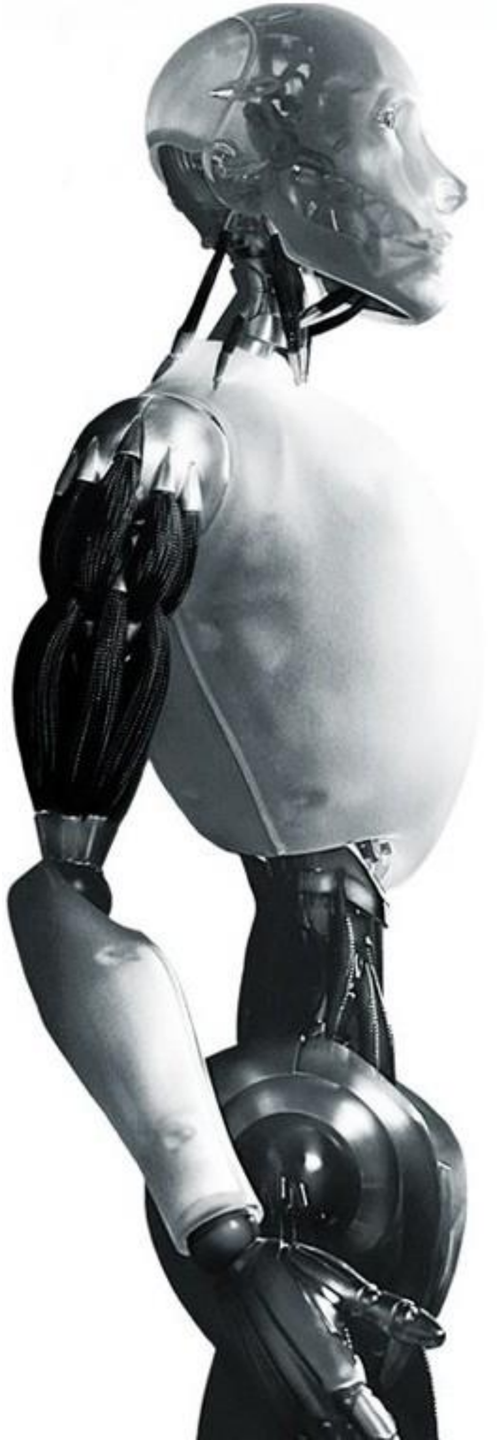
Disclaimer

These slides are intended as presentation aids for the lecture. They contain information that would otherwise be to difficult or time-consuming to reproduce on the board. But they are incomplete, not self-explanatory, and are not always used in the order they appear in this presentation. As a result, these slides should not be used as a script for this course. I recommend you take notes during class, maybe on the slides themselves. It has been shown that taking notes improves learning success.

Reading for this set of slides

- Planning Algorithms (Steve LaValle)
  - 4 The Configuration Space (4.1 – 4.3)
  - 5 Sampling-based Motion Planning (5.1, 5.5, 5.6, also skim the remaining sections)

Please note that this set of slides is intended as support for the lecture, not as a stand-alone script. If you want to study for this course, please use these slides in conjunction with the indicated chapters in the text books. The textbooks are available online or in the TUB library (many copies that can be checked out for the entire semester. There are also some aspects of the lectures that will not be covered in the text books but can still be part of the homework or exam. For those It is important that you attend class or ask somebody about what was covered in class.



