# CS 3630!

*Lecture 2: Introduction to Robotic Systems*

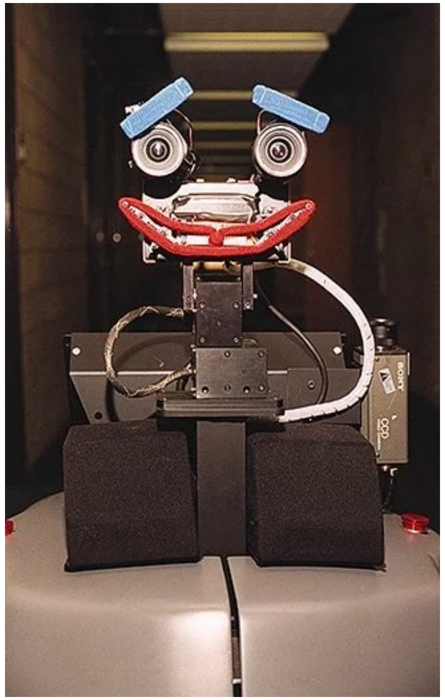
# A Taxonomy of Robotics Topics

For each module in this class, we'll consider six distinct aspects of robotics:

1. **State:** How does the robot represent its world, and itself?
2. **Actions:** What can the robot do, and how to represent this?
3. **Sensors:** What information about the world can be ascertained via sensing, and how do we model this process?
4. **Perception:** How can we combine sensor data with contextual knowledge to understand the current state?
5. **Planning:** What actions should the robot execute to transform the state of the world into a desired goal state?
6. **Learning:** How can the robot improve its knowledge over time, using information that it acquires during operation?

***Each chapter of the book includes six sections, corresponding to these topics.***

# Robots in the real world



For specific applications, these topics correspond to specific problems that robots must solve to operate effectively.

For example, a museum guide robot:

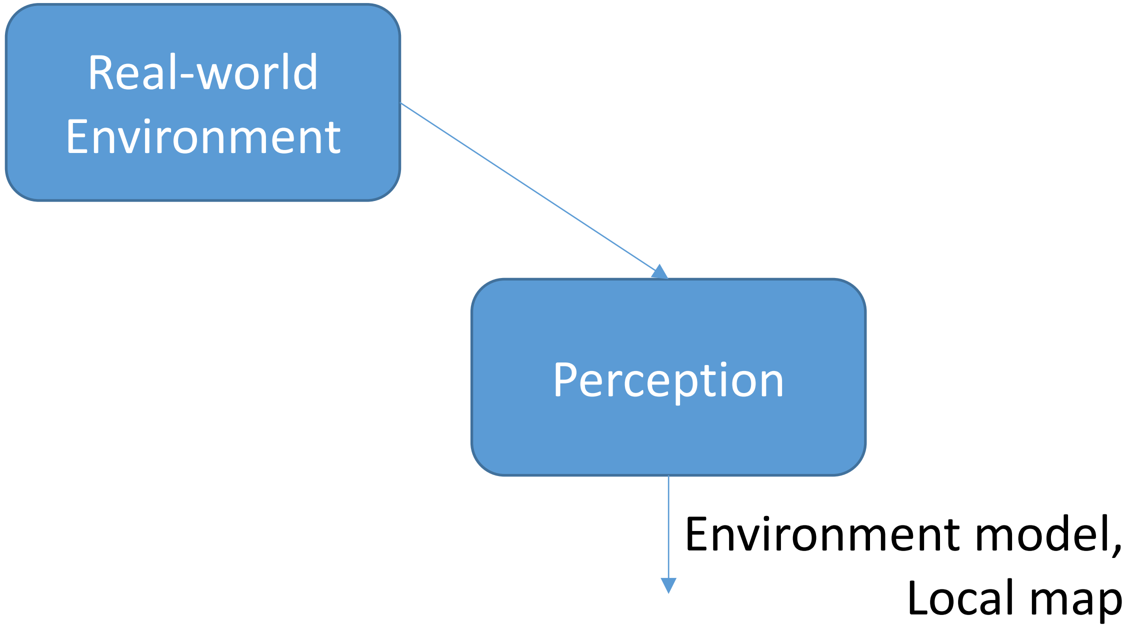
- State: where is the robot, and where are the humans to be guided?
- Actions: move from room to room
- Sensors: cameras
- Perception: use computer vision to understand human intention, and to localize
- Planning: what path to take in order to guide humans to their desired exhibit
- Learning: which parts of the museum are crowded, and when to avoid these?

