



지능형 로봇 소프트웨어 무결점 검증기 개발

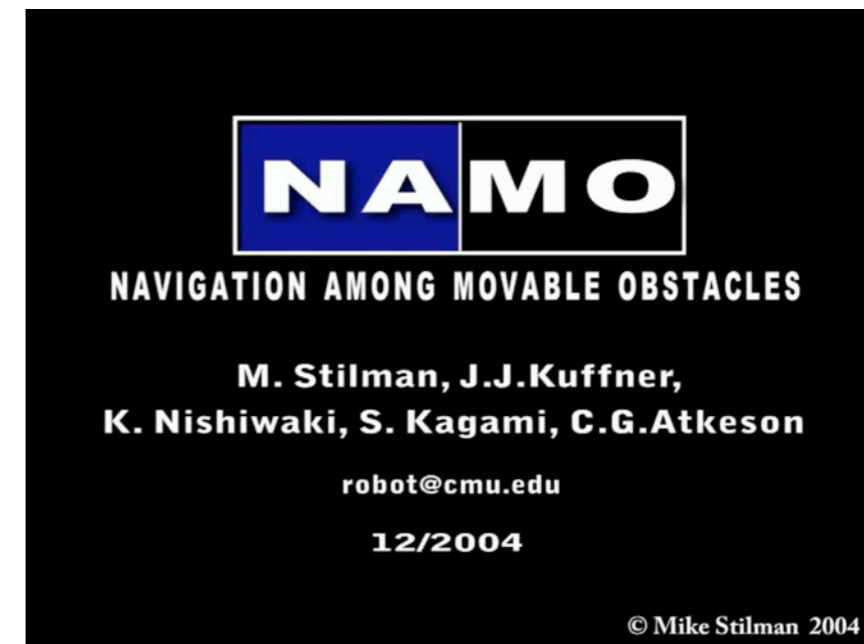
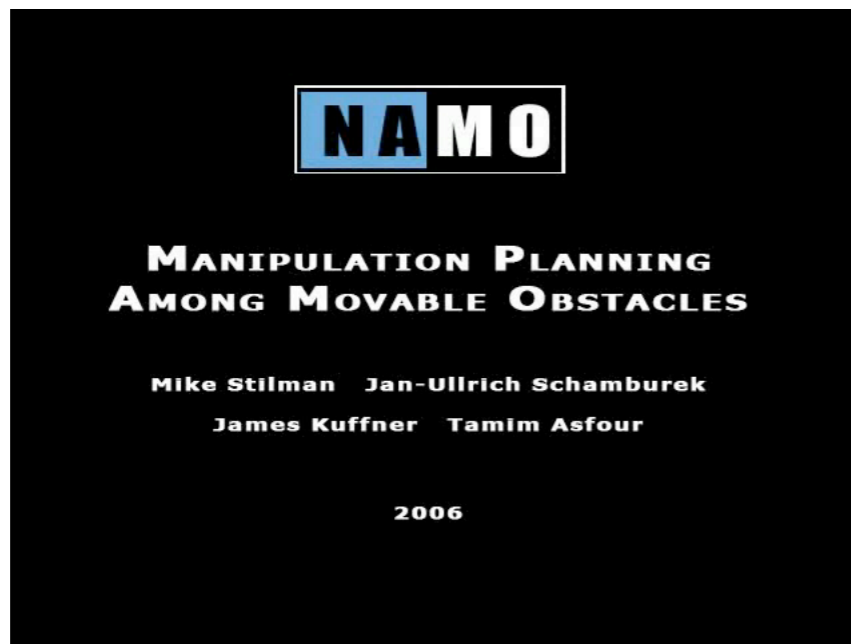
한재영

서울대학교 로봇자동화연구실



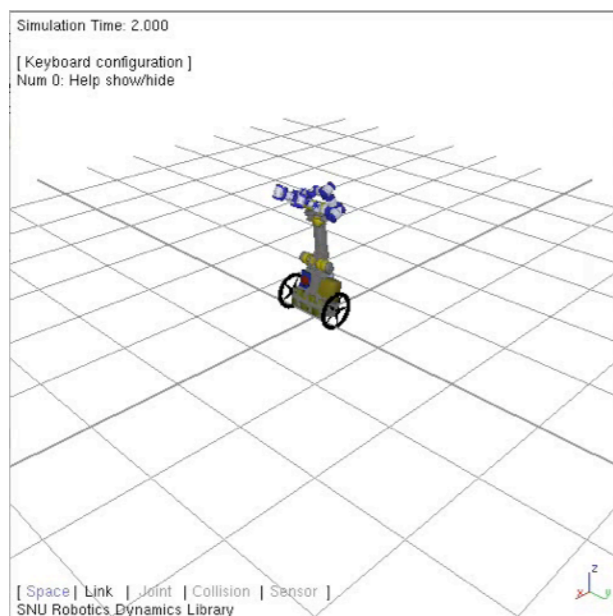
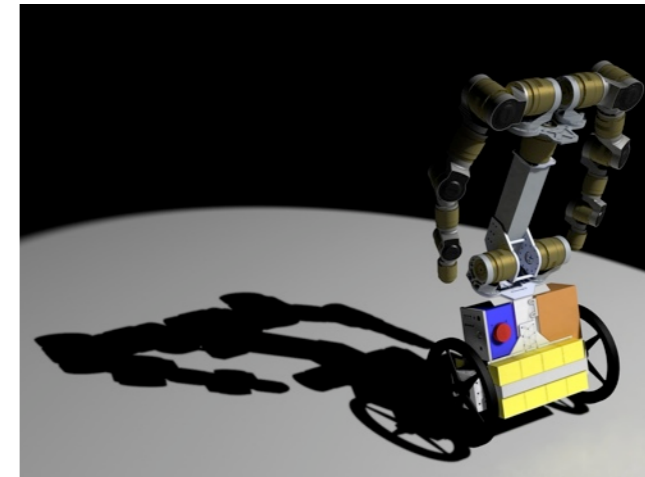
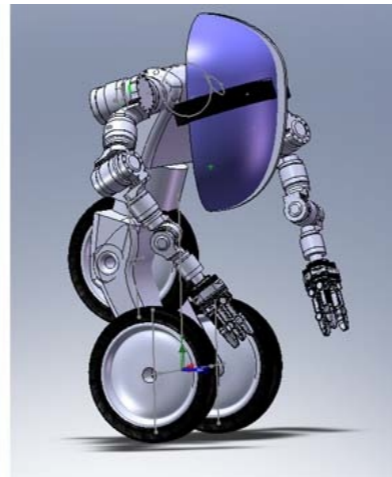
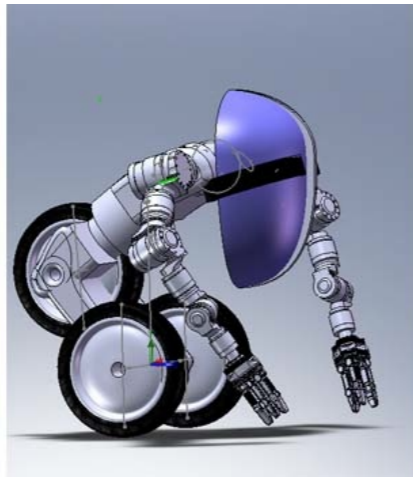
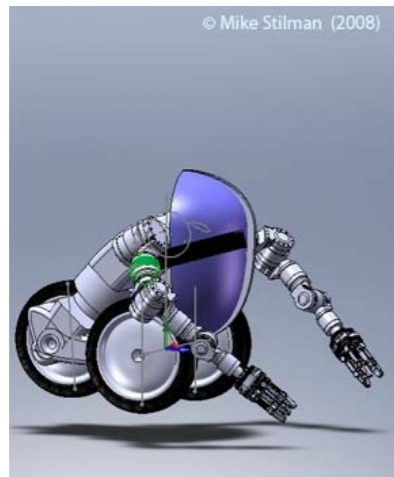
Robot Design Tool

- Mike Stilman, RIM @ Georgia Tech
 - Research field: Motion planning





Designing and Building a Robot





Consensus on Robotics Research

- “Robotics”: Wide spectrum
 - 다양한 분야의 선행지식을 요구.
 - 높은 진입장벽.
 - 선행지식이 많이 필요하나 반드시 알아야 하는 것도 아님.