Robotics

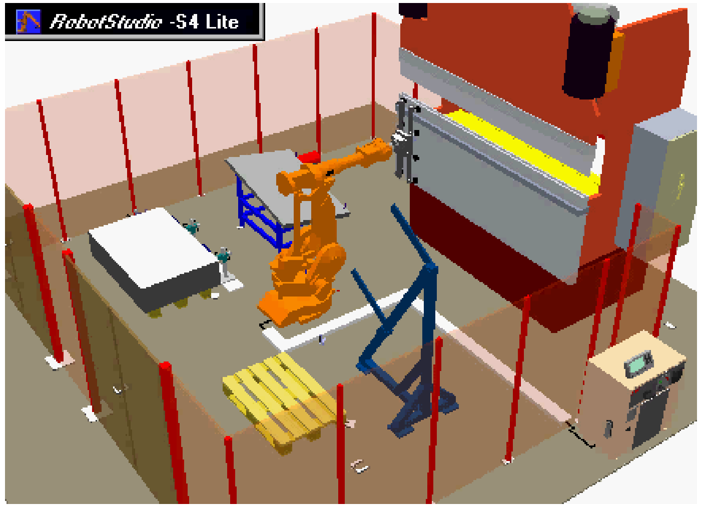
Motion Planning

TU Berlin

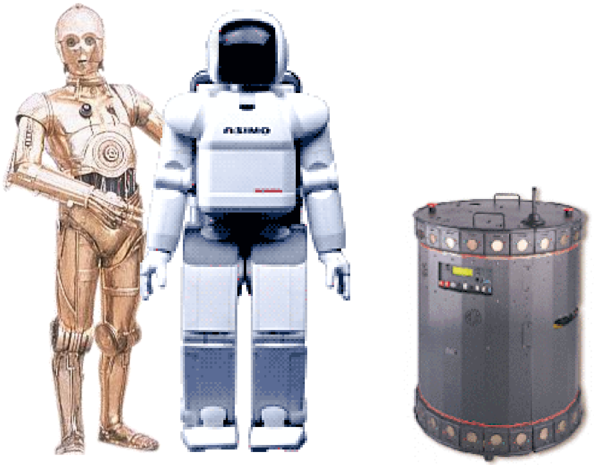
Oliver Brock

Problem	Stationary Robots	Mobile Robots
Move to a given position (ignoring obstacles)	Control, artificial potential fields; almost always holonomic kinematics	Control; artificial potential fields; often nonholonomic kinematics
Modeling disturbances	No slip but maybe flexibilities; dynamics (gravity, inertia, centrifugal and Coriolis forces); friction, stiction	Slip (very difficult); dynamics almost always ignored
Knowing obstacles around you	3D perception problem; 3D SLAM; visual servoing	Laser range finder solves 2D perception problem; SLAM; visual servoing
Knowing where you are	Use special sensors (encoders)	Localization, Mapping, SLAM
Move to a given position (considering obstacles)	Local artificial potential fields, navigation functions, **global motion planning**	Local artificial potential fields, global navigation functions (NF1, NF2, harmonic potentials)

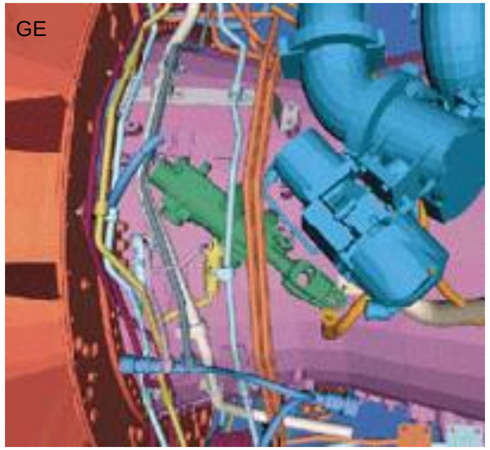
Why Motion Planning?