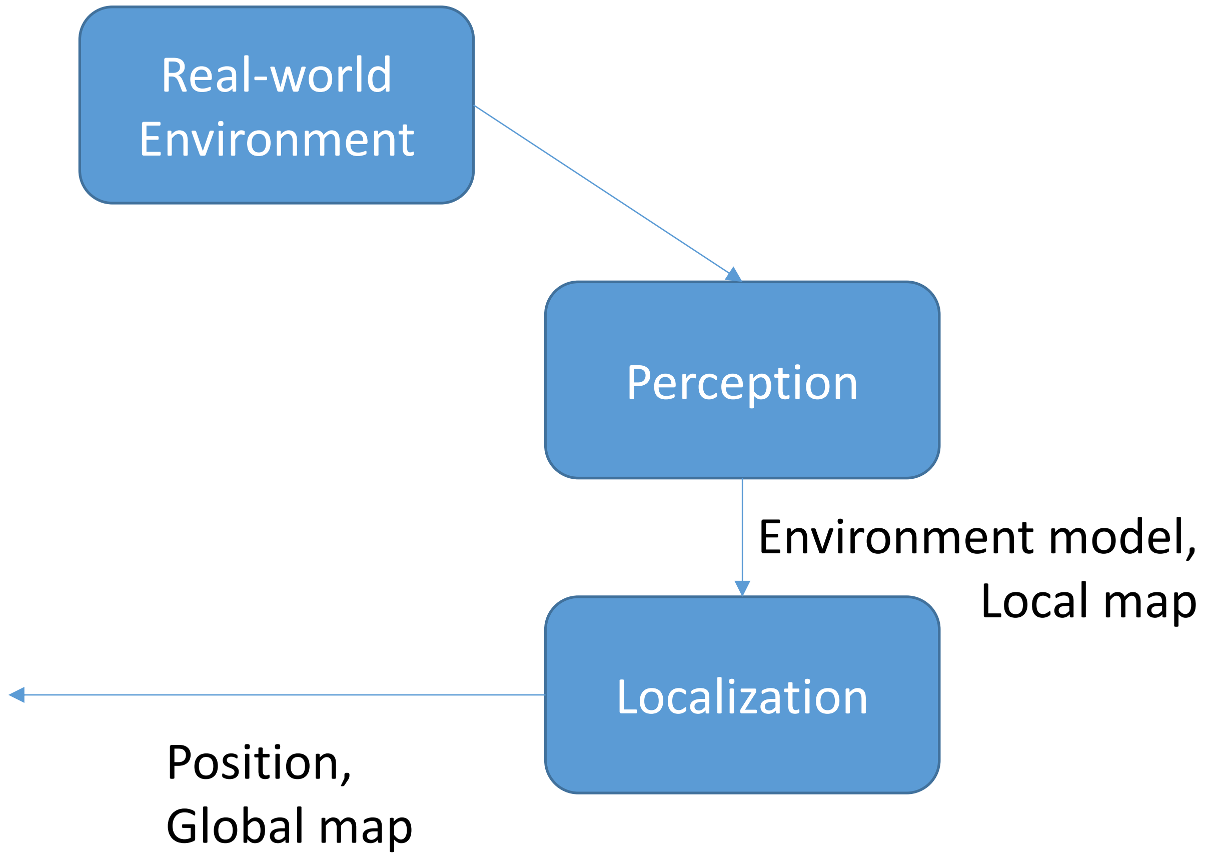
# How do robots function in the world

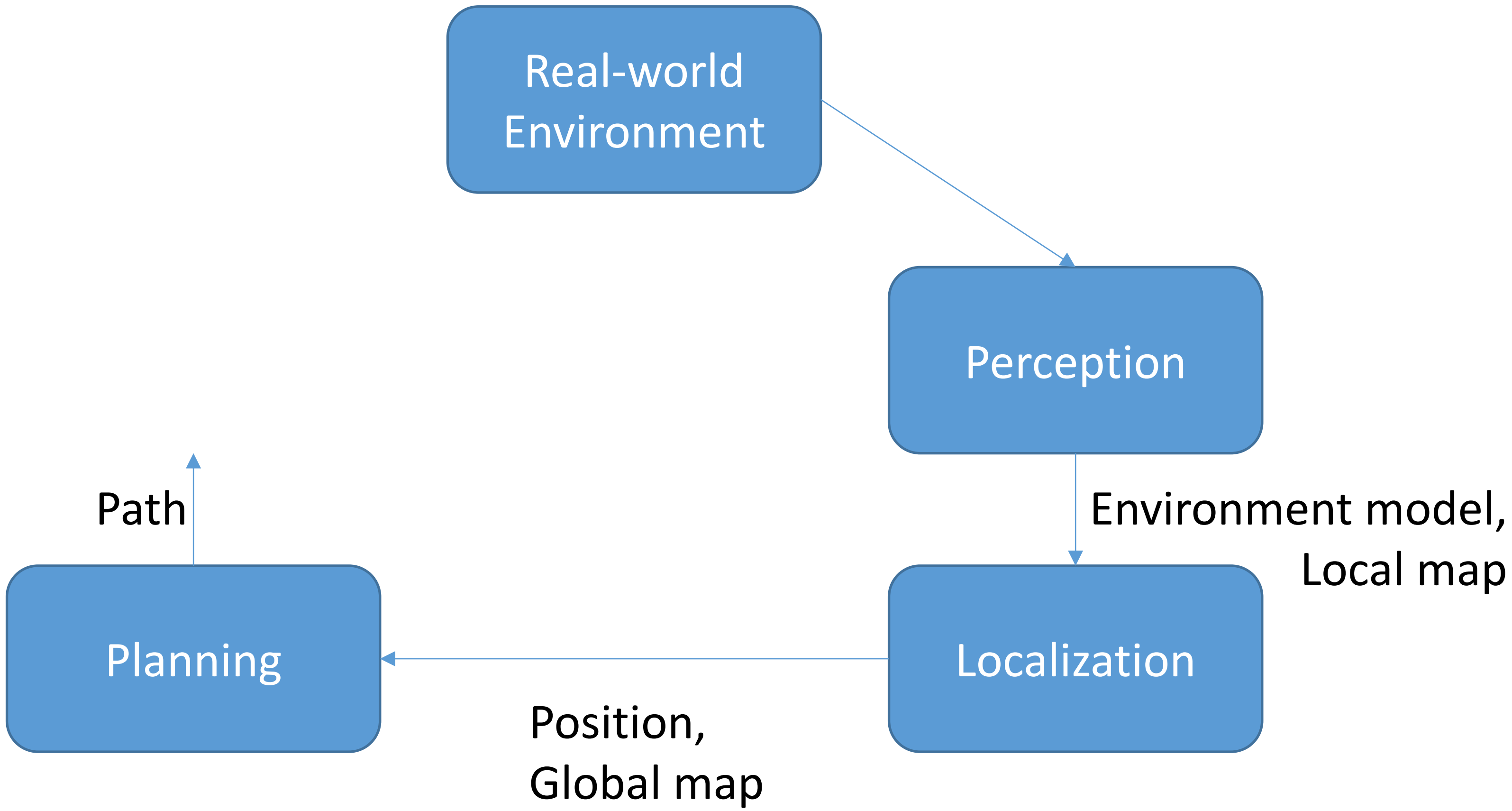
When they are deployed in the world, most robots use the so-called ***Sense-Think-Act*** paradigm of operation.

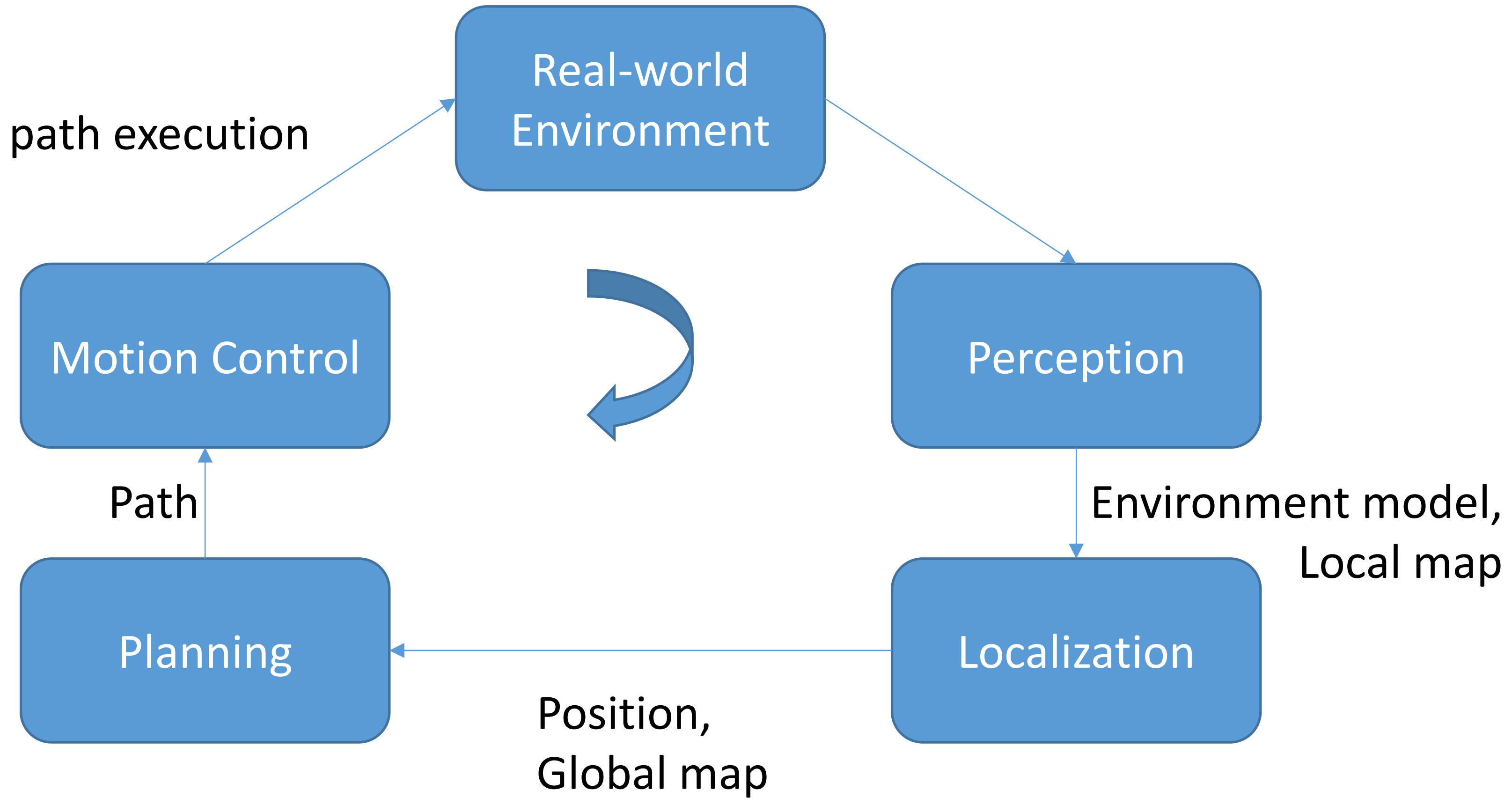
This can be viewed as an overall control structure, in which state, actions, sensors, perception, planning, and learning play specific roles.

Real-world
Environment


Real-world Environment
Perception
Environment model,
Local map


Real-world Environment
Perception
Environment model,
Local map
Localization
Position,
Global map


Real-world Environment
Perception
Environment model,
Local map
Localization
Position,
Global map
Planning
Path


Real-world Environment
path execution
Motion Control
Perception
Path
Environment model,
Local map
Planning
Localization
Position,
Global map

# Sense, Think, Act

Suppose you are given a task: *Rearrange the chairs in the room into a circle.* How would you proceed?

1. Look around the room and evaluate the situation.

   Where are the chairs? How many chairs are there?

2. Make a plan:
   1. Go the first chair, pick it up, place it in the desired position
   2. Repeat for all N chairs.

3. Execute the plan.

This is the basic strategy followed by almost all robots.