Consensus on Robotics Research

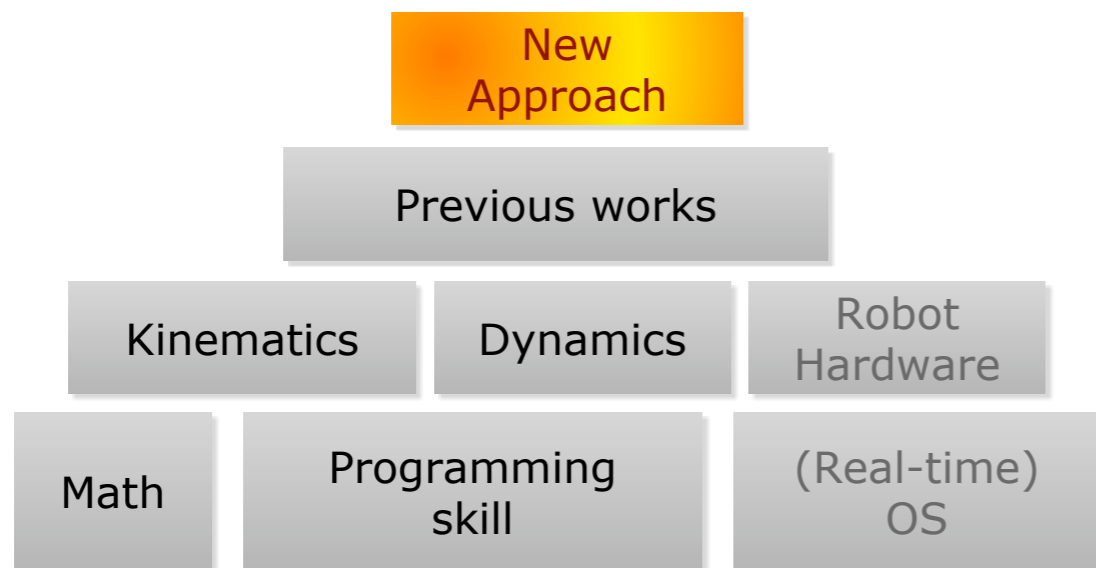
- “Robotics”: Wide spectrum
 - 다양한 분야의 선행지식을 요구.
 - 높은 진입장벽.
 - 선행지식이 많이 필요하나 반드시 알아야 하는 것도 아님.

New
Approach

Previous works

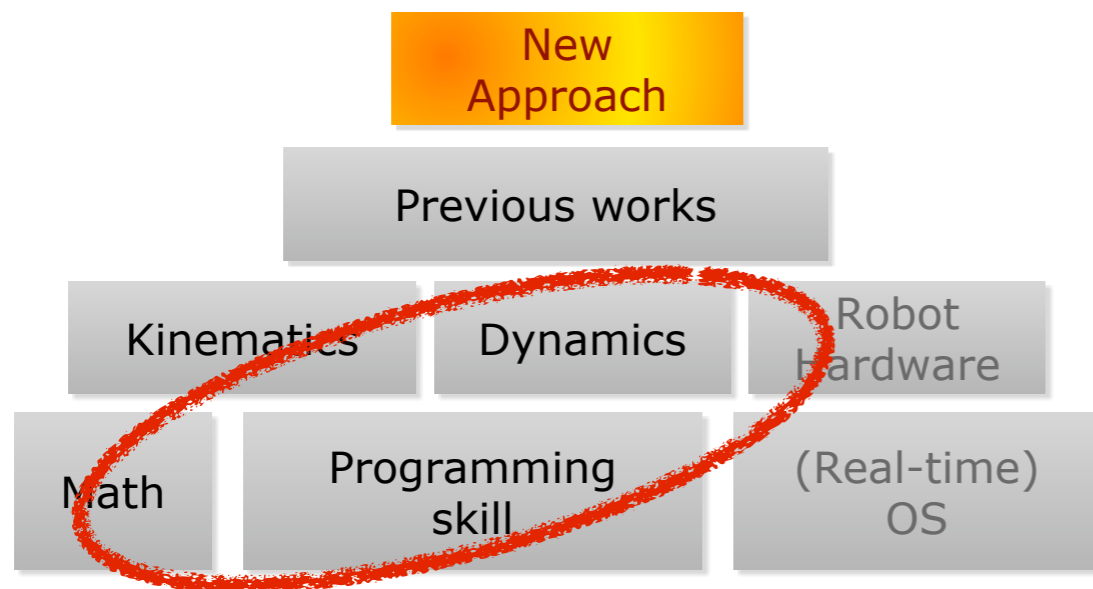
Consensus on Robotics Research

- “Robotics”: Wide spectrum
 - 다양한 분야의 선행지식을 요구.
 - 높은 진입장벽.
 - 선행지식이 많이 필요하나 반드시 알아야 하는 것도 아님.



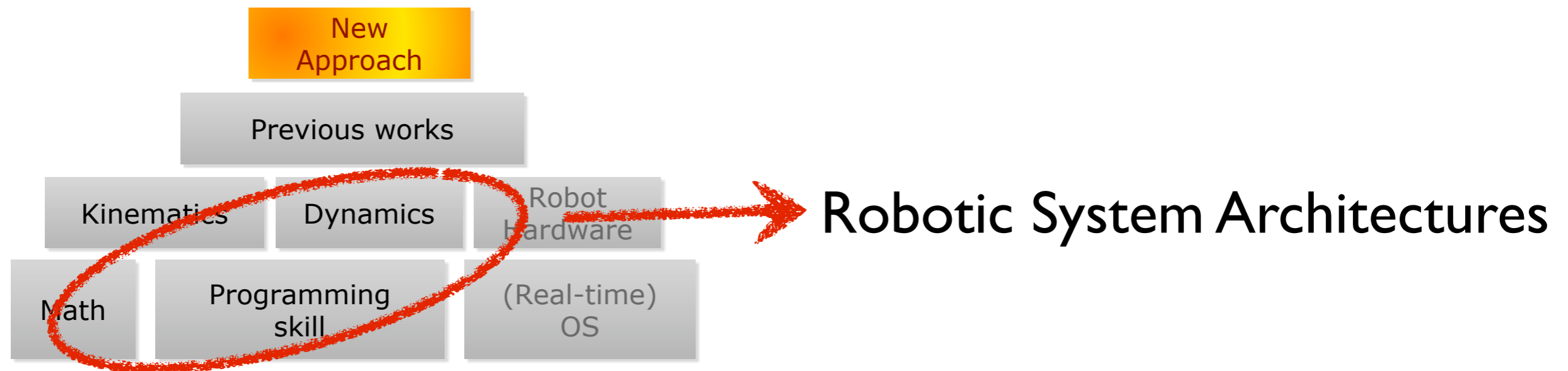
Consensus on Robotics Research

- “Robotics”: Wide spectrum
 - 다양한 분야의 선행지식을 요구.
 - 높은 진입장벽.
 - 선행지식이 많이 필요하나 반드시 알아야 하는 것도 아님.