Robert Bohlin

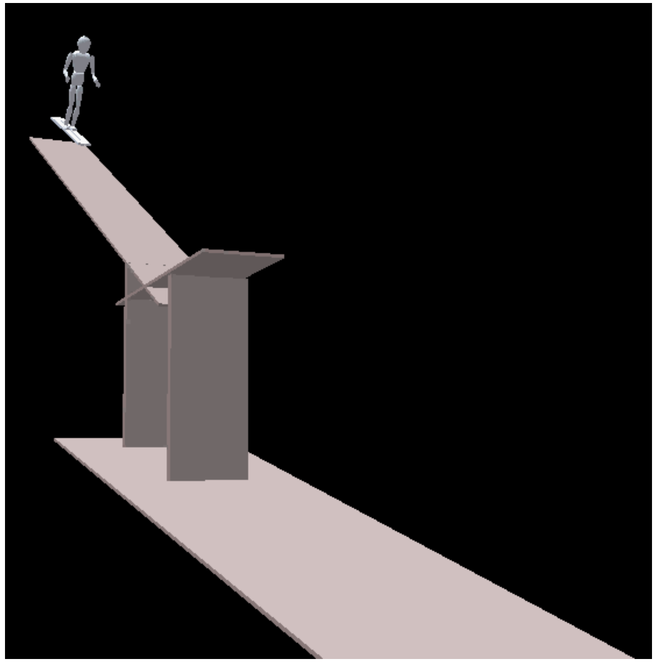


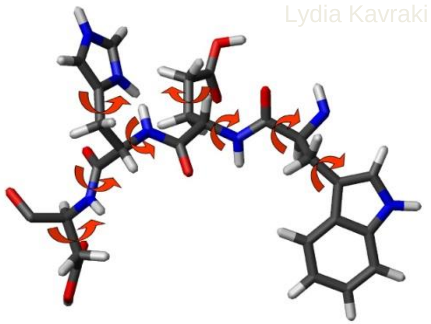
Why Motion Planning?