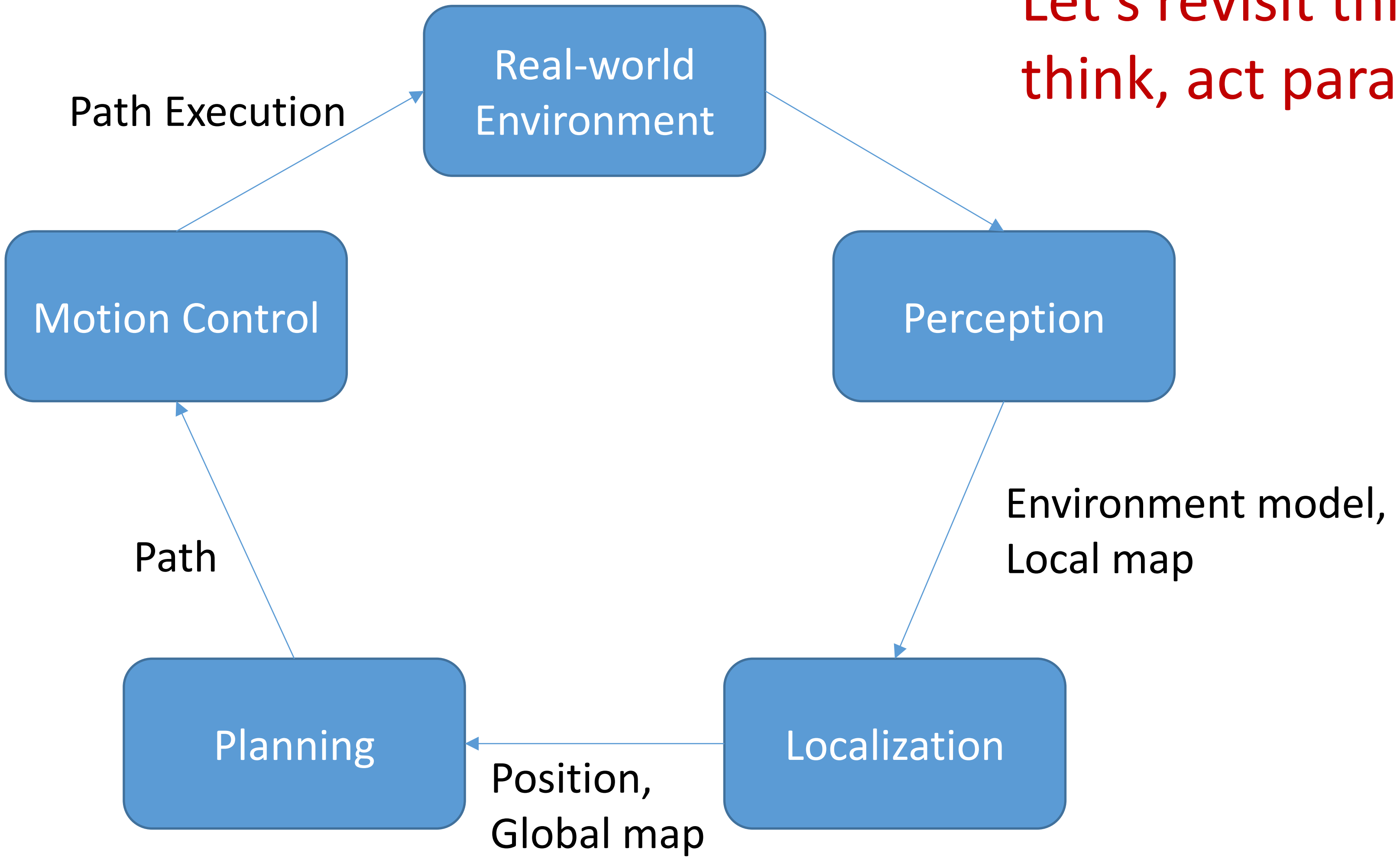
# Sense, Think, Act

Suppose you are given a task: *Rearrange the chairs in the room into a circle.* How would you proceed?

| | |
|---|---|
| 1. Look around the room and evaluate the situation.<br>Where are the chairs? How many chairs are there? | Sense |
| 2. Make a plan:<br>1. Go the first chair, pick it up, place it in the desired position<br>2. Repeat for all N chairs. | Think |
| 3. Execute the plan. | Act |

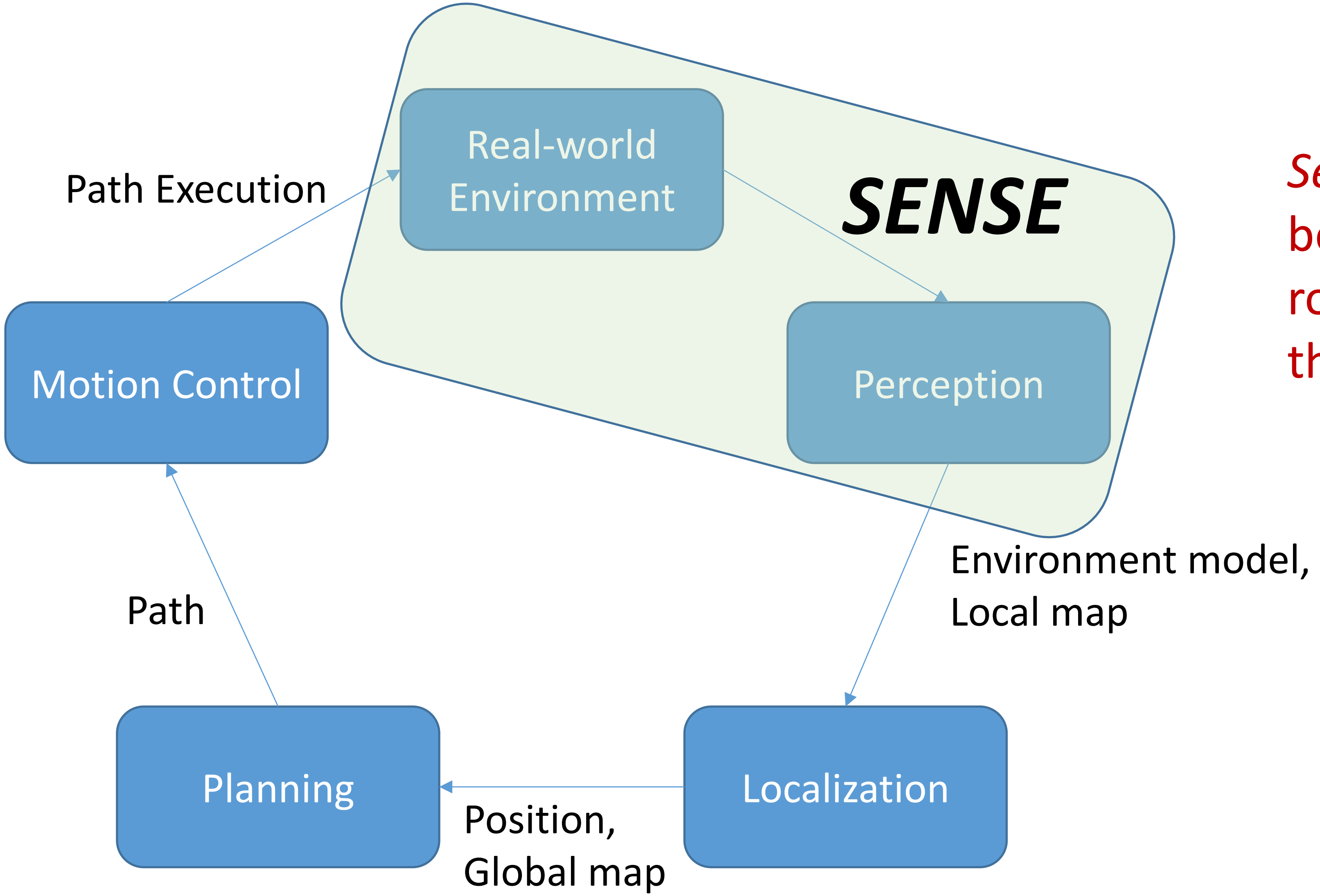
This is the basic strategy followed by almost all robots.