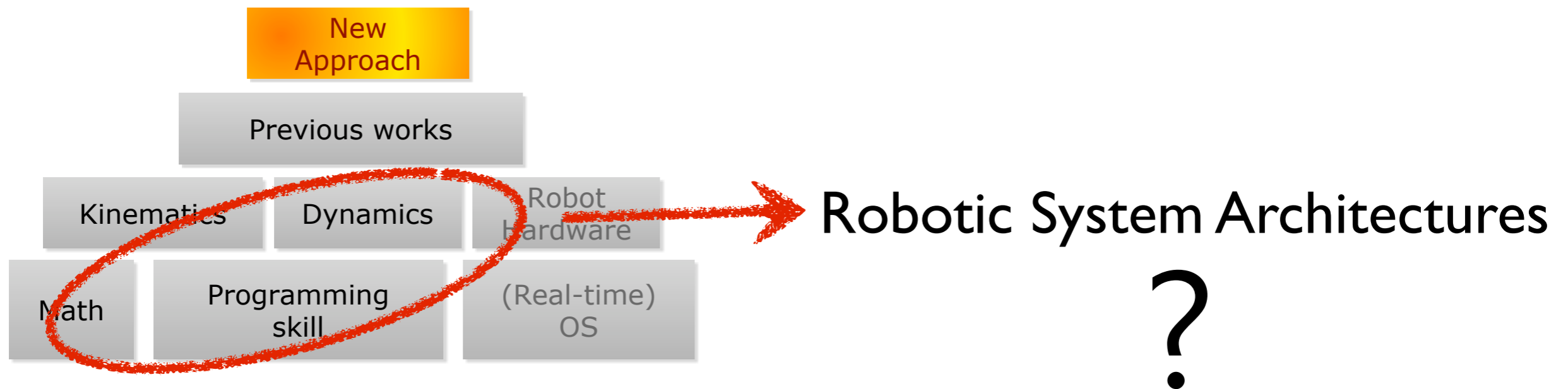
Consensus on Robotics Research

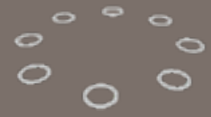
- “Robotics”: Wide spectrum
 - 다양한 분야의 선행지식을 요구.
 - 높은 진입장벽.
 - 선행지식이 많이 필요하나 반드시 알아야 하는 것도 아님.



Consensus on Robotics Research

- “Robotics”: Wide spectrum
 - 다양한 분야의 선행지식을 요구.
 - 높은 진입장벽.
 - 선행지식이 많이 필요하나 반드시 알아야 하는 것도 아님.





Robotic System Architectures

- Advantages
 - 진입장벽을 낮출 수 있다.
 - 사용자가 특정 문법을 따르게 강제함으로써 복잡성을 감춘다.
 - 코드 재사용이 가능하므로 지식/기술의 축적을 기대할 수 있다.
- Disadvantages
 - 아키텍처를 익히는 것 자체가 또 하나의 장벽이 될 수 있다.
 - 개발 당사자에 의한 아키텍처의 지속적인 관리 문제.
 - 최초 개발 목적에 맞지 않는 분야로의 활용이 용이치 않을 수 있다.
 - 대부분이 open source/free software로서 또다른 open source에 대한 의존성 문제.
- 아키텍처를 활용한 로봇 컨트롤 소프트웨어들은 찾기 쉽지 않음.
- 한정된 그룹에서 사용.



Motion Planner & Bugs



What is “Motion Planning”

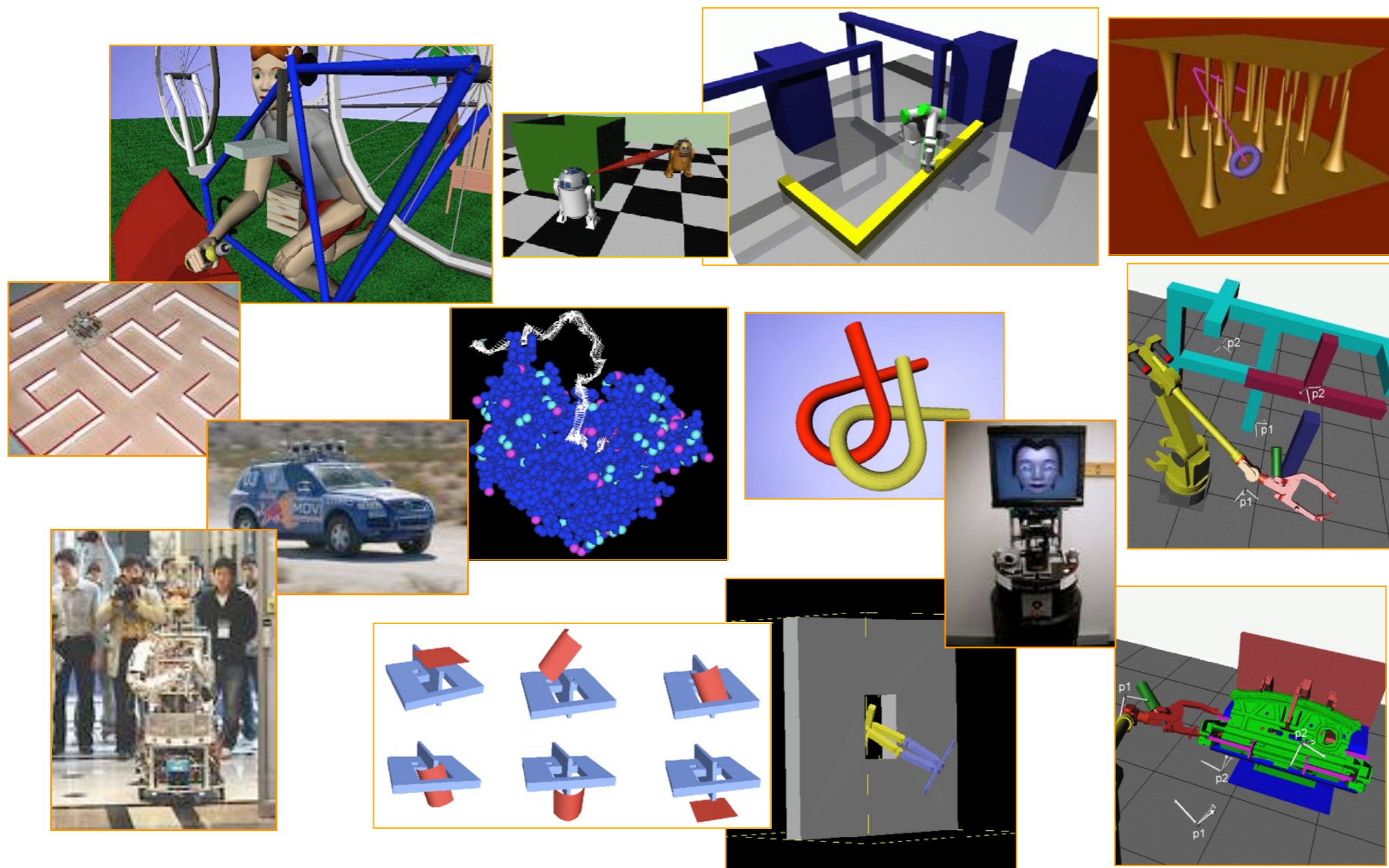
- Plan a sequence of motions in order to achieve a given task.
- Compute a collision-free path for a rigid or articulated object (the robot) among static obstacles
- Inputs:
 - Geometry of robot and obstacles
 - Kinematics of robot (degrees of freedom)
 - Initial and goal robot configurations (placements)
- Outputs:
 - Continuous sequence of collision-free robot configurations connecting the initial and goal configurations



What is “Motion Planning”



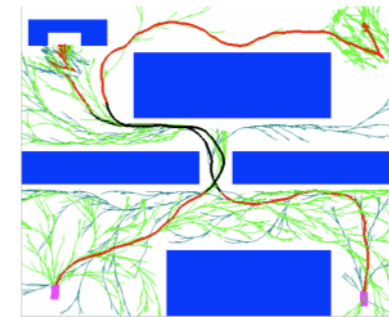
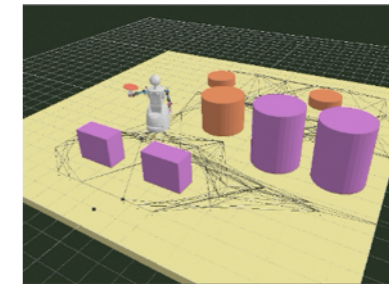
What is "Motion Planning"





