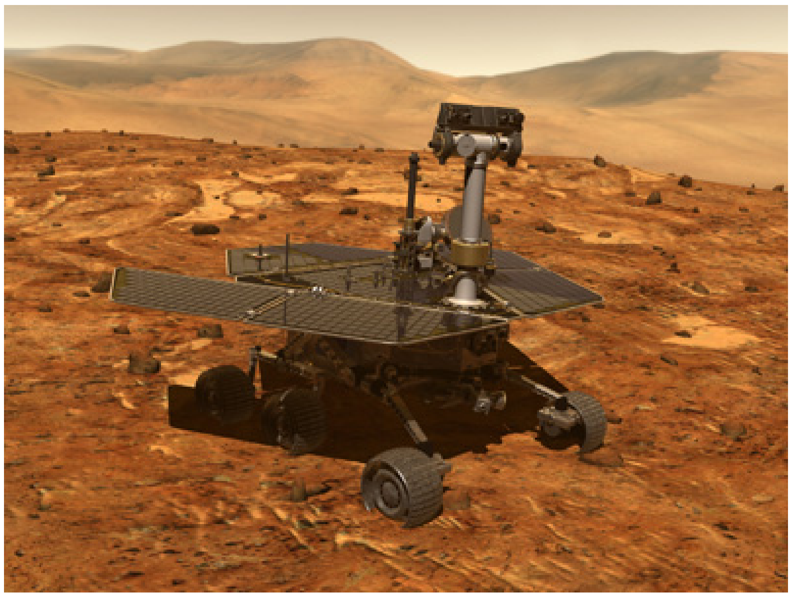
  http://www.laas.fr/~jpl/book.html
- Robotics VIP gallery

  http://parasol.tamu.edu/people/amato/Courses/643/VipGallery/

This Class

- **What is motion planning**
- Basic motion planning problems
- General principles of motion planning algorithms

Robotic Systems

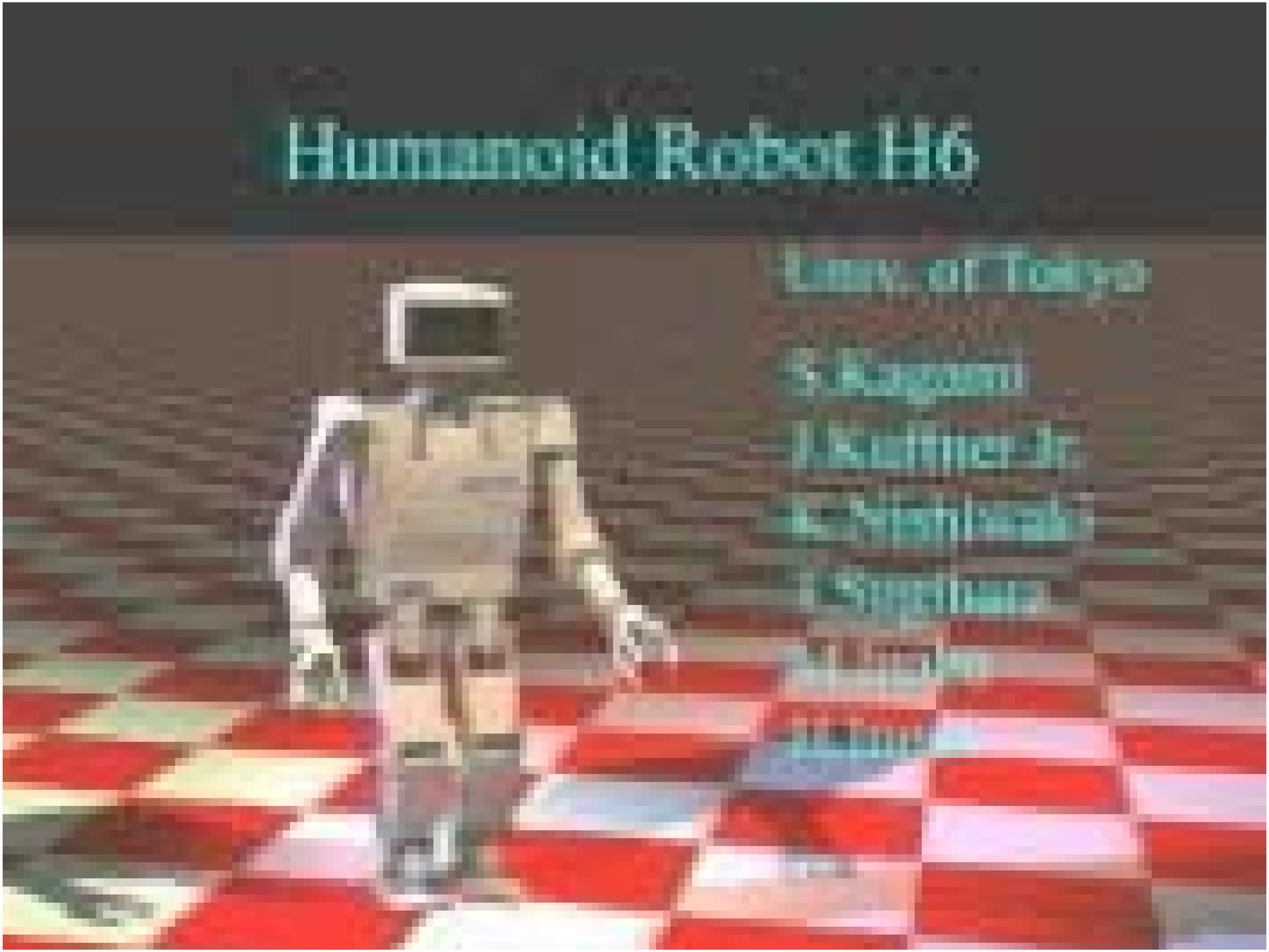


(Mars Rover, NASA)

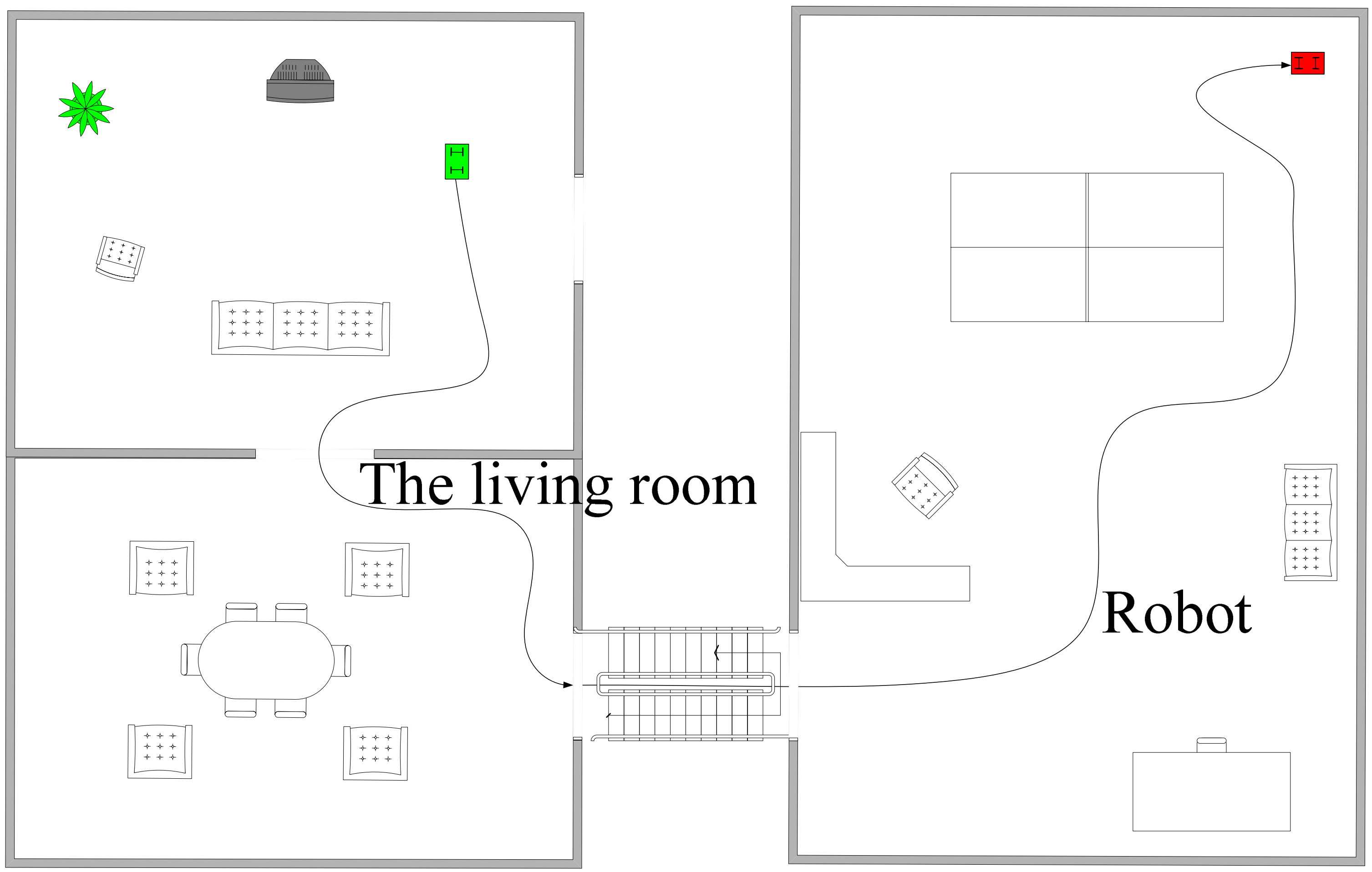
(Roomba, iRobot Co.)