Virtual Prototyping

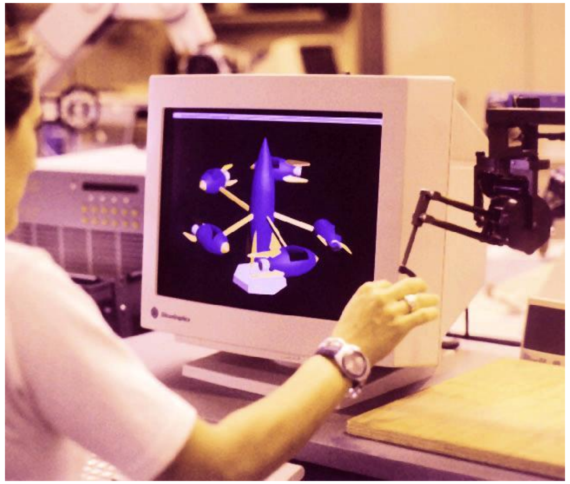
Dynamic Simulation



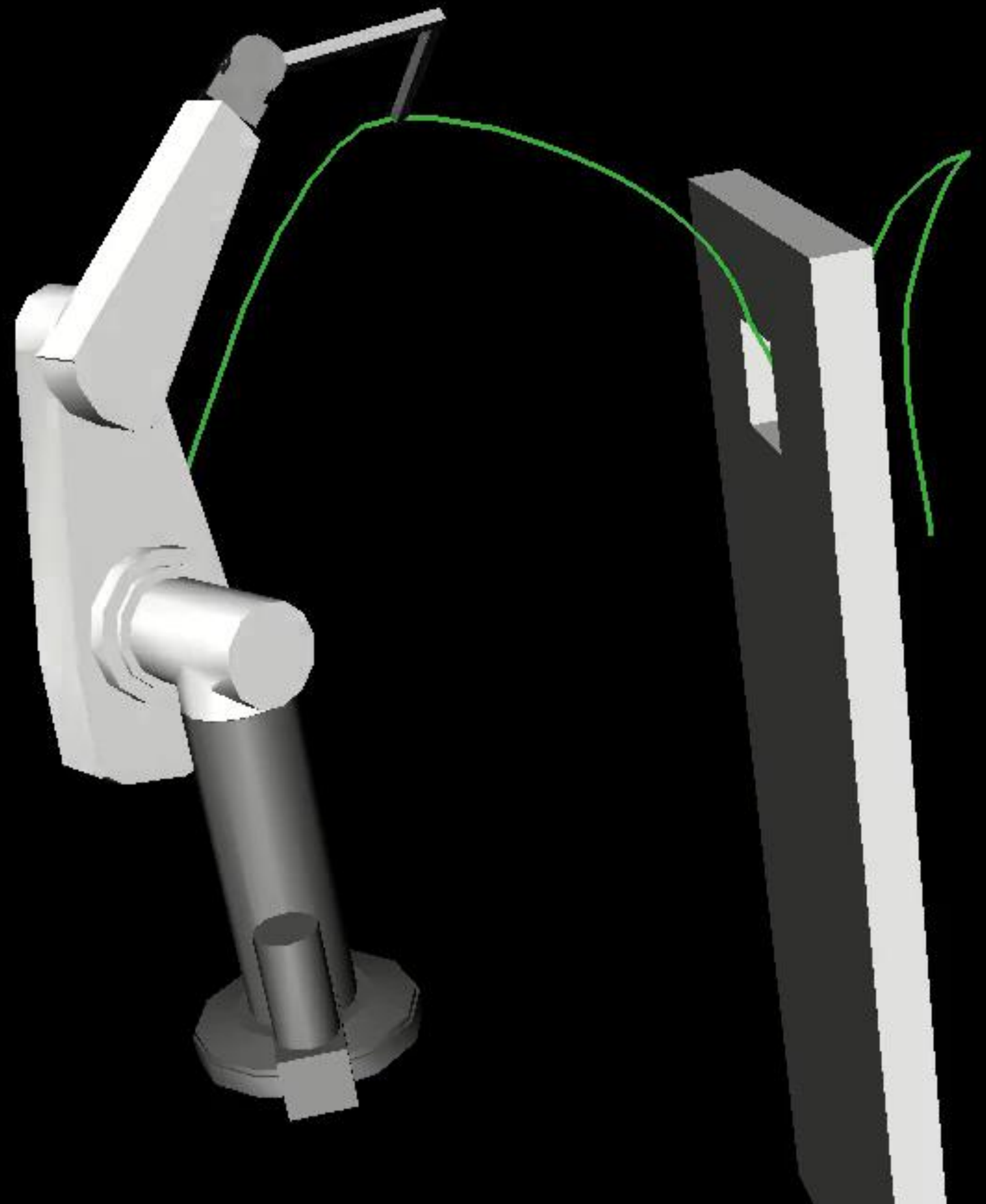


Molecular Biology

Haptics/Teleoperation



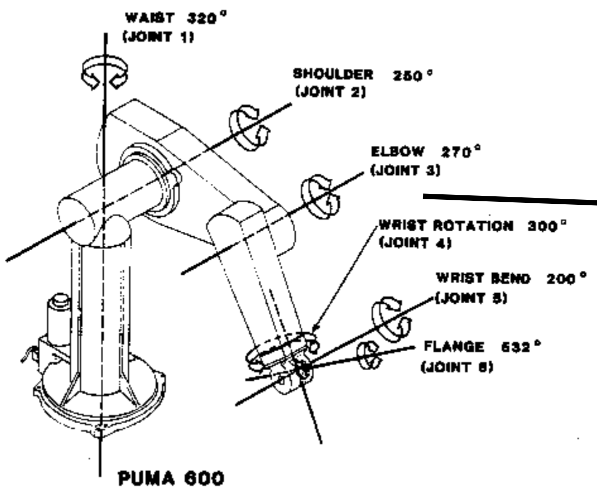
Character Animation
for Games and Movies



The Motion Planning Problem

- R : robot with n degrees of freedom
- $B_{1,\dots,m}$: obstacles, rigid objects
- W : workspace R^2 or R^3
- C : configuration space
- $R, B_{1,\dots,m} \in W$
- $q_{\text{initial}}, q_{\text{final}}$
- Find a free path τ so that R moves from q_{initial} to q_{final} without intersecting with any B_i