# Example: Navigation in a Known Environment



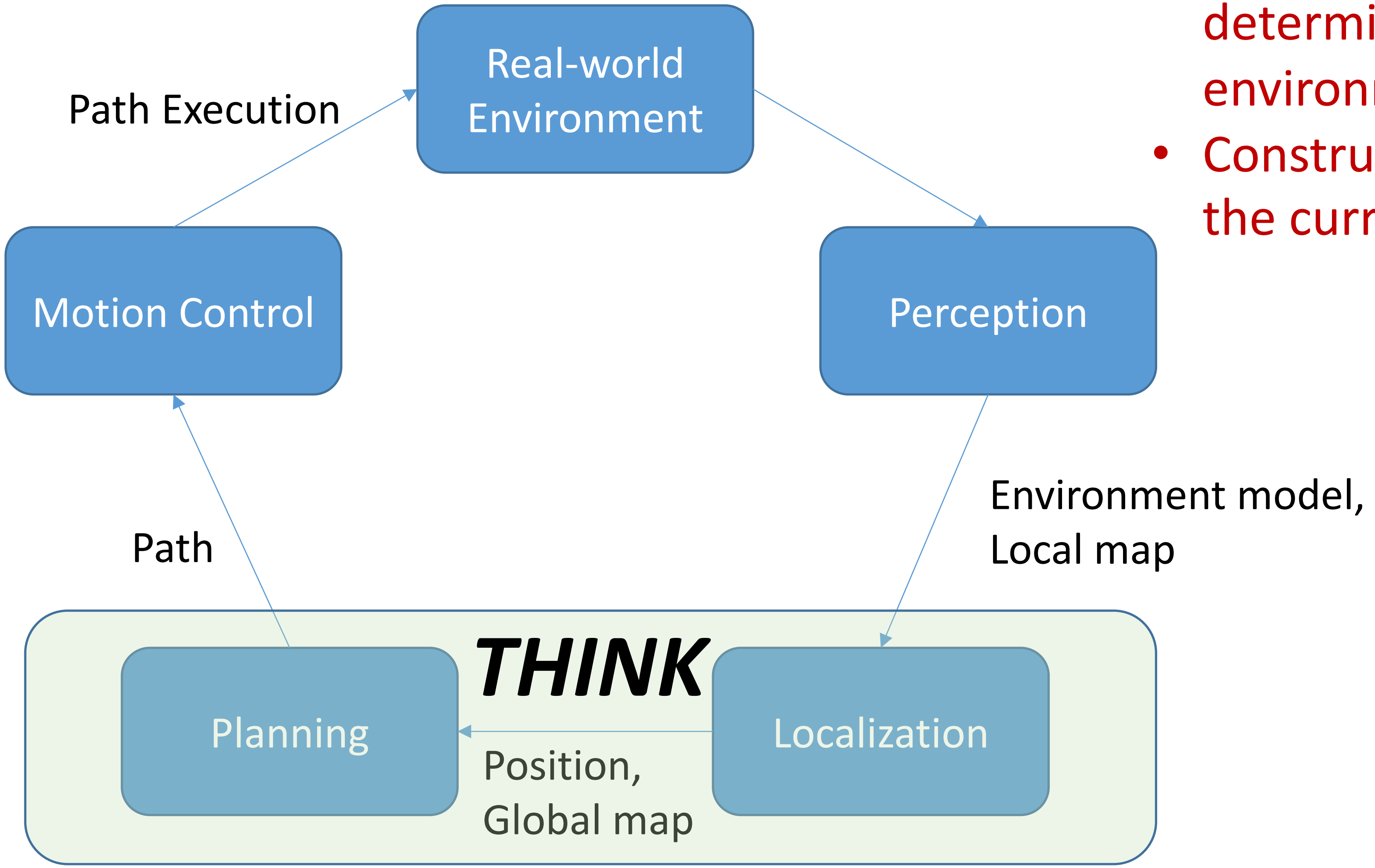

Let's revisit this in terms of the sense, think, act paradigm.

# Example: Navigation in a Known Environment




*Sensing* provides a connection between the real world and the robot's internal representation of the world.

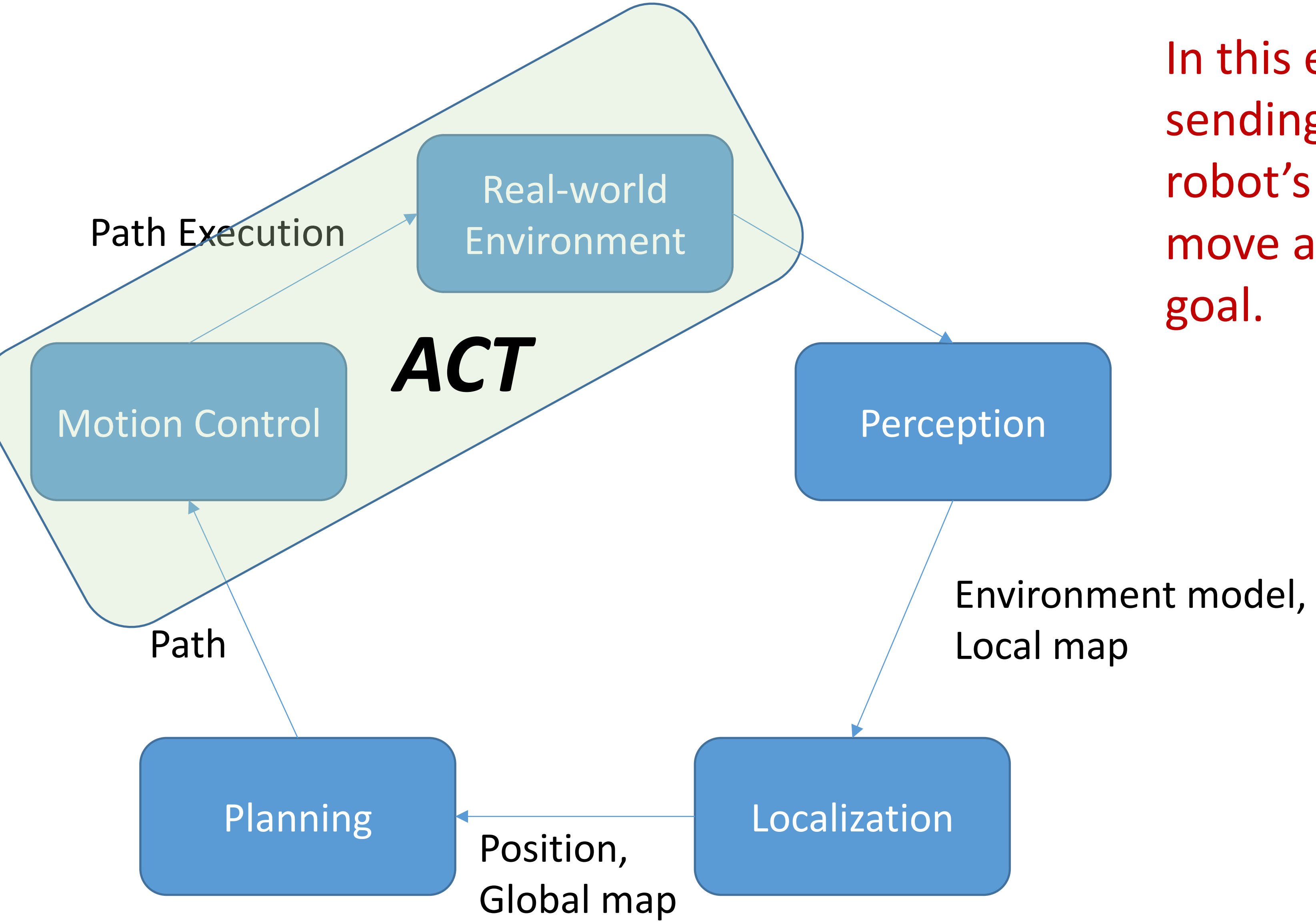
# Example: Navigation in a Known Environment




In this example, *thinking* involves:

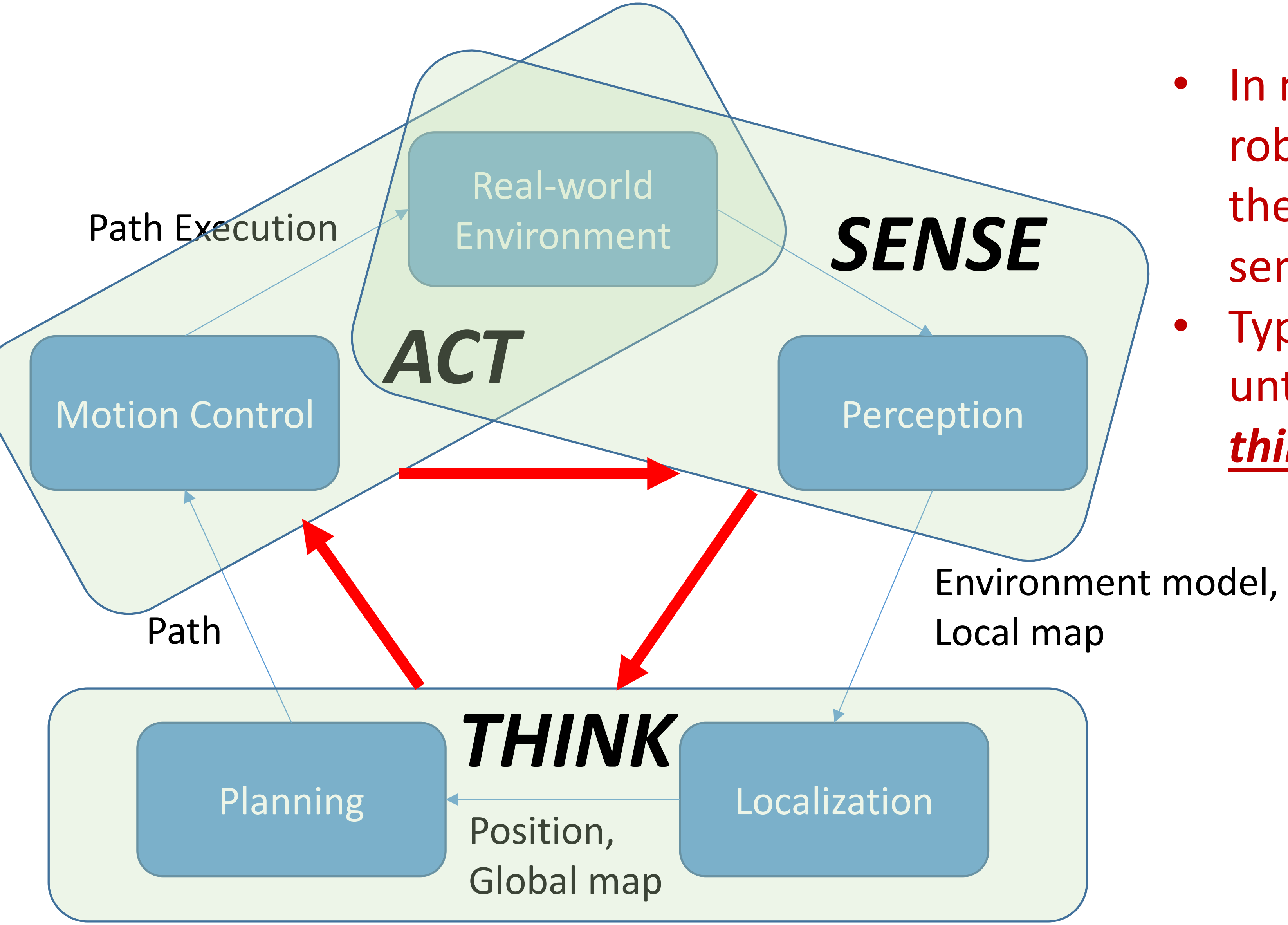
- Processing perceptual information to determine the position of the robot in its environment
- Constructing a motion plan to move from the current position to the goal position.