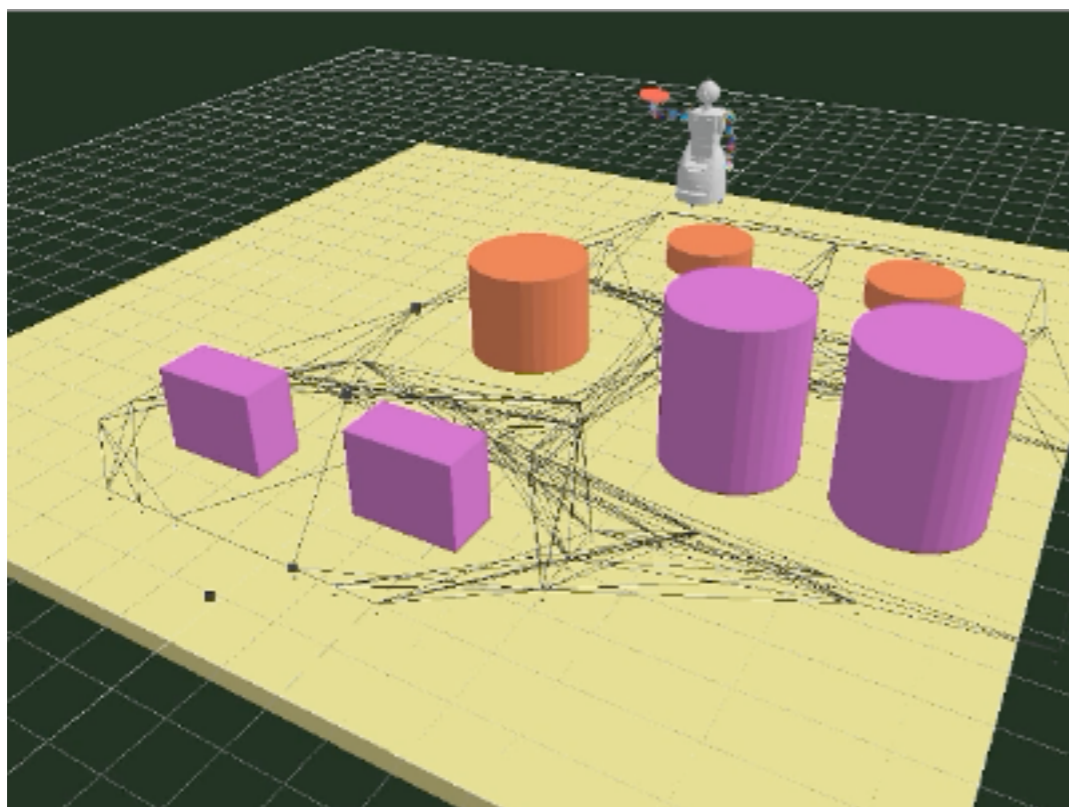
Extensions to the Basic Problem

- Moving obstacles.
- Multiple robots.
- Movable objects.
- Optimal planning.
 - shortest path/time, minimum energy.
- Uncertainty in environment.



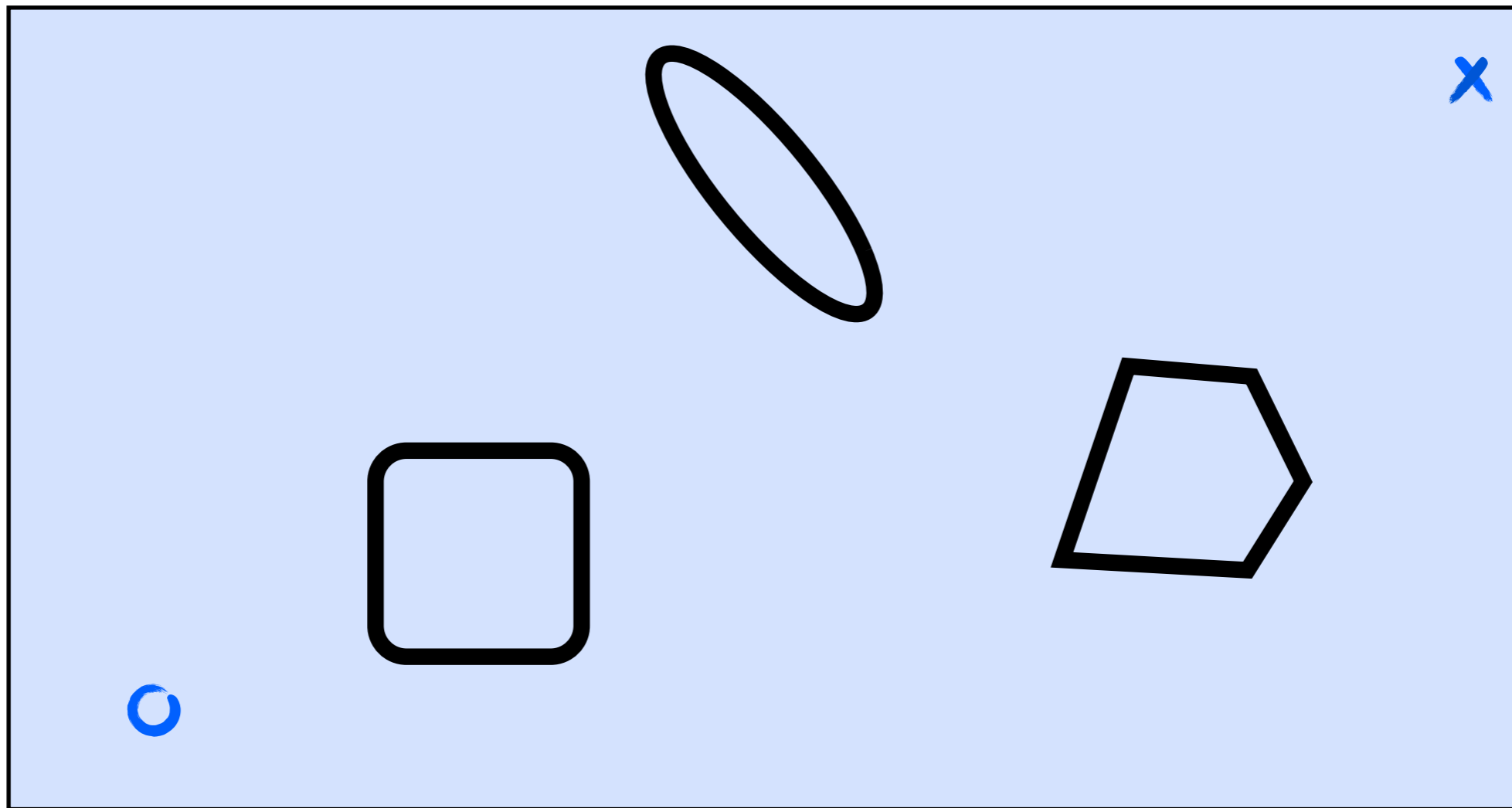


Motion Planning Demo.