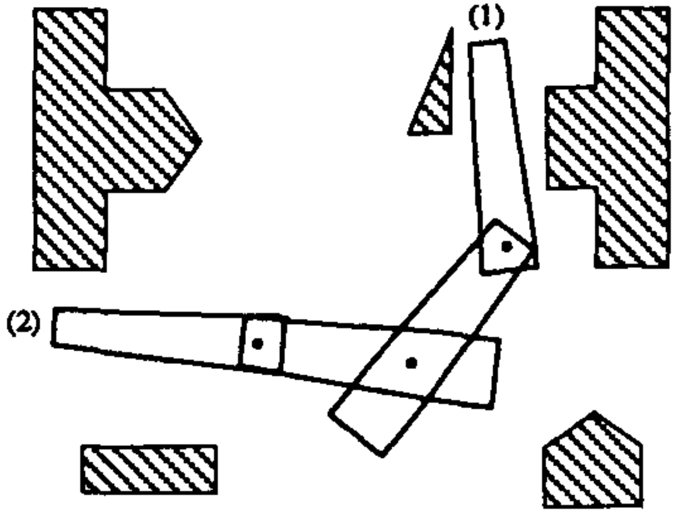
Reminder: Configuration Space

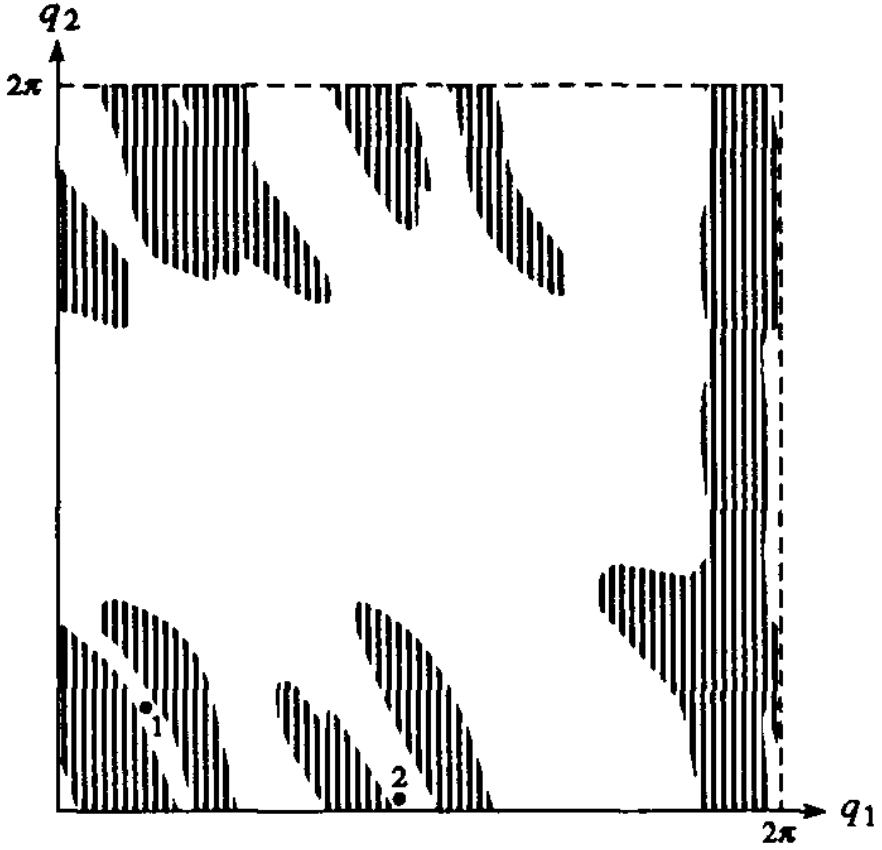


$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{bmatrix}$$

- Configuration: *Minimal* set of parameters *uniquely* describing R
- $q \in \mathrm{R}^n$ so that robot is represented as a point
- How do we represent the world in R^n ?