# Example: Navigation in a Known Environment




In this example, *acting* involves sending motion commands to the robot's motors, so that the robot will move along the desired path to its goal.

# Example: Navigation in a Known Environment

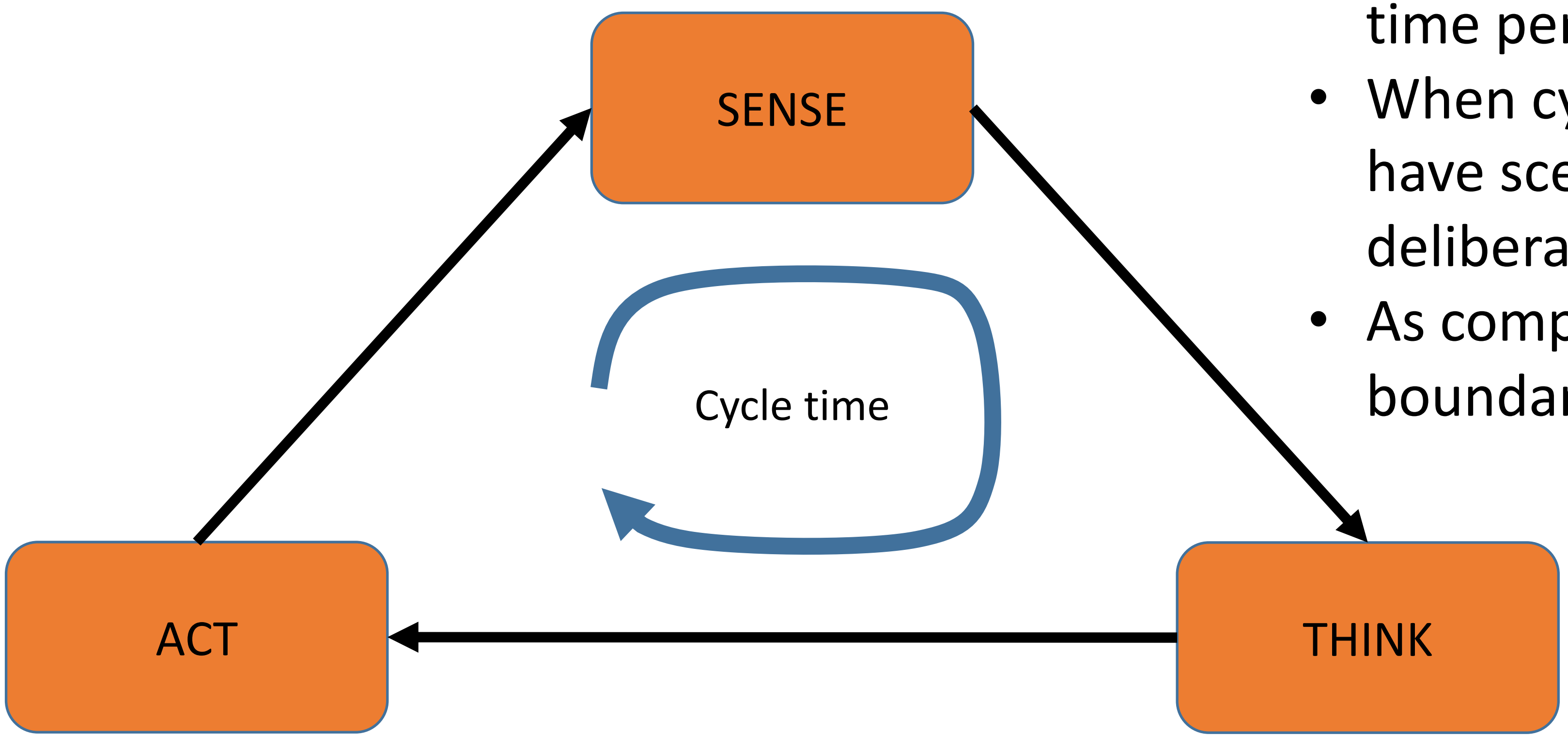



- In most robotics applications, the robot does not succeed to perform the task using a single episode of sense, think, act.
- Typically, these stages are repeated until the task is achieved: the ***sense, think, act loop.***

# Sense, Think, Act at Different Time Scales

The time to complete one cycle of this loop depends on the task:

- Playing chess: minutes
- Hand-eye coordination: 30 Hz
- Force controlled robot: Order of KHz




- When cycle time is very fast, we use tools from control theory, and model systems using differential equations (continuous time performance).
- When cycle time is very slow, we might have scene understanding and deliberative planning.
- As computers become faster, the boundary between these begins to blur.

# *State*

# Representing the Robot and the World

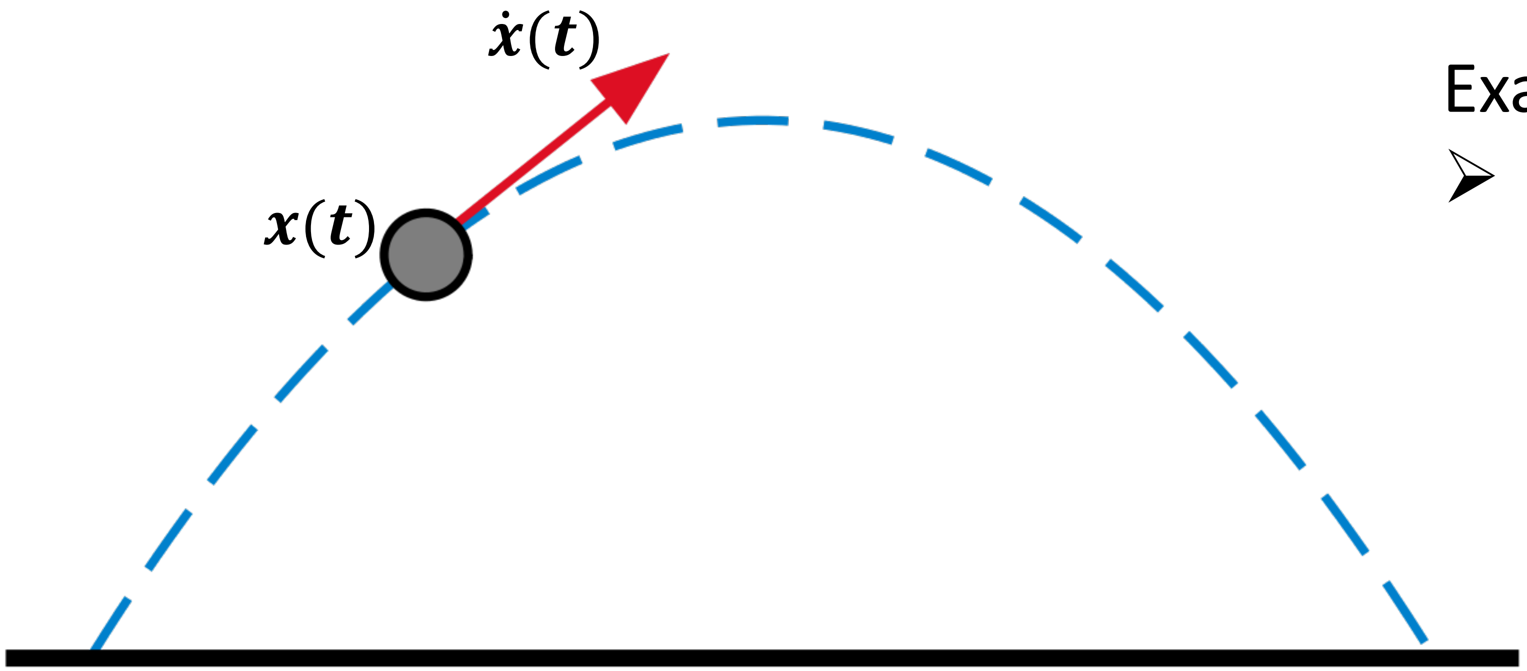
- Perception has the responsibility of converting sensor measurements into a representation of the world and of the robot’s current situation.
- Planning uses these representations to reason about the effects of actions in the world.