Autonomously complete assigned tasks:

- Security and surveillance
- Search and rescue
- Outer space exploration
- Home services


Humanoid Robot H6
Univ. of Tokyo

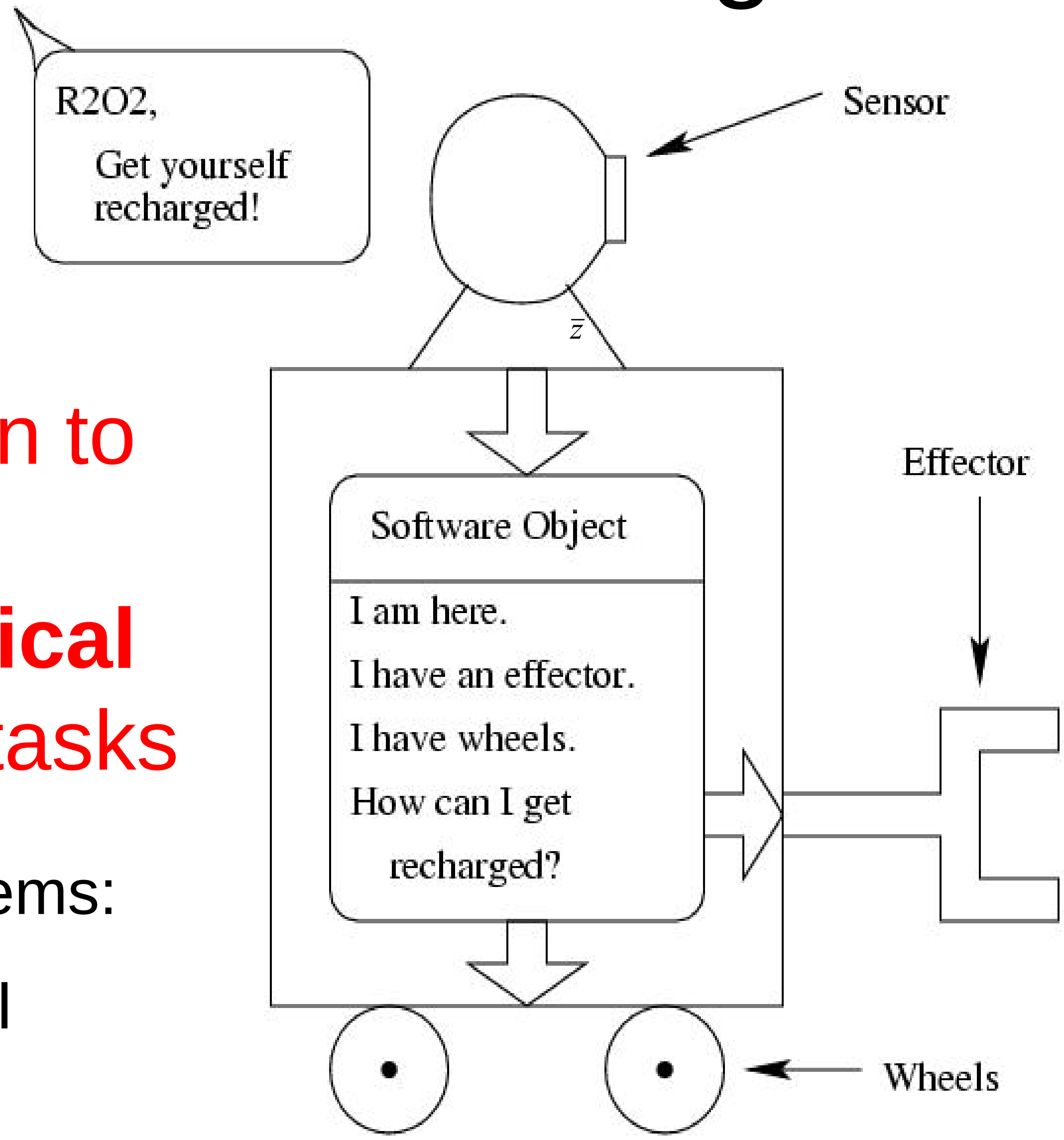
A Recharging Task for a Robot



General Motion Planning

An informal definition:

A mapping from sensing information to action inputs to complete **mechanical motion-required** tasks



Non motion planning problems:

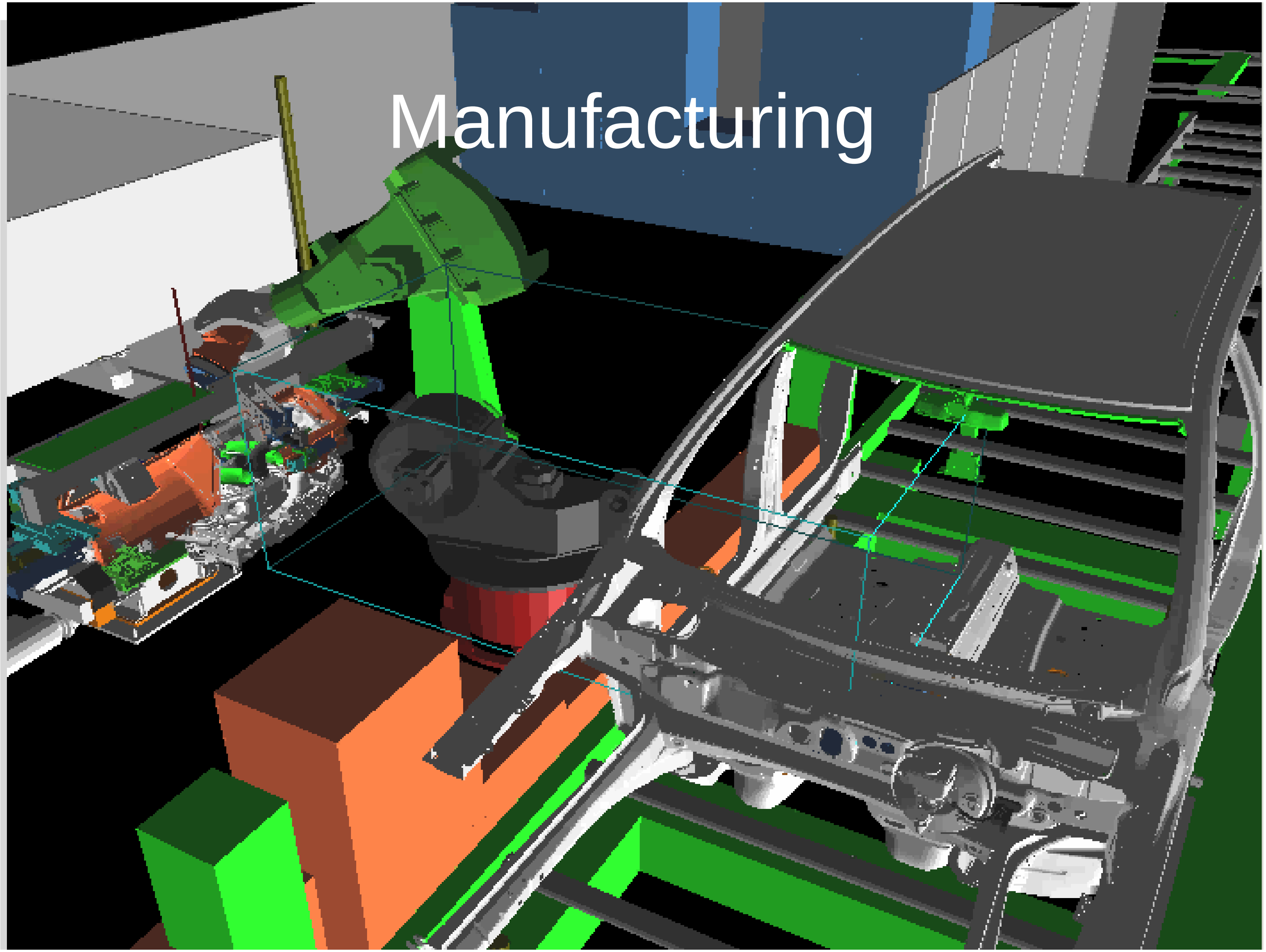
- Chemical reaction control
- Electronic motor control

Applications of Motion Planning

- Humanoid
- Manufacturing
- Autonomous systems
- Computer animation
- Rational drug design
- Virtual prototyping
- Many other cool things

(Asimo, Honda)