World in C-Space





Definitions of C-Space Concepts

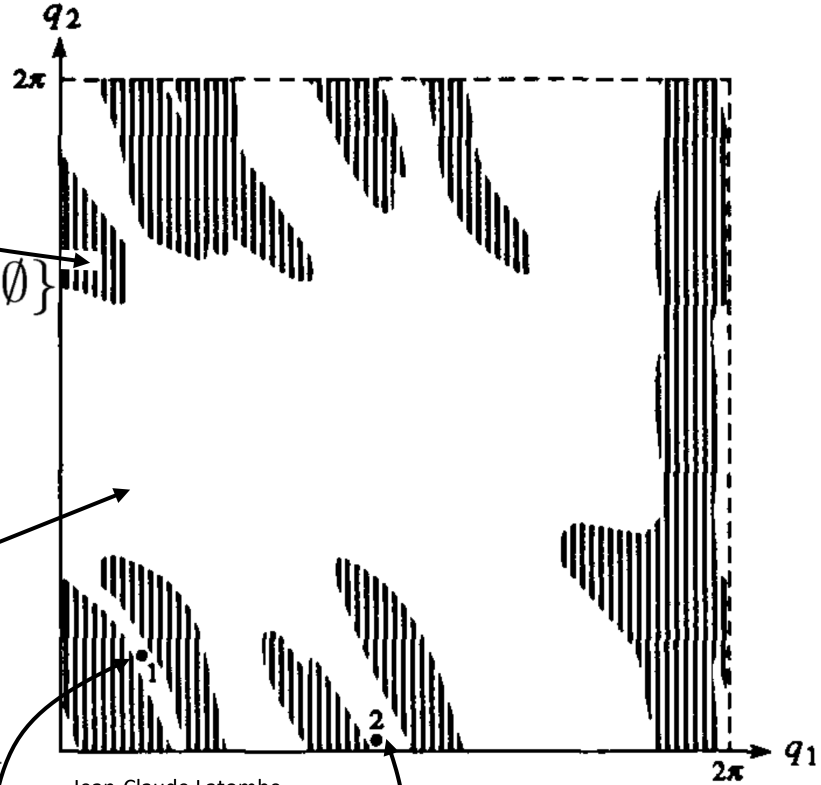
How can we obtain C_{obst}?

$$C_{\text{obst}} = \{\mathbf{q} \in C \mid \exists i : R(\mathbf{q}) \cap B_i \neq \emptyset\}$$

$$C_{B_i} = \{\mathbf{q} \in C \mid R(\mathbf{q}) \cap B_i \neq \emptyset\}$$

How can we obtain C_{free}?

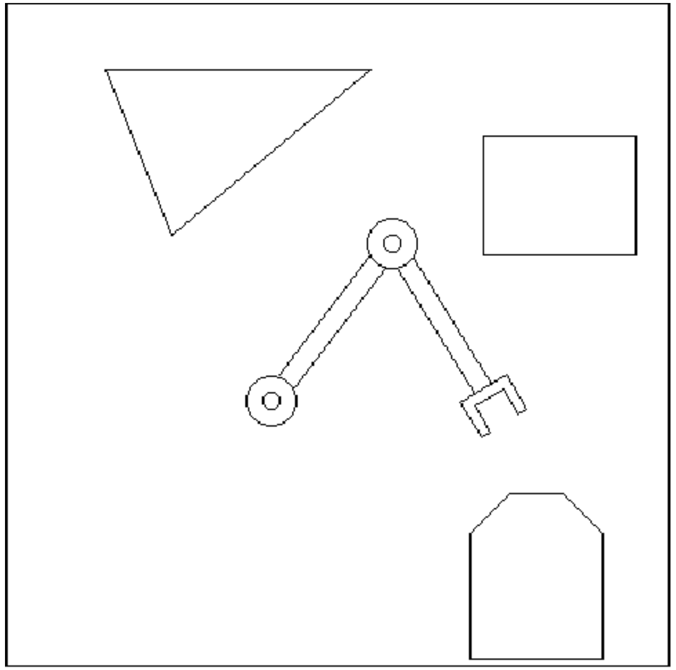
$$\begin{aligned} C_{\text{free}} &= C \setminus C_{\text{obst}} \\ &= \{\mathbf{q} \in C \mid R(\mathbf{q}) \cap (\cup_i B_i) = \emptyset\} \end{aligned}$$