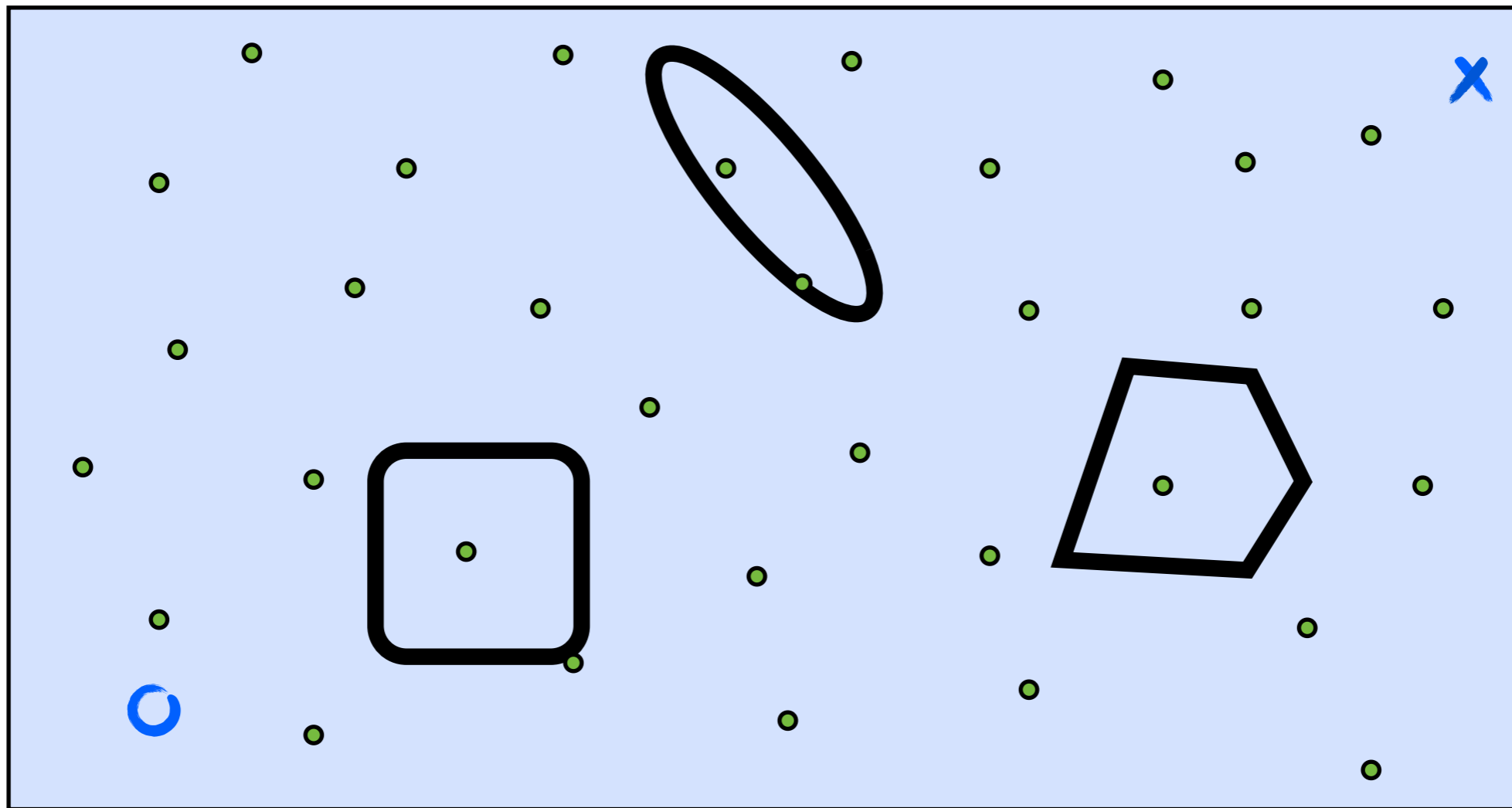


Example: Probabilistic Roadmap Method



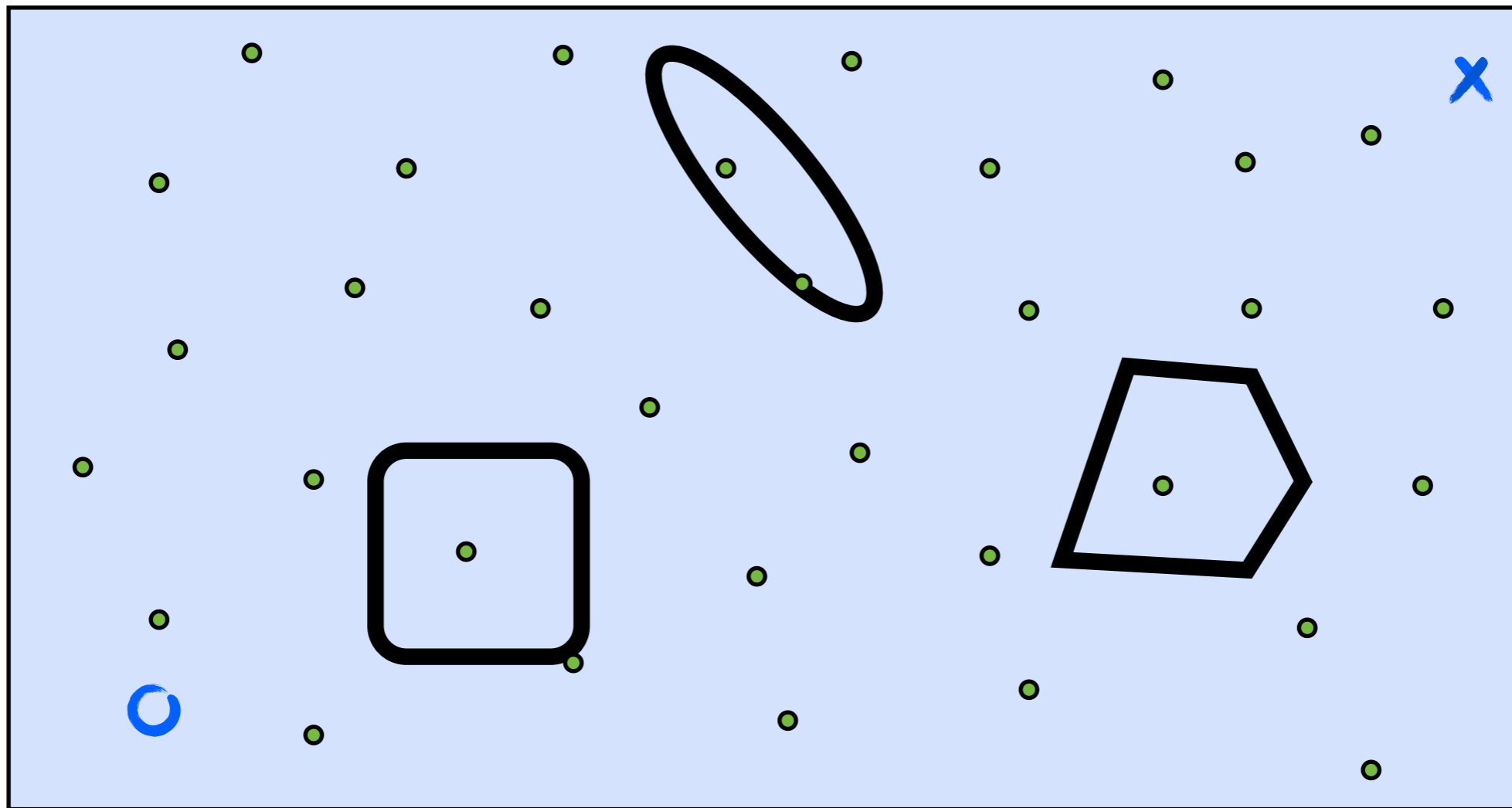


Example: Probabilistic Roadmap Method



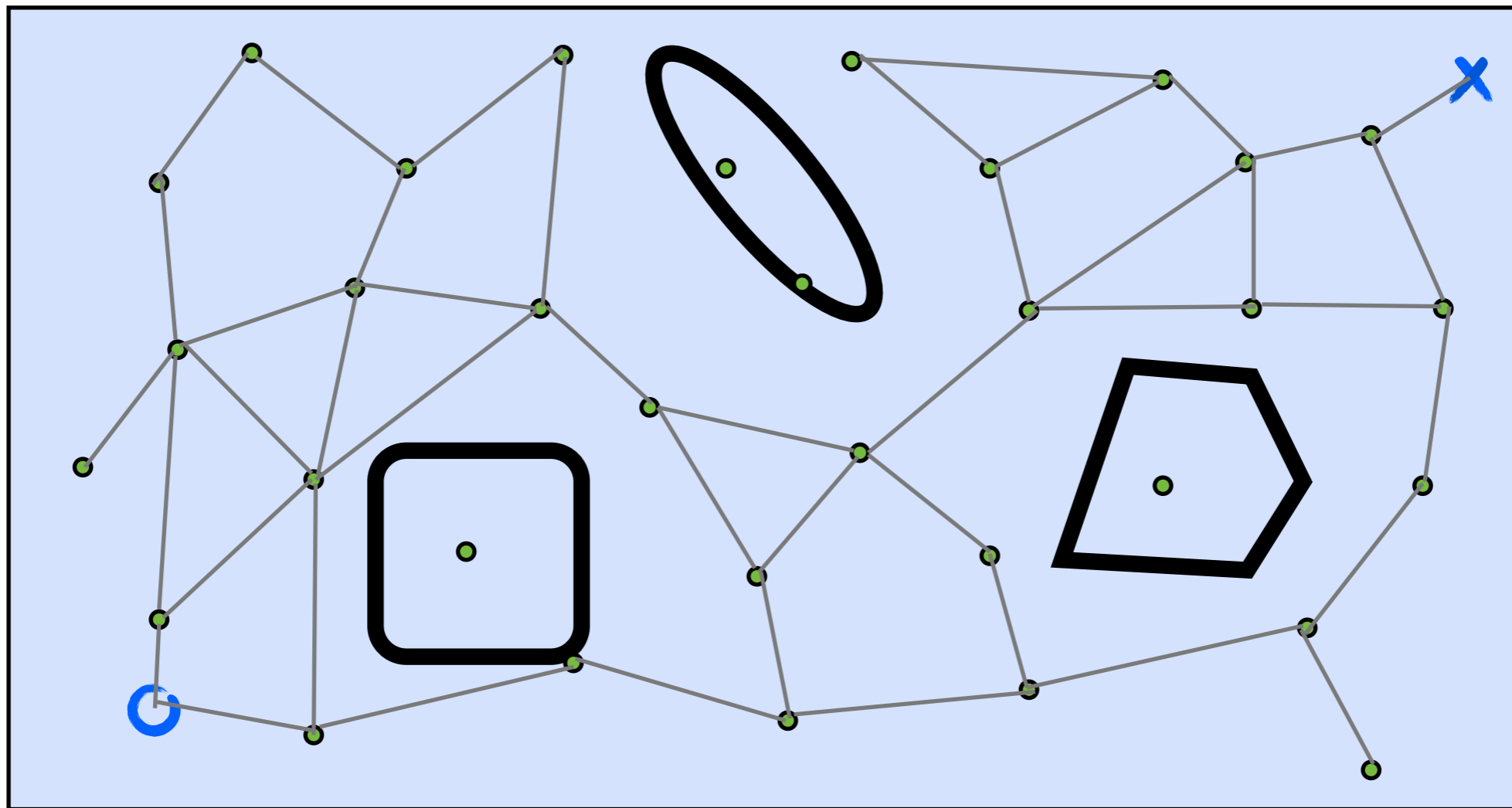


Example: Probabilistic Roadmap Method



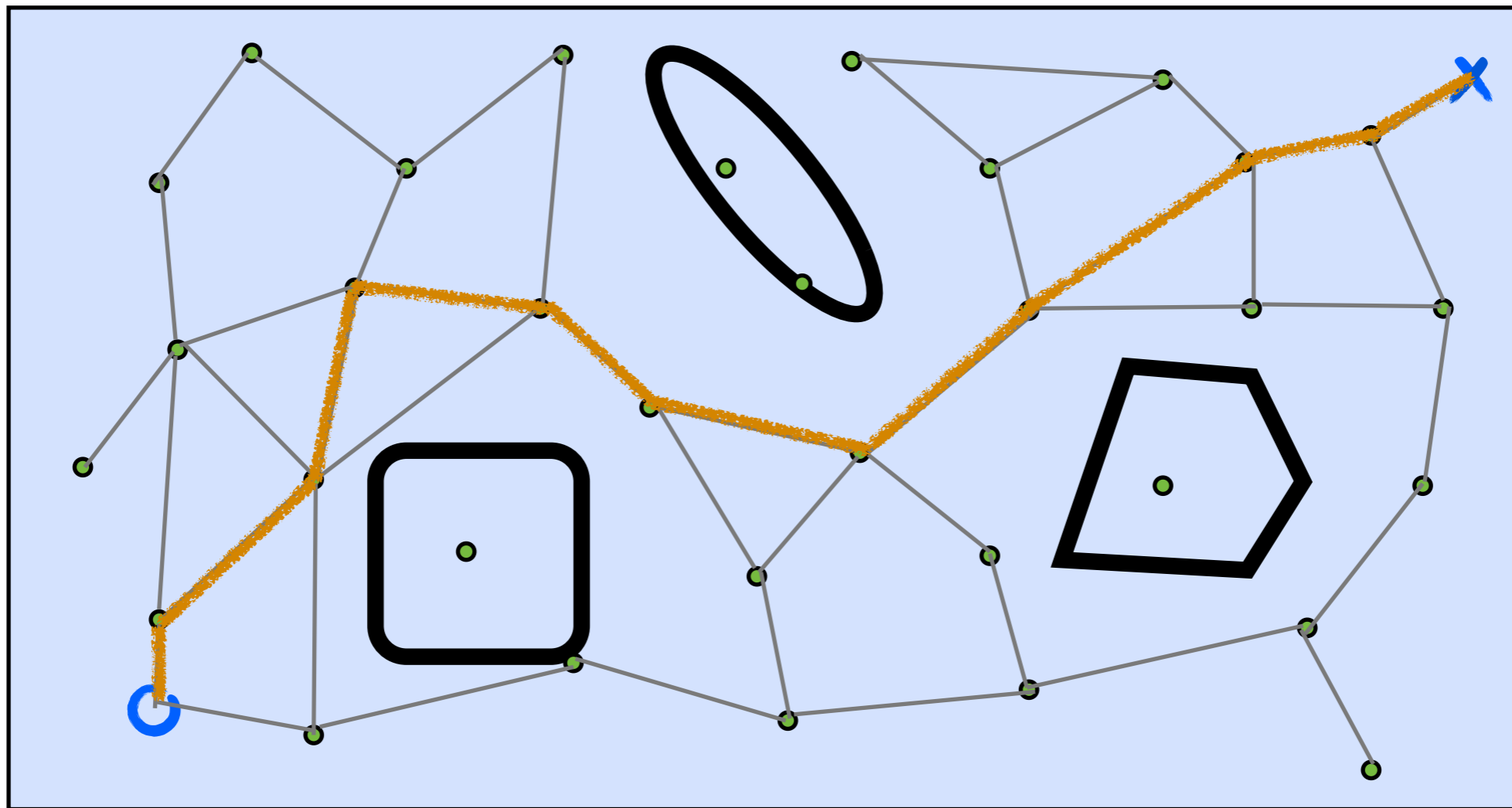


Example: Probabilistic Roadmap Method





Example: Probabilistic Roadmap Method





Frequent Bugs

- Buffer overrun/memory leak/null dereference
 - Many random points.
 - complex environment
 - high-dimensional space planning
 - Frequent addition/removal.
 - vector, list structure
- Divide-by-zero
 - Matrix inversion.
 - Inverse kinematics problem
- Over/underflow
 - Odometer.

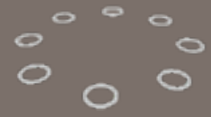


Frequent Bugs

- Un-initialized memory access.
- Ill-ordered function call.
 - Cause un-initialized memory access.
- Ill-posed robot. (abnormal joint values)
 - Simulation is essential.