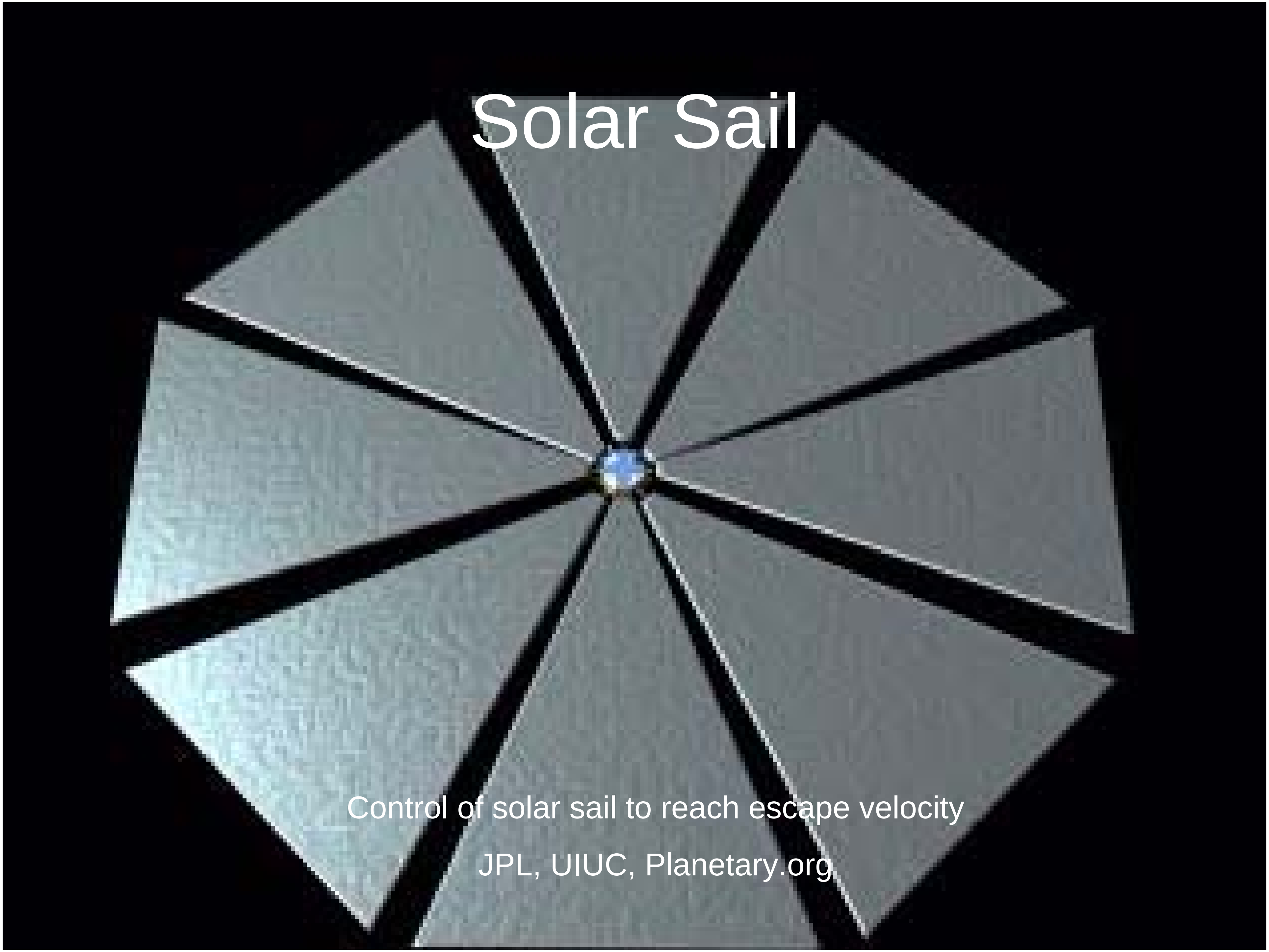
Manufacturing



Autonomous Agile Systems

Rapid taking-off and landing, OGI

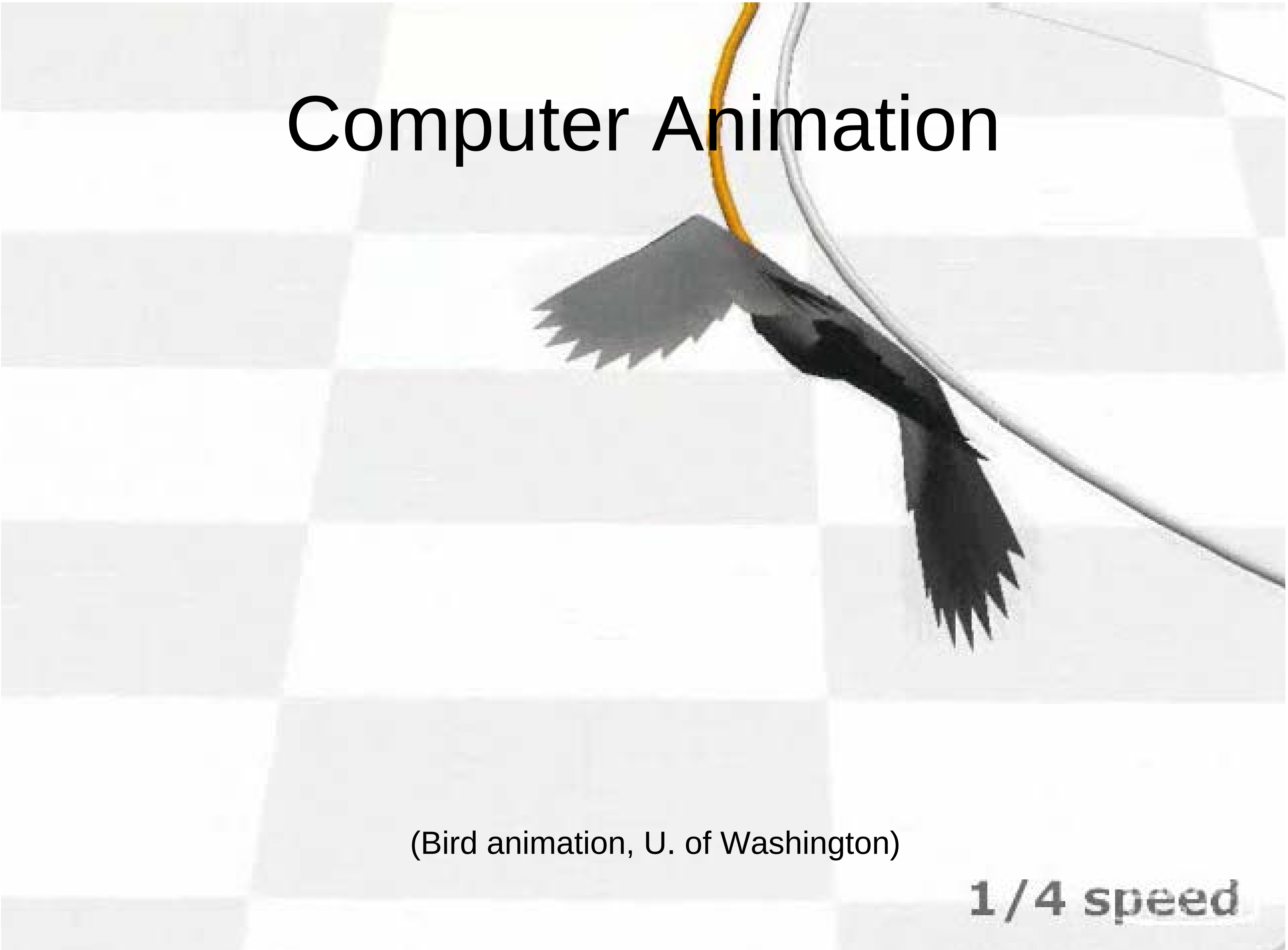
Solar Sail



Control of solar sail to reach escape velocity

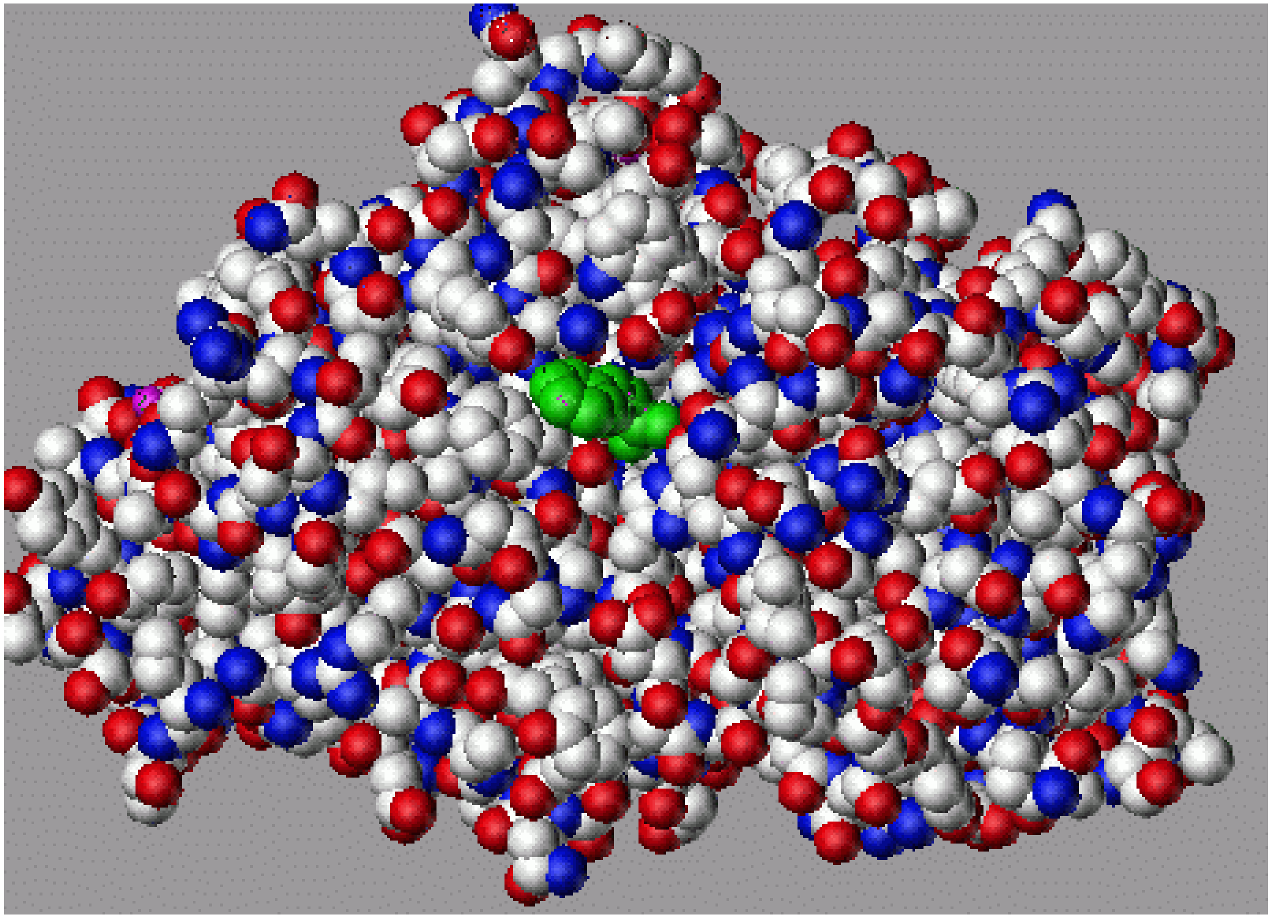
JPL, UIUC, Planetary.org

Computer Animation



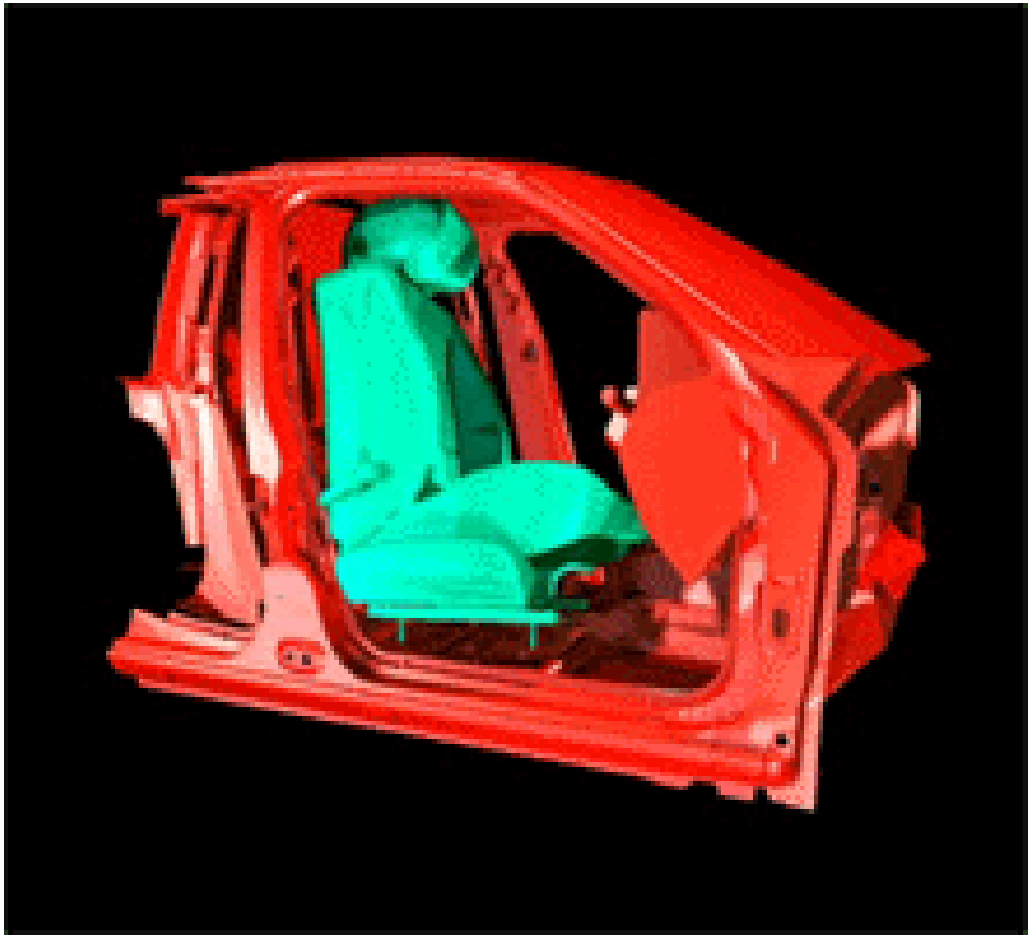
(Bird animation, U. of Washington)

Rational Drug Design



(Rational drug design, Rice Univ.)

Virtual Prototyping



Checking maintainability

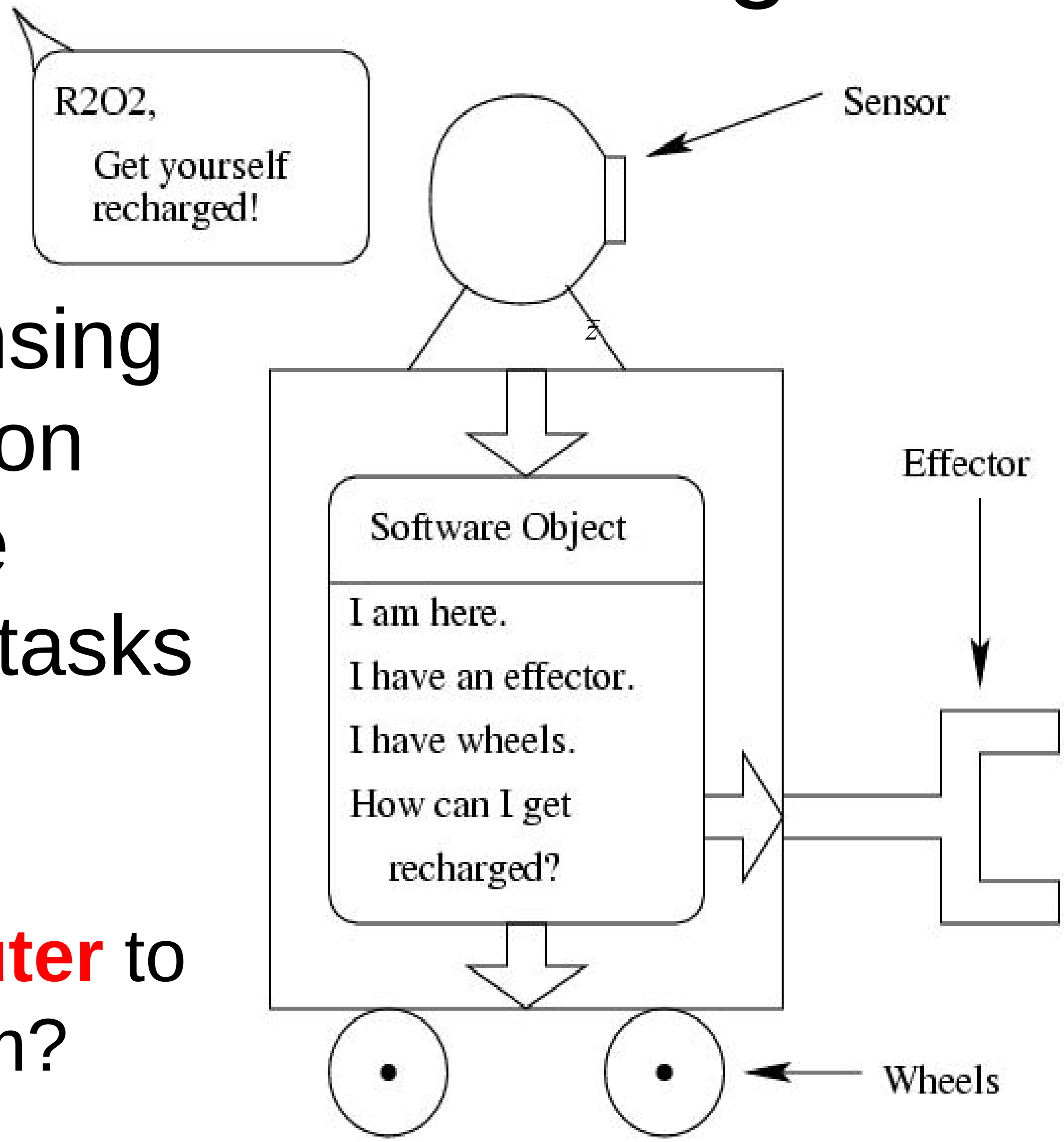
Outline

- What is motion planning
- **Basic motion planning problems**