These representations define the robot’s ***state***, and the world ***state***.

# State

The term ***state*** is used in the study of dynamical systems to describe the relevant aspects of an objects motion.

If we know the state $x$ at time $t_0$ along with the system input for all $t \geq t_0$, then we can predict the state at all future times.




Example:

- If we know the position and velocity of a projectile at a given time, we can compute its entire trajectory.

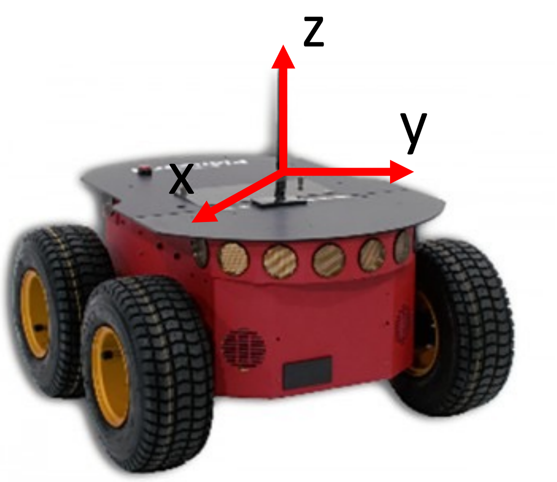
# Geometric Representations

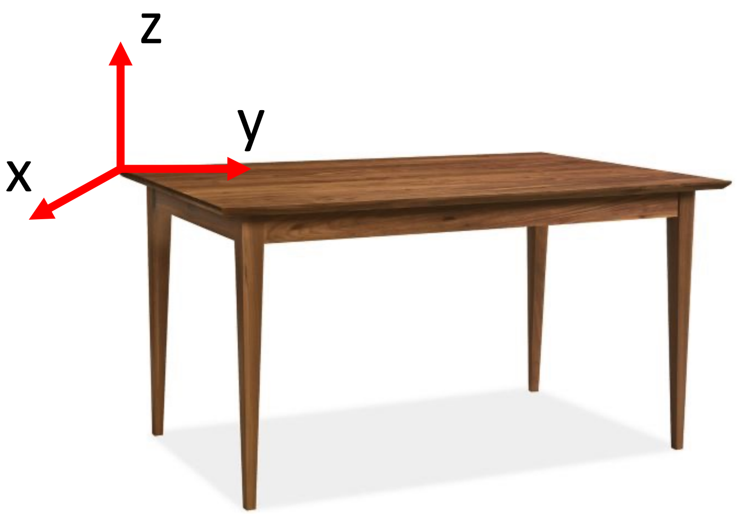
In robotics, we often require specific geometric information.

To describe an object's position:

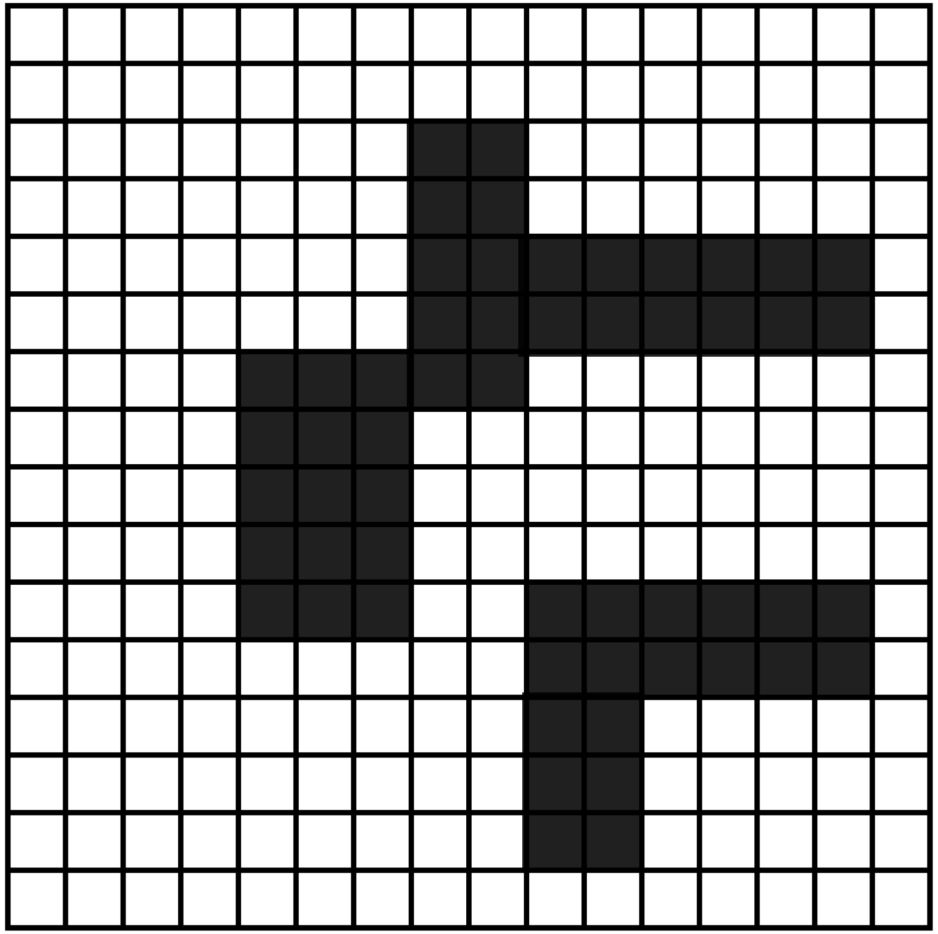
- Attach a coordinate frame to the object (rigid attachment of frame to the object)
- Specify the position and orientation of the coordinate frame.

If we know this information, we know everything about the object's position!




# Grid World



- For many mobile robotics applications, one can represent the world as a grid.
- The robot state is defined by its current grid cell location.
- Each grid cell is either free or occupied by an obstacle (world state).
- There are many variations, e.g., assign to each cell in the grid a *probability* that it is occupied by an obstacle.

# Symbolic Representations

For high-level task planning, it is often sufficient to represent the world using symbolic descriptions.

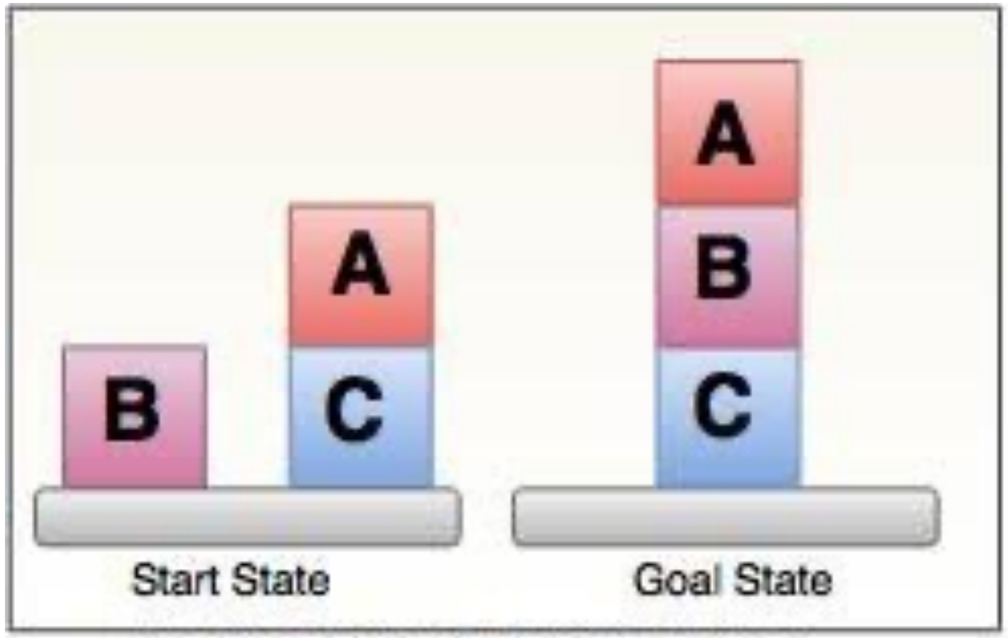


Fig: Blocks-World Planning Problem

**Representation of Blocks World using simple predicates**

***Initial State:***

- On(table,B)
- On(table,C)
- On(A,C)
- Clear(B)
- Clear(A)