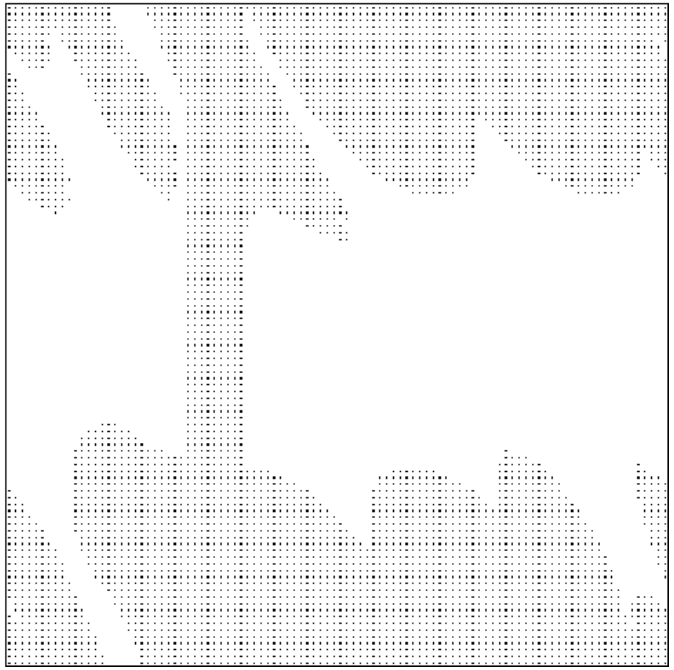


Jean-Claude Latombe

How can we obtain a path from 1 to 2?