- General principles of motion planning algorithms

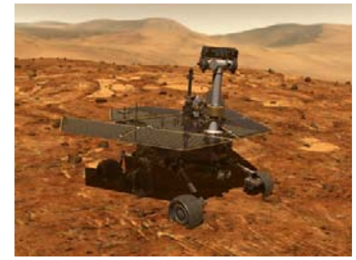
General Motion Planning

A mapping from sensing information to action inputs to complete **motion-required** tasks

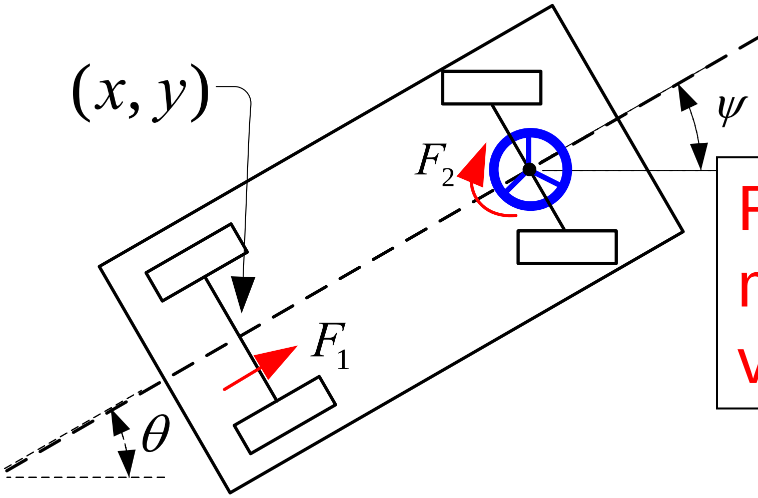
How to use **a computer** to solve this problem?



Building a Robot Model



- Motion model



Configuration, $q = (x, y, \theta, \psi) \in C$

Velocity, $\dot{q} = \dfrac{dq}{dt}$

Robot might not be able to move in any direction, velocity, and acceleration.

Input, $u = (F_1, F_2) \in U$

Motion equation, $\dot{z} = f(z, u)$

- Sensing model

  A mapping from sensin

Robot might not where it is exactly.