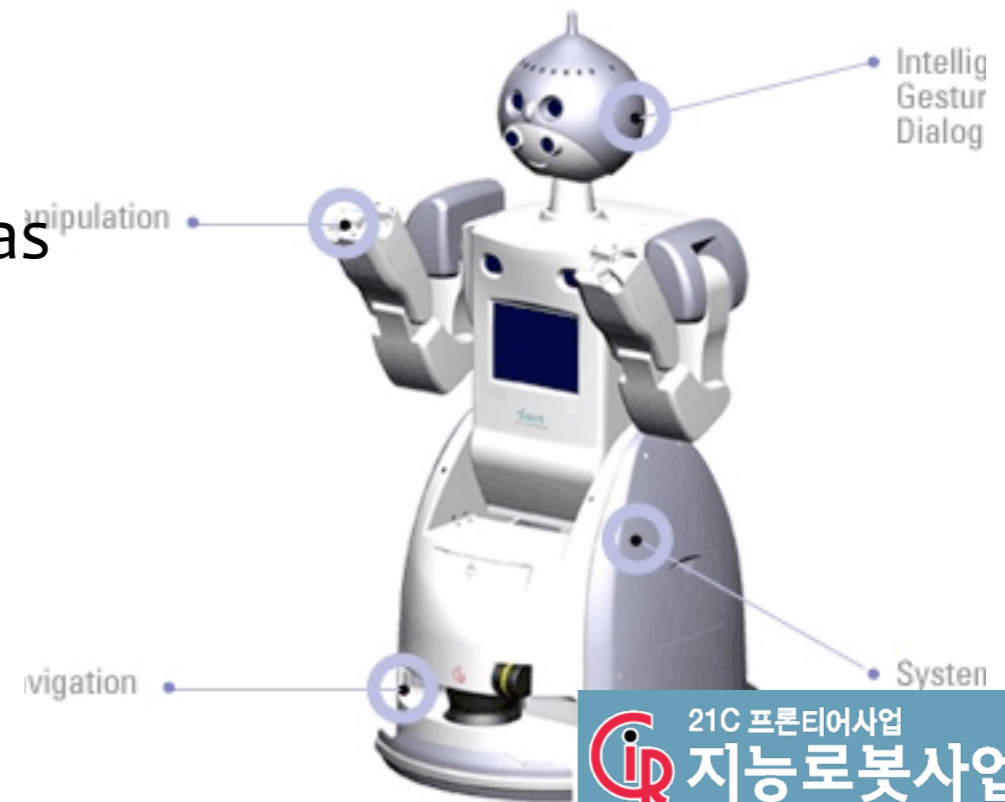


Multiple Module Integration



Target Hardware: CIROS

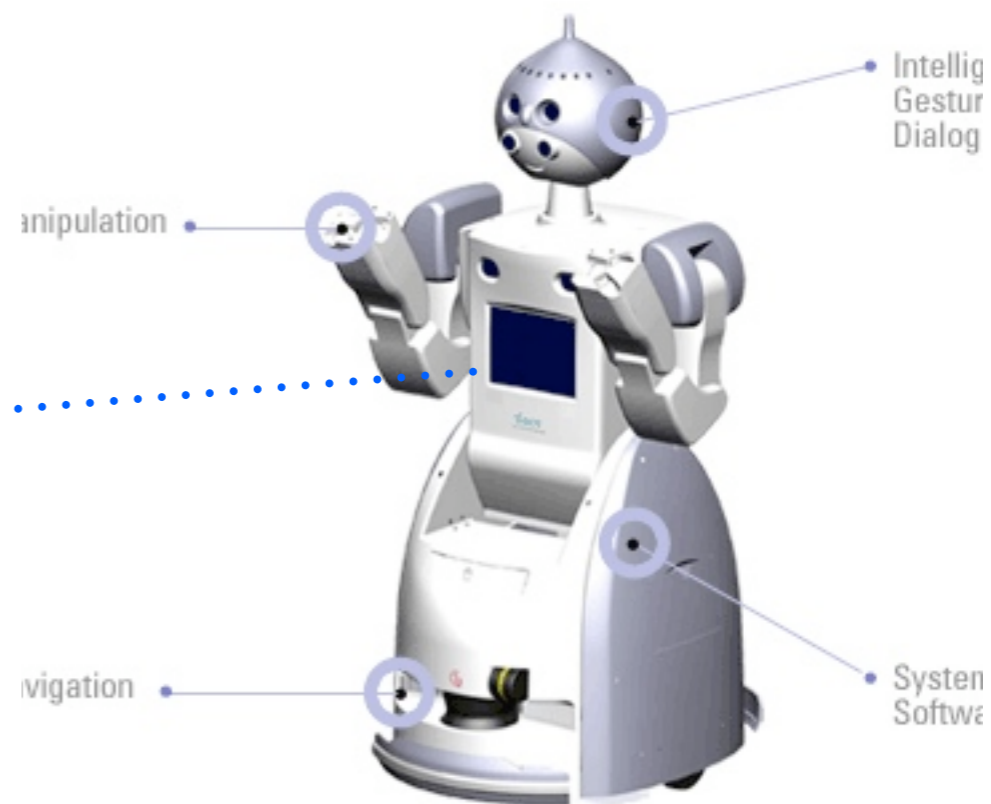
- Service robot for the elder.
 - Dependable manipulation.
 - Dependable navigation.
 - Recognition, etc.
- At least, robot should be as intelligent as 5-year-old child.
- Two 7-DOF arms
- Two 3-fingered hands
- Two-differential-wheel mobile base.