***Goal State:***

- On(table,C)
- On(A,B)
- On(B,C)
- Clear(A)

# *Actions and Planning*

# High-Level Planning

A high-level planner uses a symbolic representation of actions:

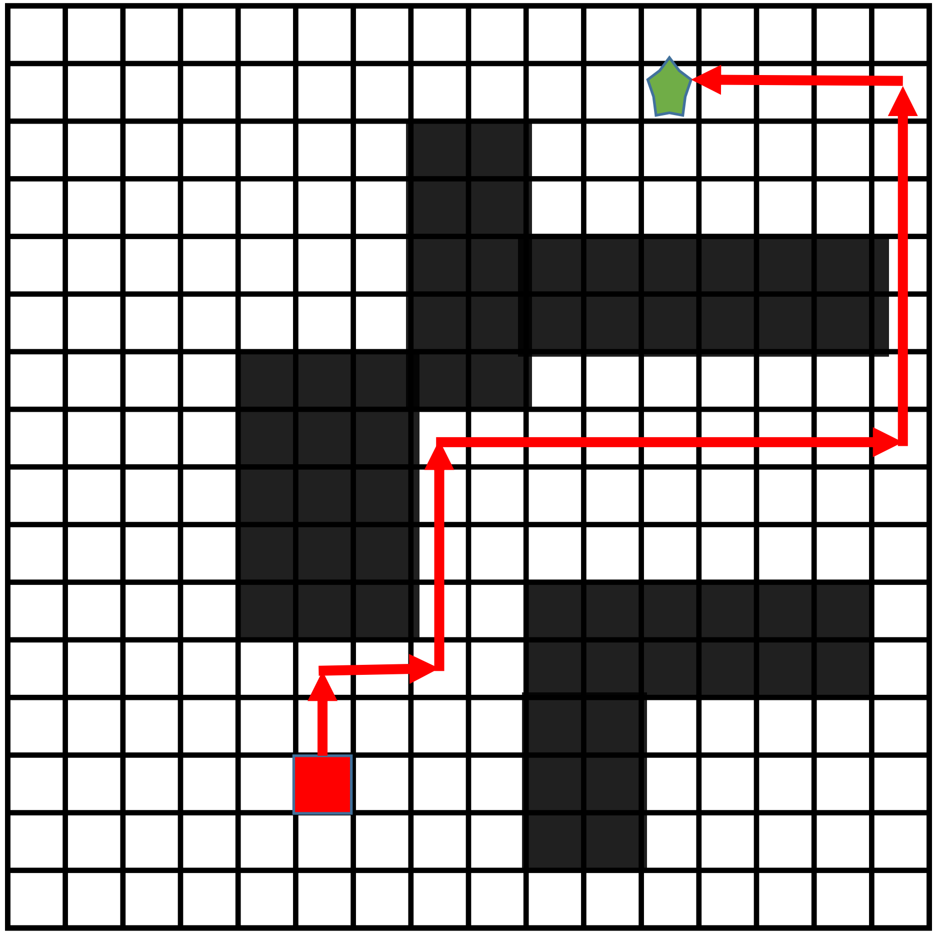
- Preconditions: what must be true in the world before the action is applied?
- Effects: what changes occur in the world after the action occurs?

**Pickup**(?X):
  **Preconditions**: Gripper(empty)
  **Effects**: Gripper(full), Holding(?X)

If the goal is to be holding Block B, the planner can instantiate the variable ?X to B

**Pickup**(B):
  **Preconditions**: Gripper(empty)
  **Effects**: Gripper(full), Holding(B)

# Grid World: Path Planning

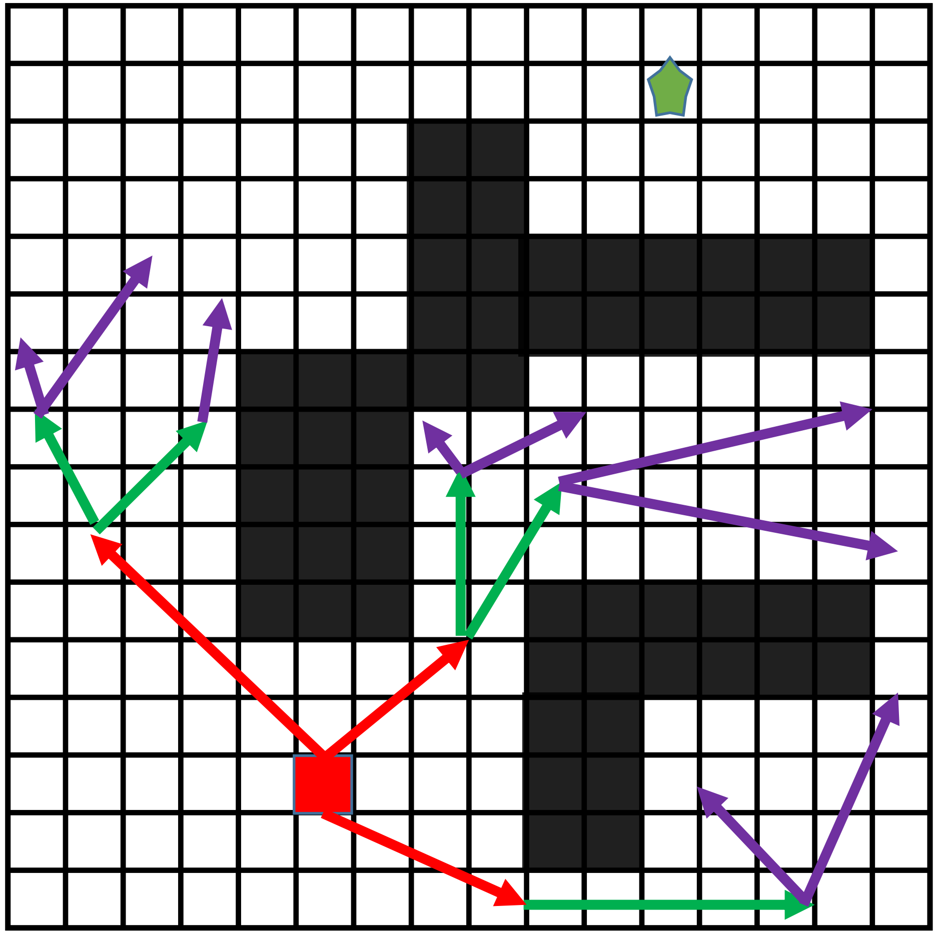


Actions: move to an adjacent cell

The path planning problem is to find a free path from start to goal.

- How can we effectively find any path from start to goal?
- How should we decide which path to take?

# Grid World



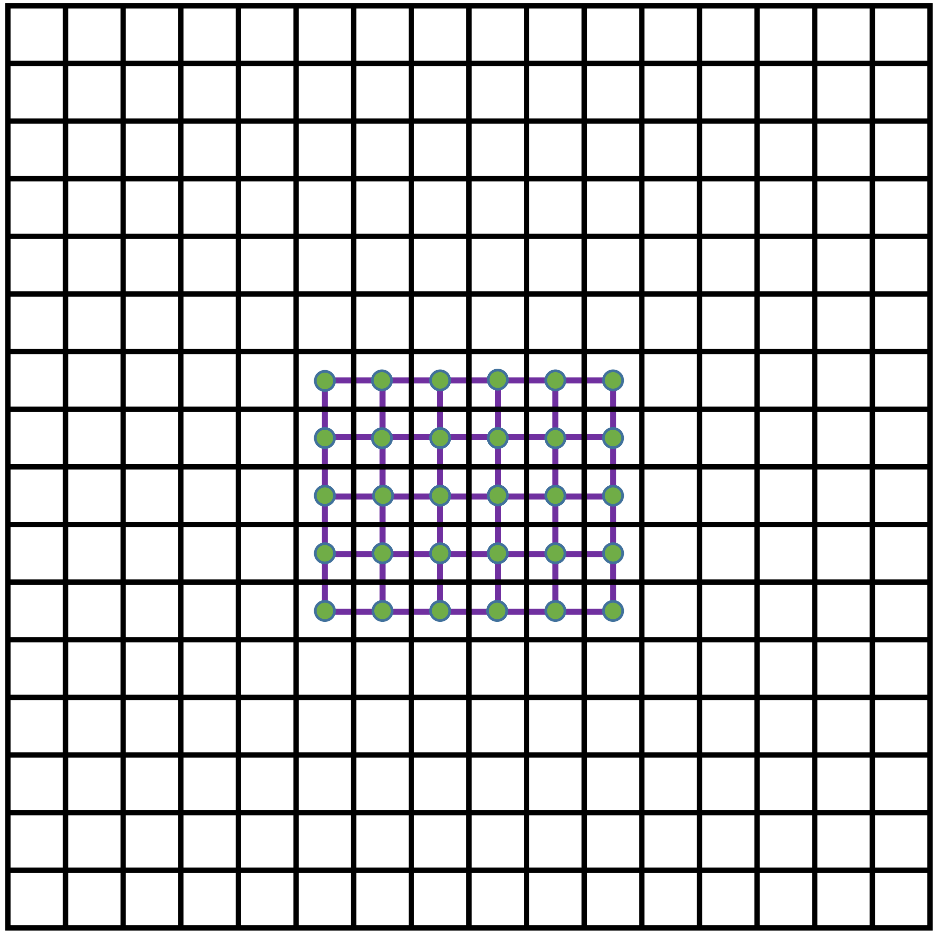
**Start position**

**Goal position**

One strategy is to systematically explore various possible solution paths.