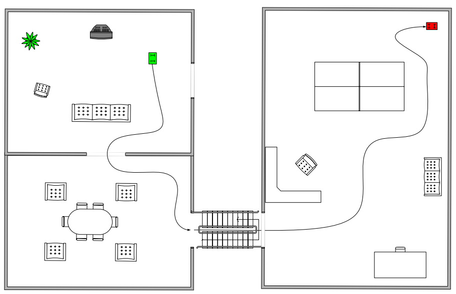
- Action model

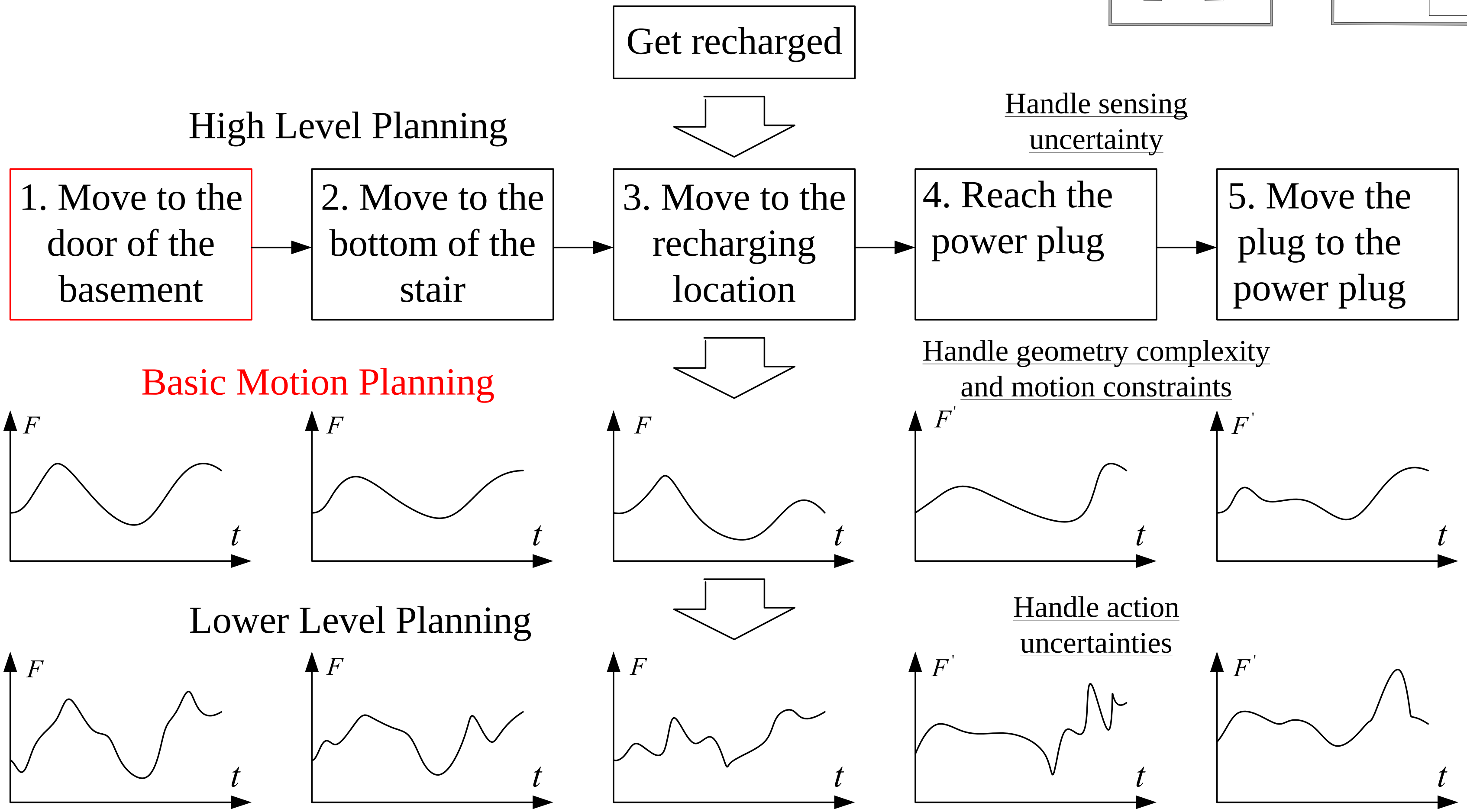
  A function from action i

Robot might not move to where it wants.

- Geometry model

The First Hierarchical Simplification





Basic Motion Planning Problem

Perfect sensing and action model

Given :

A robotic system,

work environment,

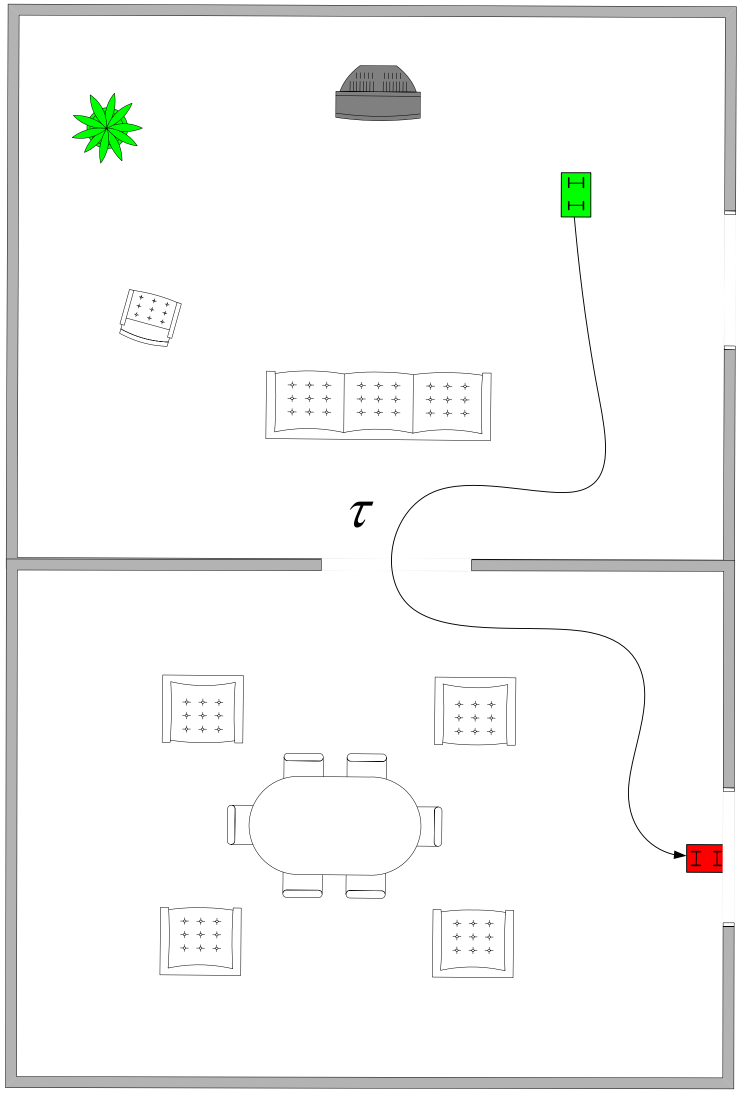
initial state,

and goal state.

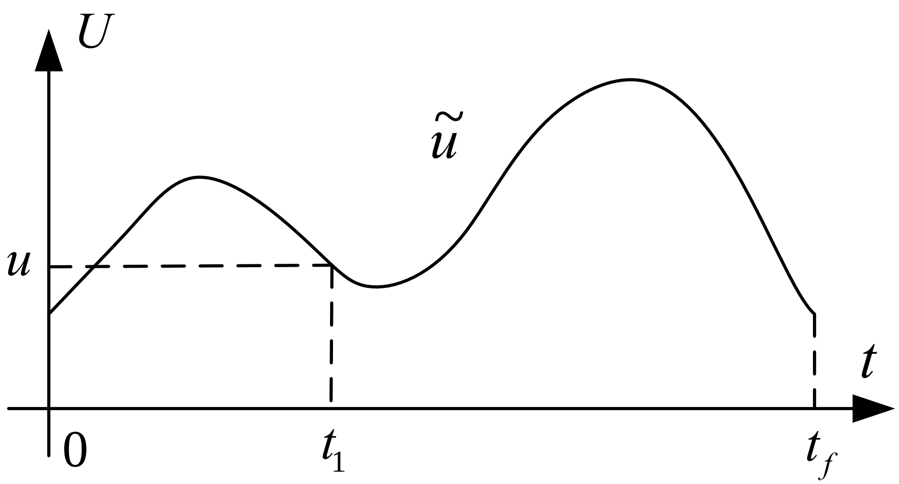
Solution :

a control, $\tilde{u} : [0, t_f] \to U$,

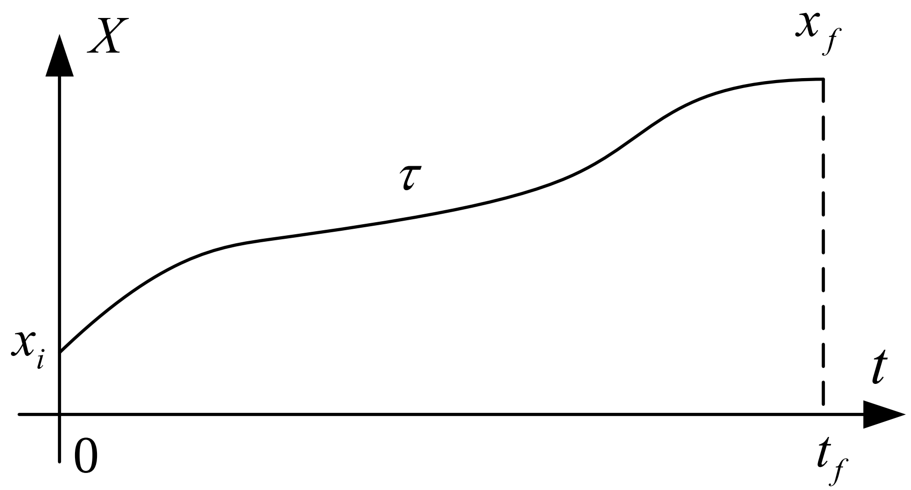
and its trajectory, $\tau : [0, t_f] \to X$.



A Control and Its Trajectory



control $\tilde{u}$

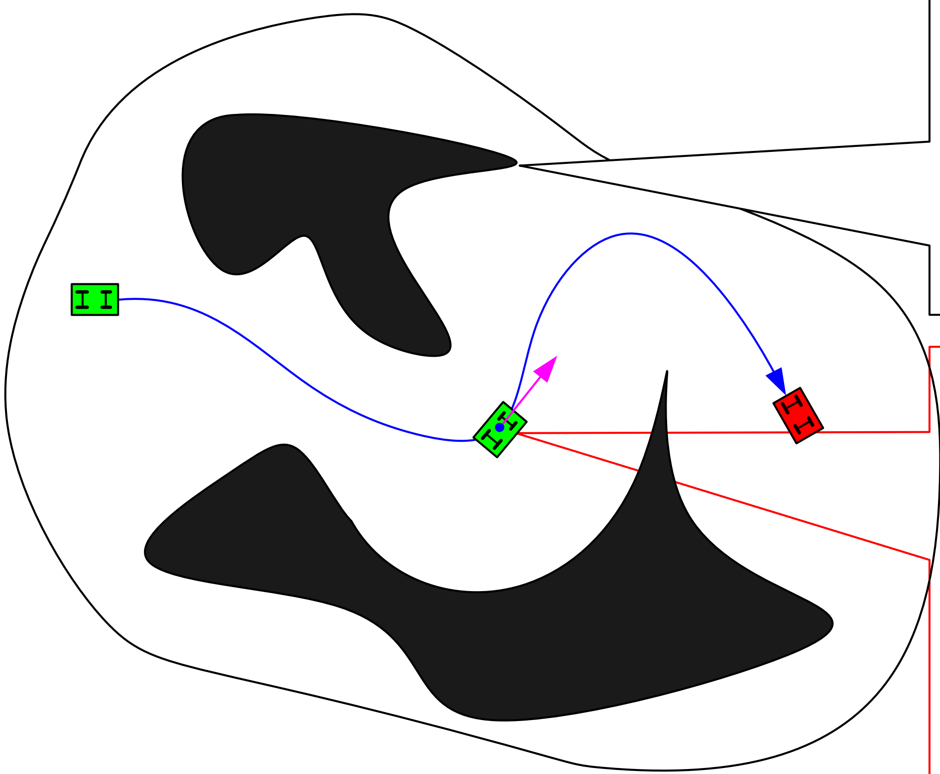


Trajectory τ of control $\tilde{u}$ from state x_i

Given a control $\tilde{u}:[0,t_f]\rightarrow U$ and state x_i,
a trajectory $\tau:[0,t_f]\rightarrow X$ is obtained by integrating
motion equation $\dot{x}=f(x,\tilde{u}(t))$ from state x_i.