Another World in C-Space





Sean Quinlan

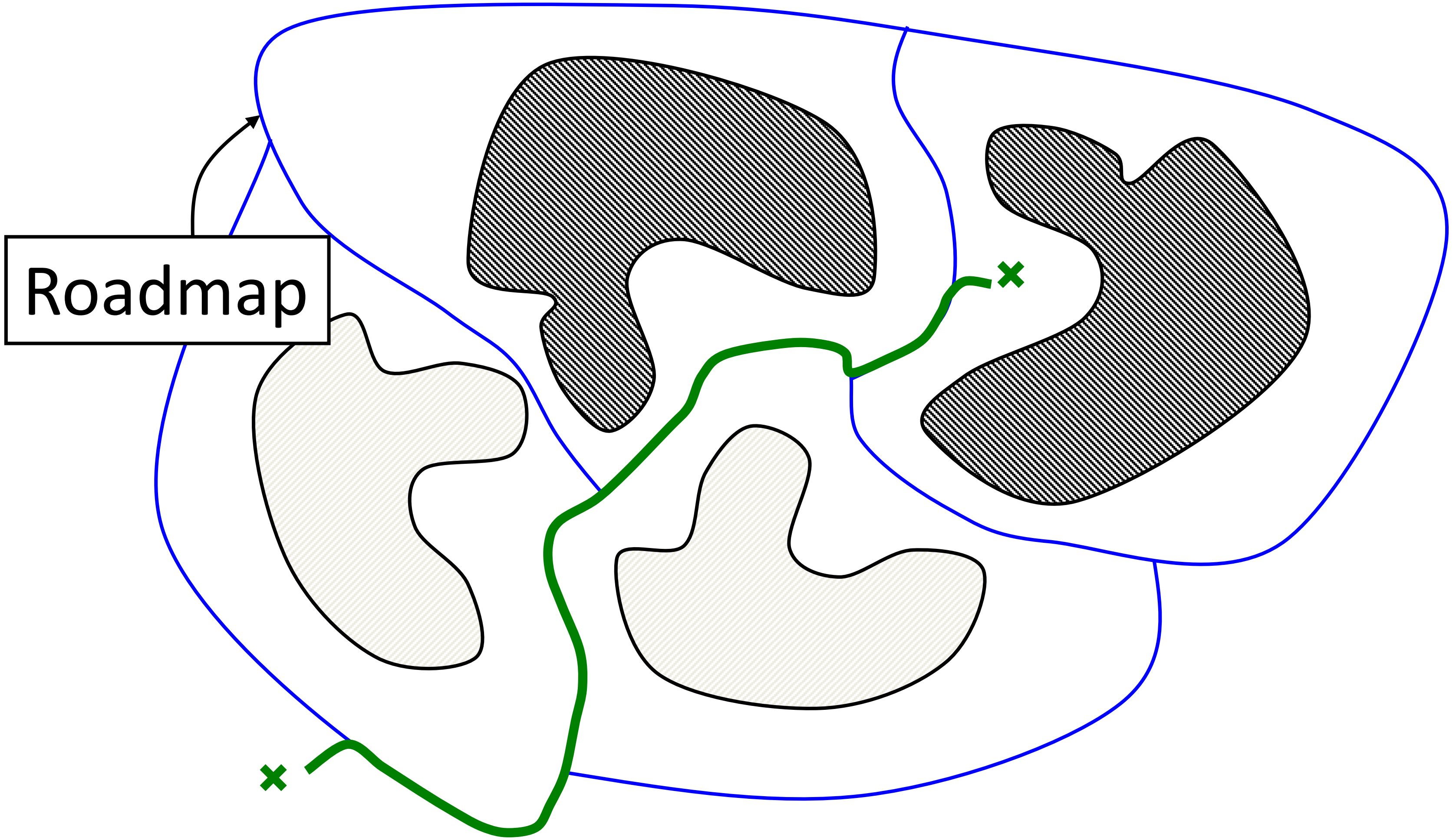
C-Space Obstacles

- Good representation: robot is a point!
- But we are left with the **BIG** problem of Motion Planning:

  C-obstacle computation

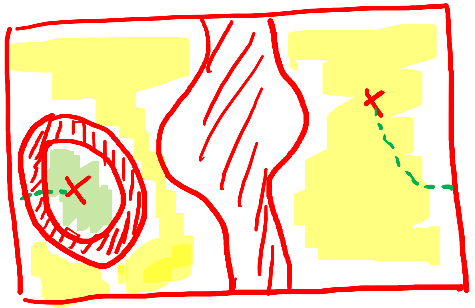
- How to translate the world into C-space?

Roadmap (Graph)



How to obtain the roadmap?

Free Path / Connected Component



A **free path** between two configuraions $\mathbf{q}_{\text{init}}$ and $\mathbf{q}_{\text{goal}}$ is a continuous map $\mathcal{P} : [0,1] \rightarrow C_{\text{free}}$ with $\mathcal{P}(0) = \mathbf{q}_{\text{init}}$ and $\mathcal{P}(1) = \mathbf{q}_{\text{goal}}$.

Two configurations belong to the same **connected component** of C_{free} if and only if they are connected by a free path.

Motion versus Path

- “Path” refers to continuous sequence of configurations
- “Trajectory” includes time parameterization
- Most people just refer to Motion Planning
- Here: Path Planning = Motion Planning
- We ignore time parameterization!

C-Space Construction for *n* dof

- Don't underestimate the problem!
  - 6 rotational degrees of freedom
  - 360 degrees of orientation per joint
  - 2×10^{15} configurations
  - 1 million checks per second
  - 69 years of computation
- Naïve grid methods are **computationally intractable** if d is larger than a very small constant!