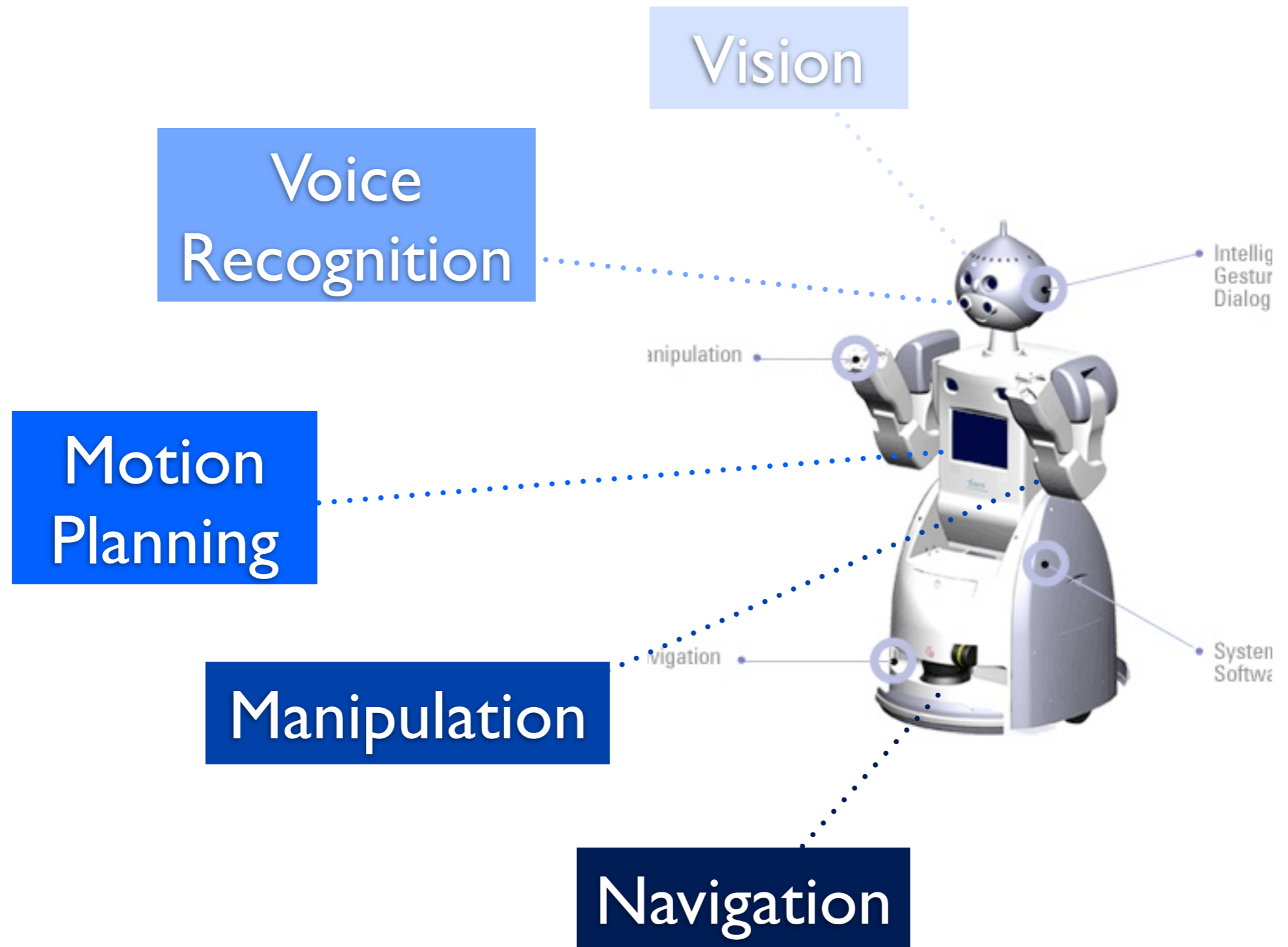
Cooperation with Multiple Modules

Motion
Planning



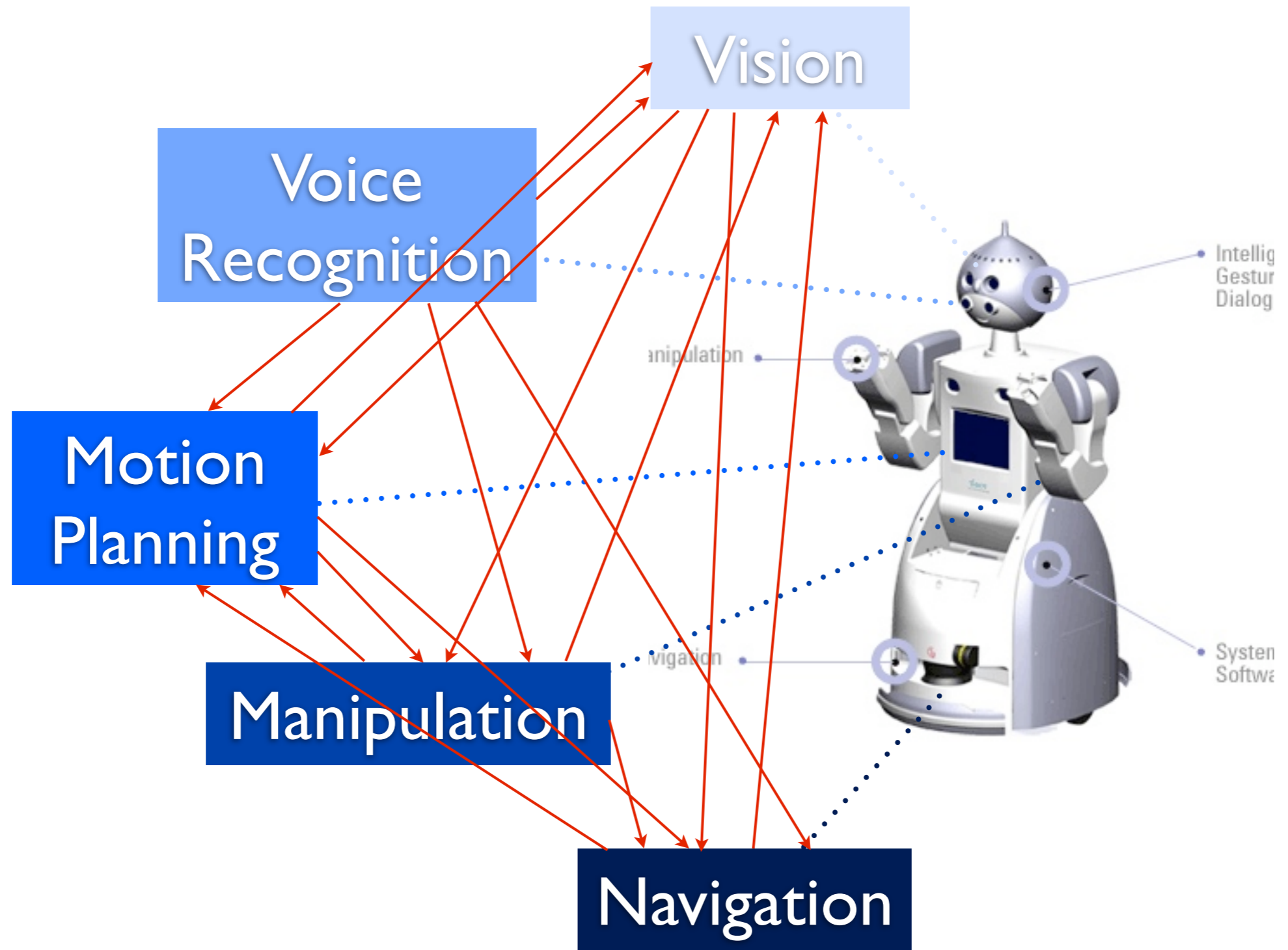


Cooperation with Multiple Modules





Cooperation with Multiple Modules





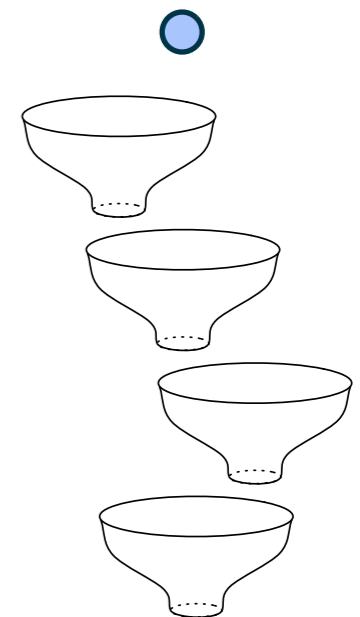
Bugs in Integration

- Disagreement of input/output data structure.
- Dependency.
- Improper use of function.
 - Ill-ordered function call.
 - Ill-conditioned arguments.
- Abnormal resultant value: ensemble of incomplete modules.
 - Hard to find out cause.



Bugs in Integration

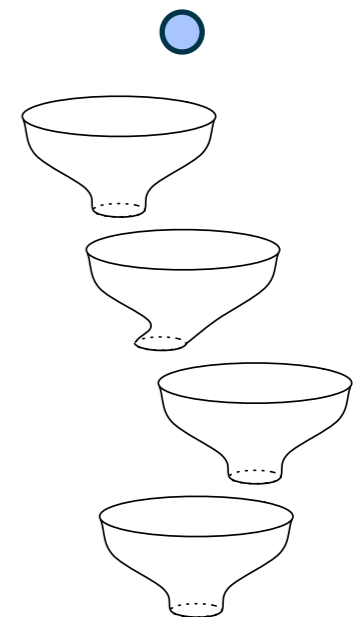
- Disagreement of input/output data structure.
- Dependency.
- Improper use of function.
 - Ill-ordered function call.
 - Ill-conditioned arguments.
- Abnormal resultant value: ensemble of incomplete modules.
 - Hard to find out cause.





Bugs in Integration

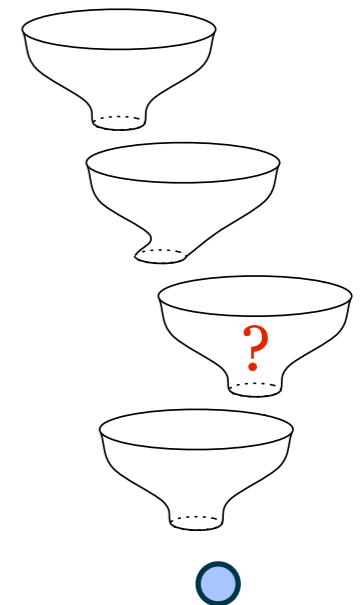
- Disagreement of input/output data structure.
- Dependency.
- Improper use of function.
 - Ill-ordered function call.
 - Ill-conditioned arguments.
- Abnormal resultant value: ensemble of incomplete modules.
 - Hard to find out cause.





Bugs in Integration

- Disagreement of input/output data structure.
- Dependency.
- Improper use of function.
 - Ill-ordered function call.
 - Ill-conditioned arguments.
- Abnormal resultant value: ensemble of incomplete modules.
 - Hard to find out cause.





Goal

- Target of this year: ‘Motion planner’ of SNU Robotics lab.
 - Safety bugs.
 - Over/underflow of joint values of a robot.
 - Ill-ordered function call.
- Very particular case,
 - But most motion planning algorithms have similar structure and sub algorithms.