Core Challenges

Search for control, $\tilde{u}$, and its trajectory, τ, which satisfy:



Configuration Constraints

$\{g_e(q) = 0; g_i(q) > 0\}$

- Collision avoidance
- Complicated geometry
- High Dimension

Differential Constraints

$\{h_e(q,\dot{q}) = 0; h_i(q,\dot{q}) > 0\}$

$\{k_e(q,\dot{q},\ddot{q}) = 0; k_i(q,\dot{q},\ddot{q}) > 0\}$

- Restricted velocities and accelerations
- Nonlinear constraints

The Second Hierarchical Simplification

- Path planning

  With only configuration constraints

$$\{g_e(q) = 0; g_i(q) > 0\}$$

- Nonholonomic planning

  With first order differential constraints

$$\{h_e(q, \dot{q}) = 0; h_i(q, \dot{q}) > 0\}$$

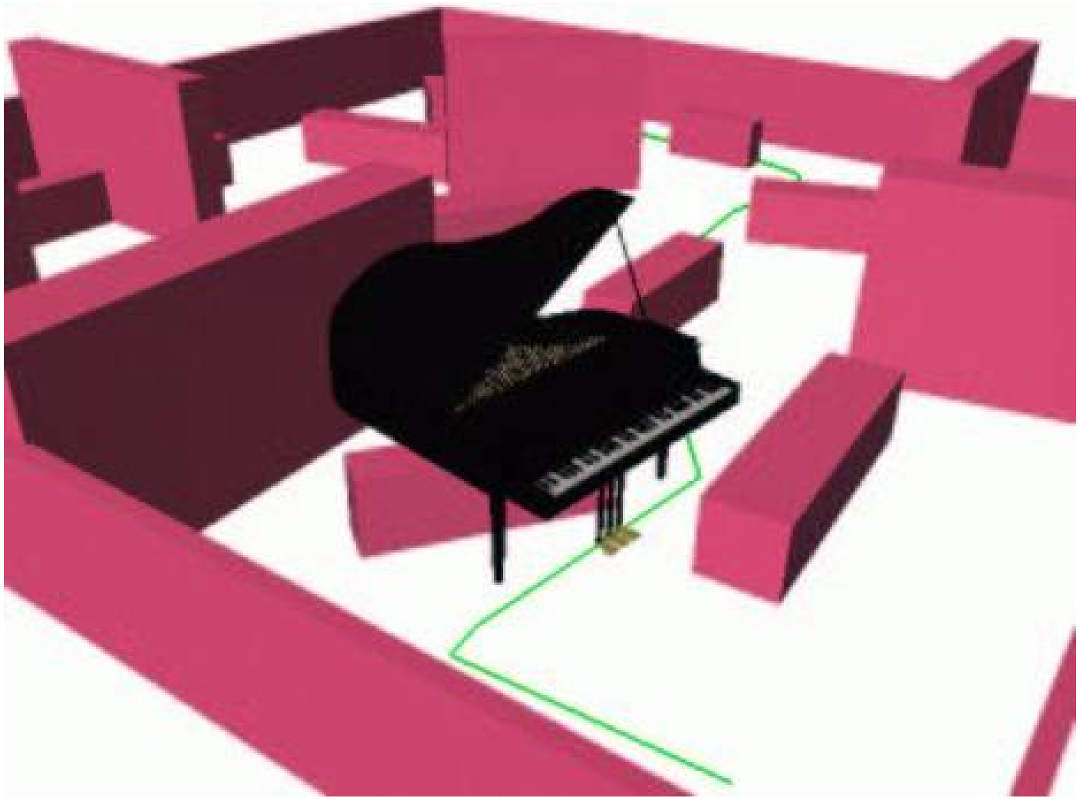
- Kinodynamic planning

  With second order differential constraints

$$\{k_e(q, \dot{q}, \ddot{q}) = 0; k_i(q, \dot{q}, \ddot{q}) > 0\}$$

Path Planning Problem

(Piano Movers' Problem)



Given:

Geometry and configuration constraints $\{g_e(q) = 0; g_i(q) > 0\}$

Objective:

A **collision-free** path connecting the initial and goal configurations

Paths from Path Planning Could Be Infeasible

(Parallel Parking)

Infeasible because the robot cannot follow it.

Nonholonomic planning Problem

(called steering problem when no obstacles exist)

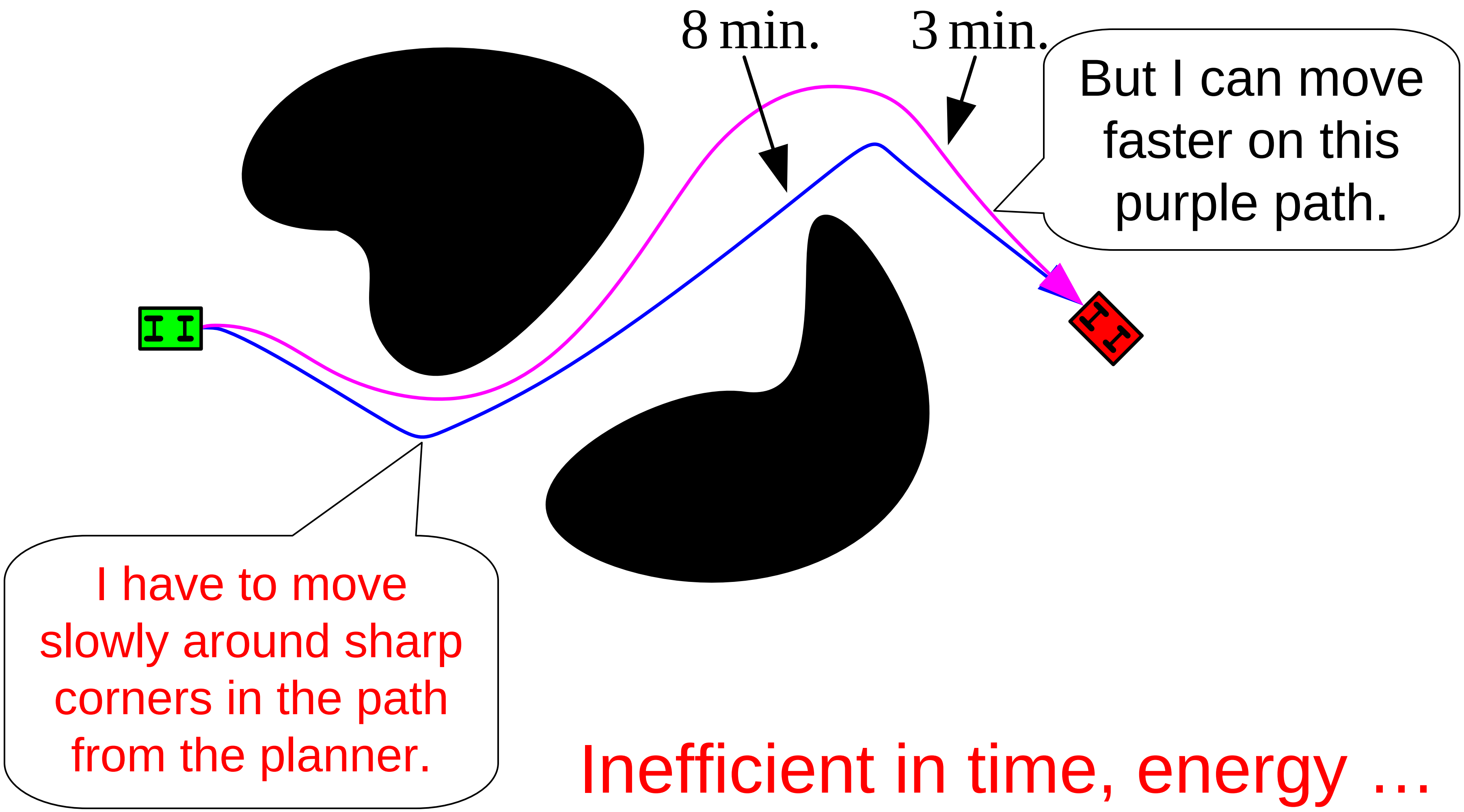
Given:

A system under nonholonomic constraints $\{h_e(q,\dot{q})=0; h_i(q,\dot{q})>0\}$

Objective:

A open loop control to move from the initial configuration to the goal configuration

Paths Could Be Inefficient



Inefficient in time, energy …

Kinodynamic Planning Problem

Given:

A system under dynamics constraints

$$k_e(q,\dot{q},\ddot{q}) = 0, k_i(q,\dot{q},\ddot{q}) > 0$$

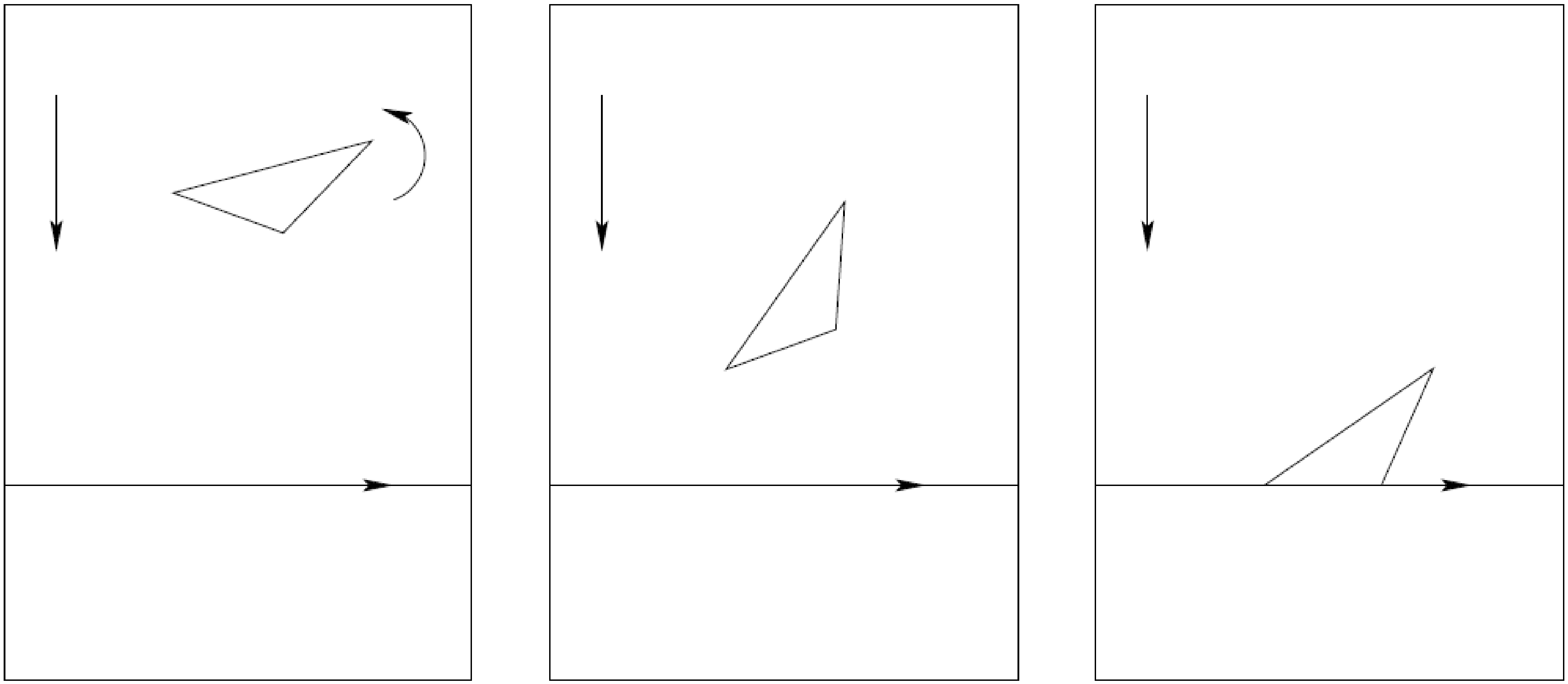
Objective:

A open loop control to move from the initial state to goal state

Extensions of Basic Motion Planning Problems

Planning with Sensing Uncertainties

Perfect action model



Part orienting (Akella, Mason, 99)

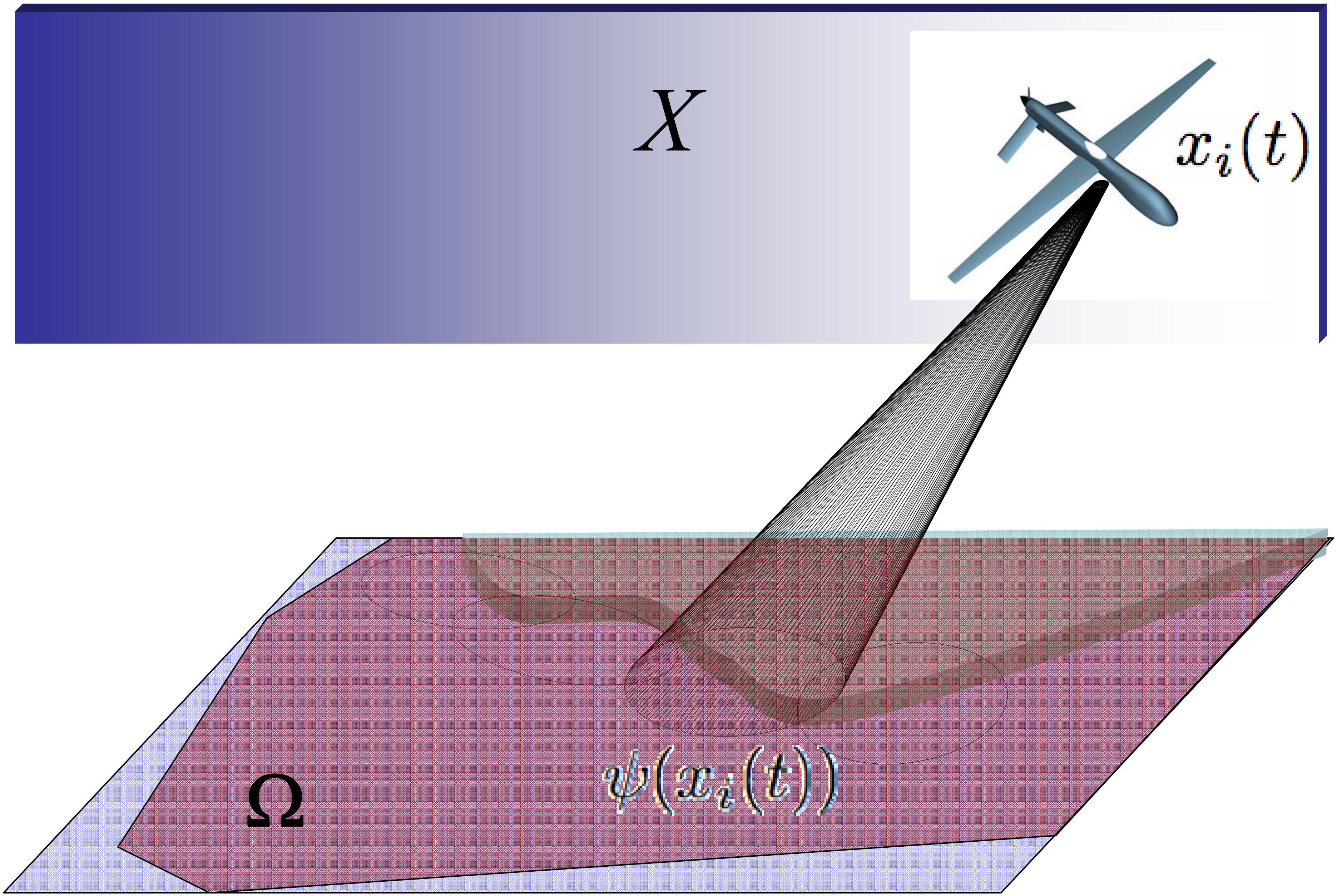
Coverage Planning

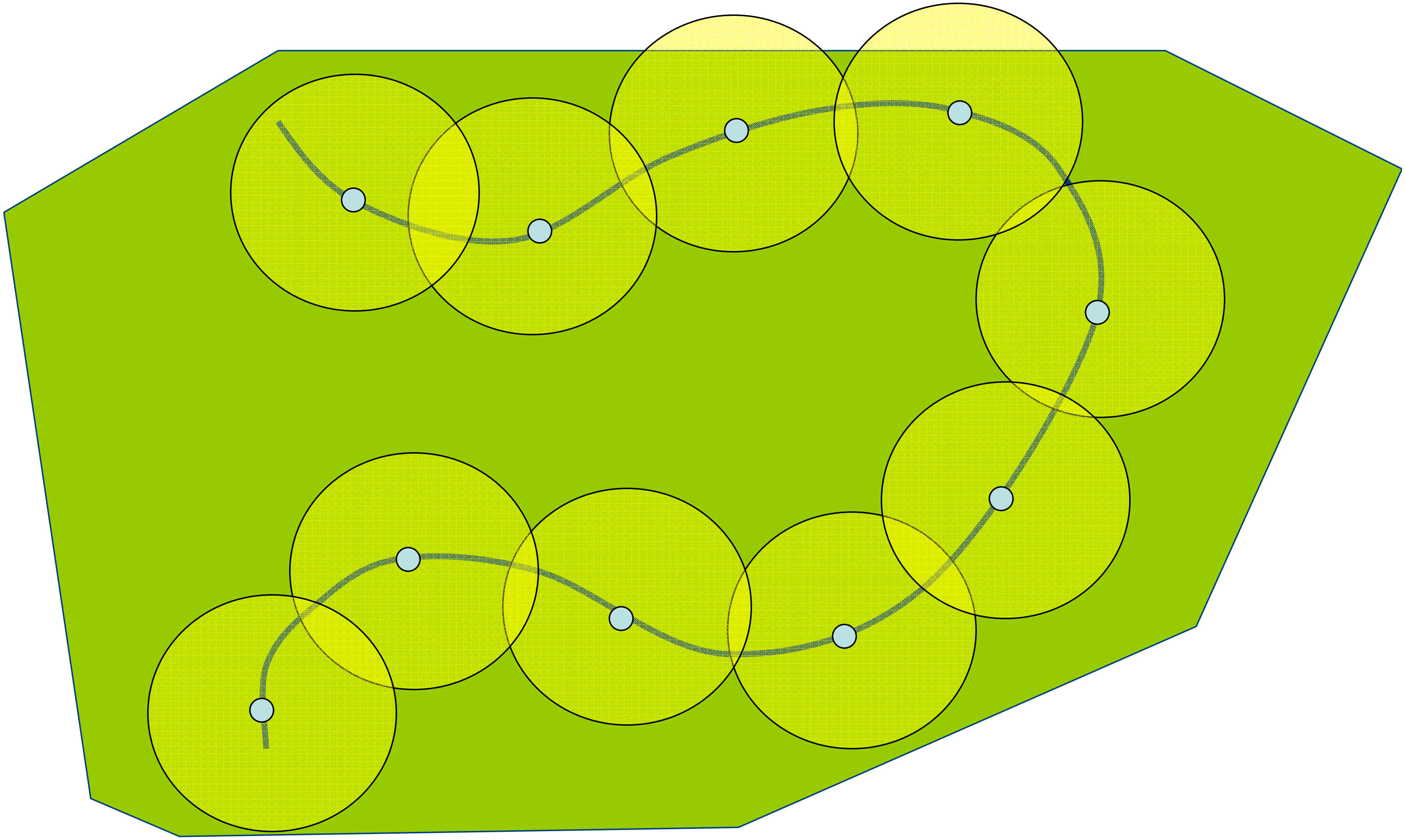
Given:

- Robotic systems
- A closed area
- A footprint function

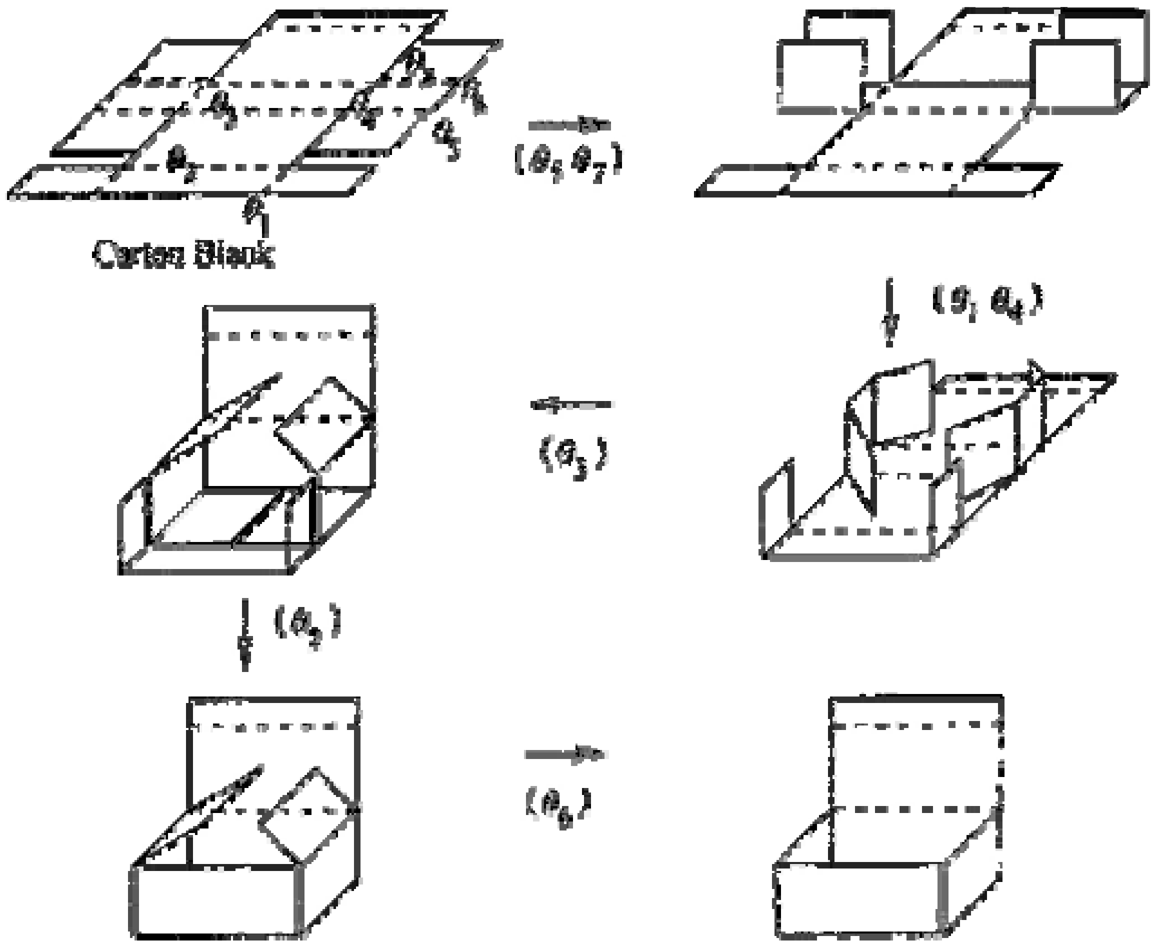
Objective:

- Find controls for the systems to cover the given area.


X
$x_i(t)$
Ω
$\psi(x_i(t))$



Folding

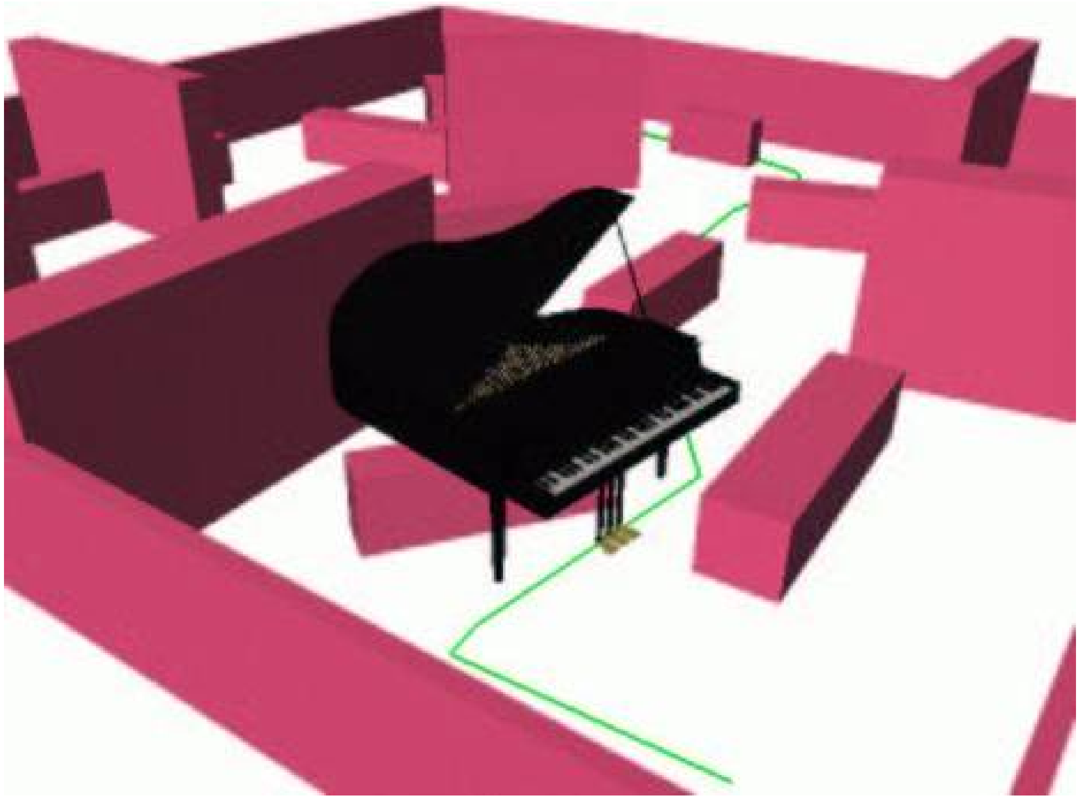


A folding sequence for a carton (Lu, Akella, 00)

Outline

- What is motion planning
- Different motion planning problems
- **<u>General principles of motion planning algorithms</u>**

Path Planning Problem
(Piano Movers' Problem)



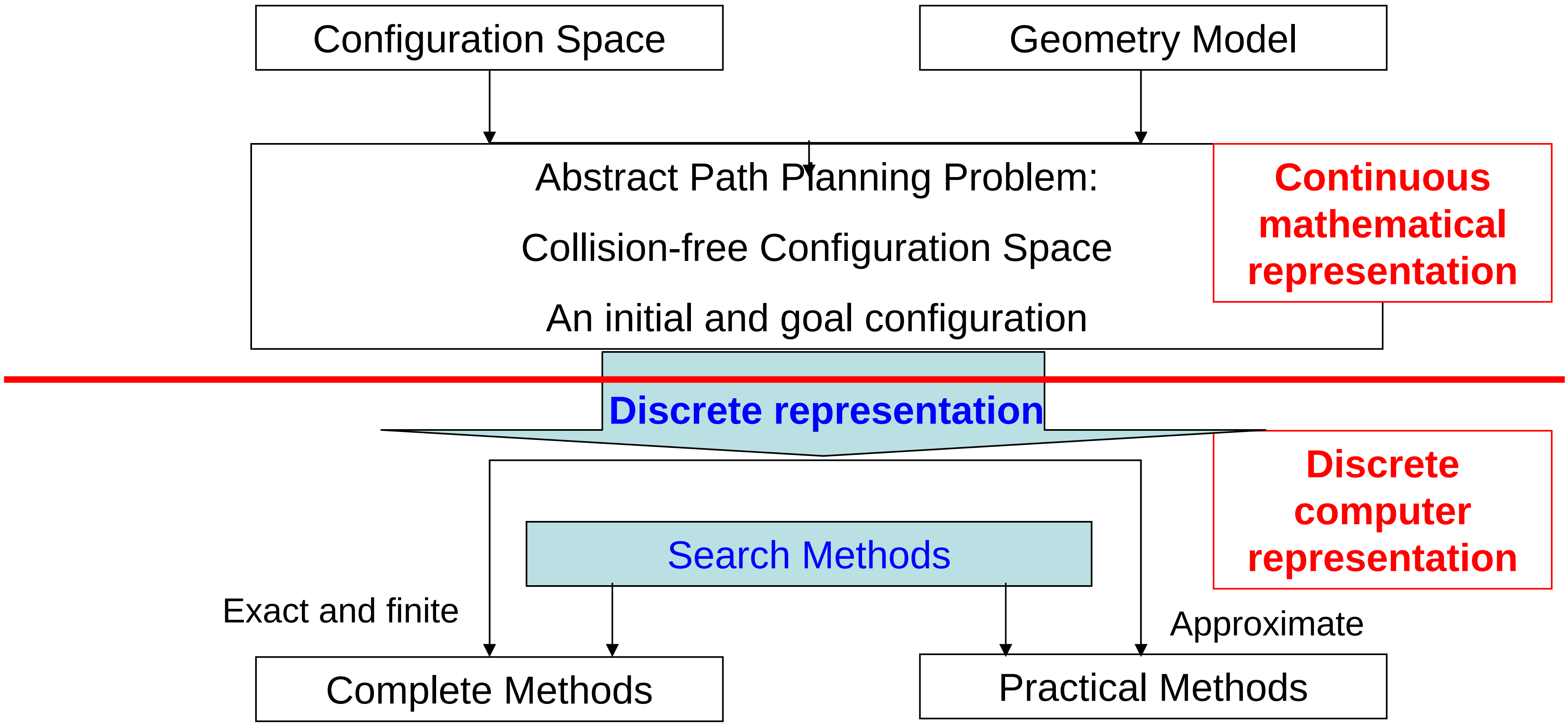
Given:

Geometry and configuration constraints $\{g_e(q) = 0; g_i(q) > 0\}$

Objective:

A collision-free path connecting the initial and goal configurations

Path Planning Algorithms



Representations

- Continuous mathematical representation
  - Mathematical formulation of the problem
  - Topology of the configuration space
  - Parameterization of configuration space
  - Computation of collision-free configuration space
  - Abstract path planning problems
- Discrete computational representations
  - Required for computer algorithms
  - Exact and finite representations of the set of collision-free configurations
  - Approximate representations

Search Methods

- Discrete search methods
  - Search with finite states and actions
- Deterministic complete search
  - Best-first, A*, Dijstra, …
- Sampling-based algorithms
  - PRM(Probabilistic Road Map)-based methods
  - RRT (Rapidly Exploring Random Tree)-based methods
  - Heuristic design

Reading

- [LaValle 05], II-Overview
- [LaValle 05], II-3.1
- [LaValle 05], II-4.1,4.2,4.3