This raises the question:
What strategies should we use to explore alternative paths?

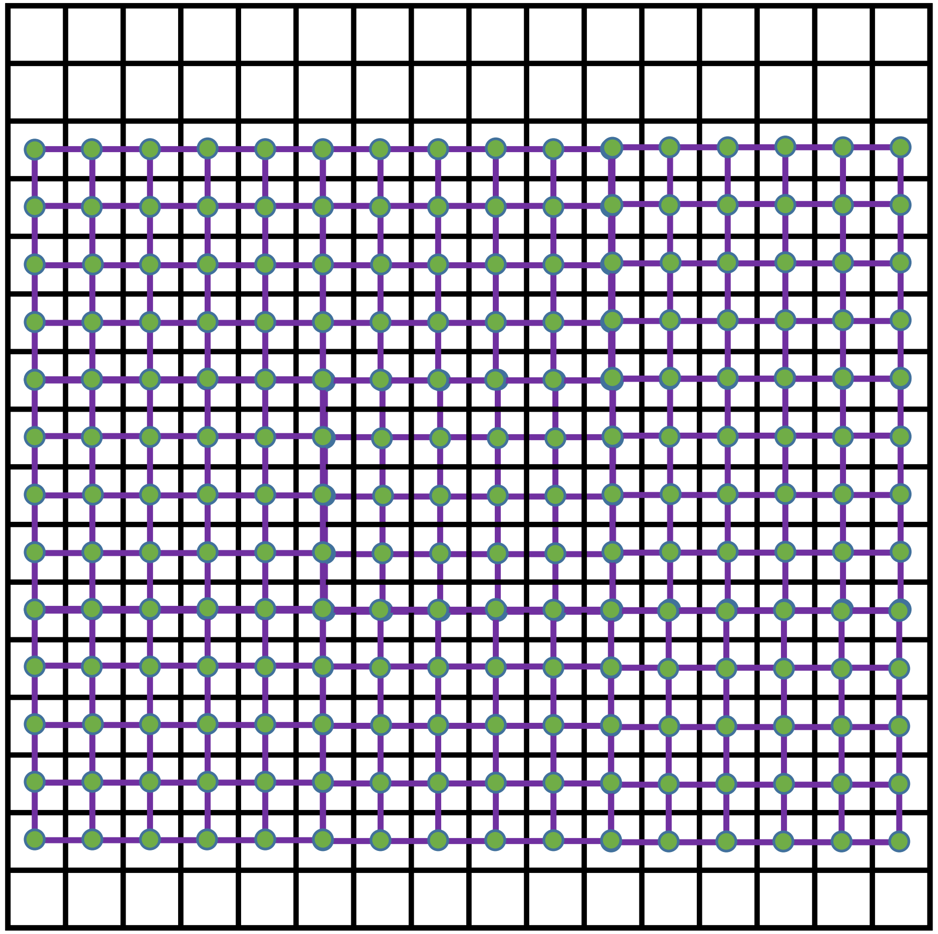
# Grid World



A grid can be represented as a graph:

- Each cell in the grid corresponds to a vertex in the graph
- Vertices that correspond to adjacent grid cells are connected by an edge.

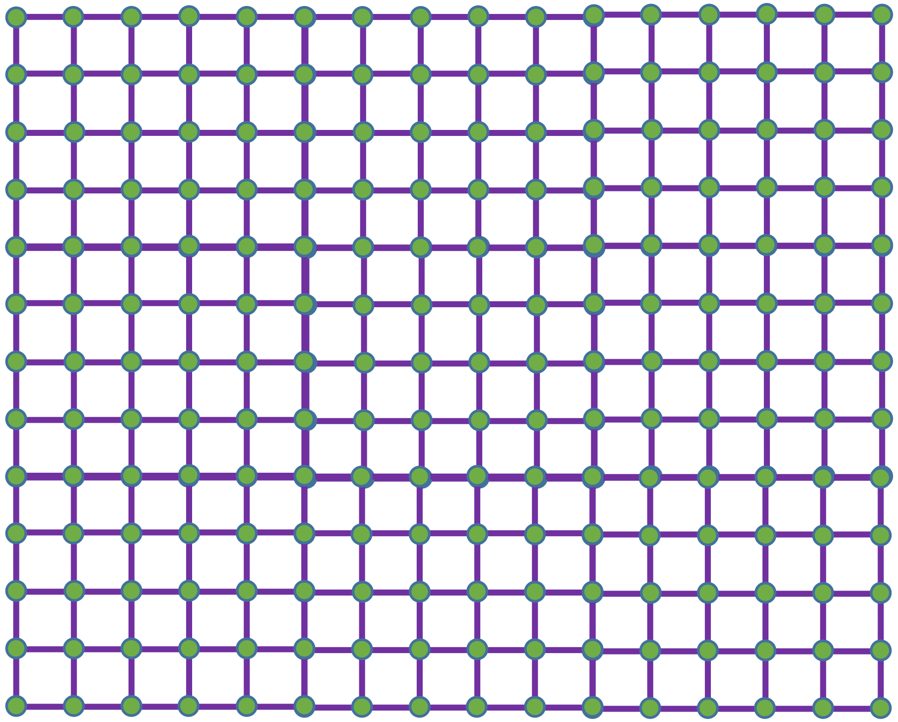
# Grid World



A grid can be represented as a graph:

- Each cell in the grid corresponds to a vertex in the graph
- Vertices that correspond to adjacent grid cells are connected by an edge.

# Grid World

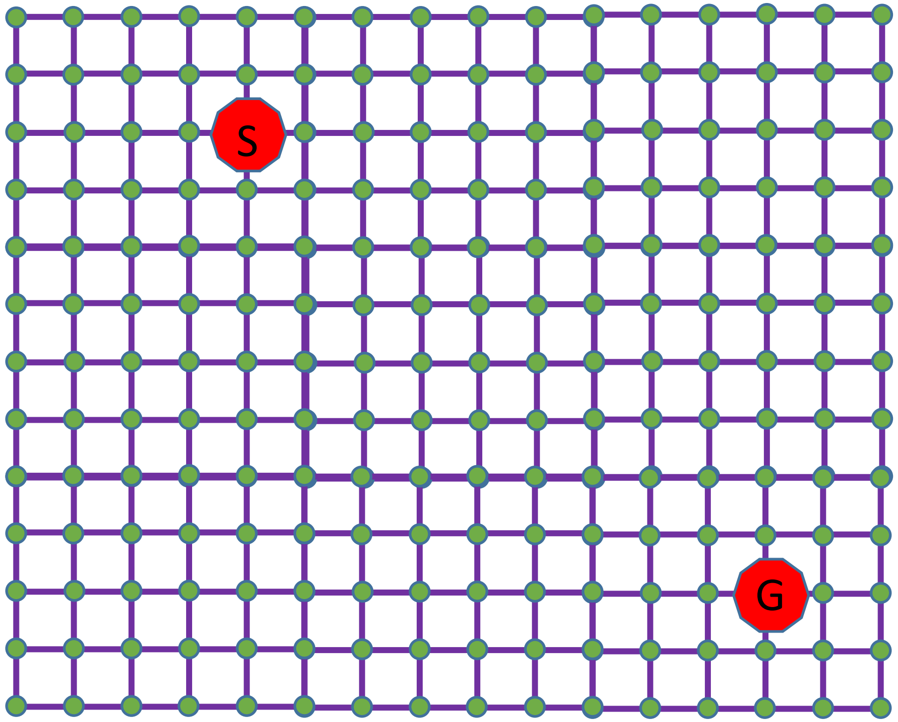


A grid can be represented as a graph:

- Each cell in the grid corresponds to a vertex in the graph
- Vertices that correspond to adjacent grid cells are connected by an edge.

And now, we can use graph search algorithms to find a path!

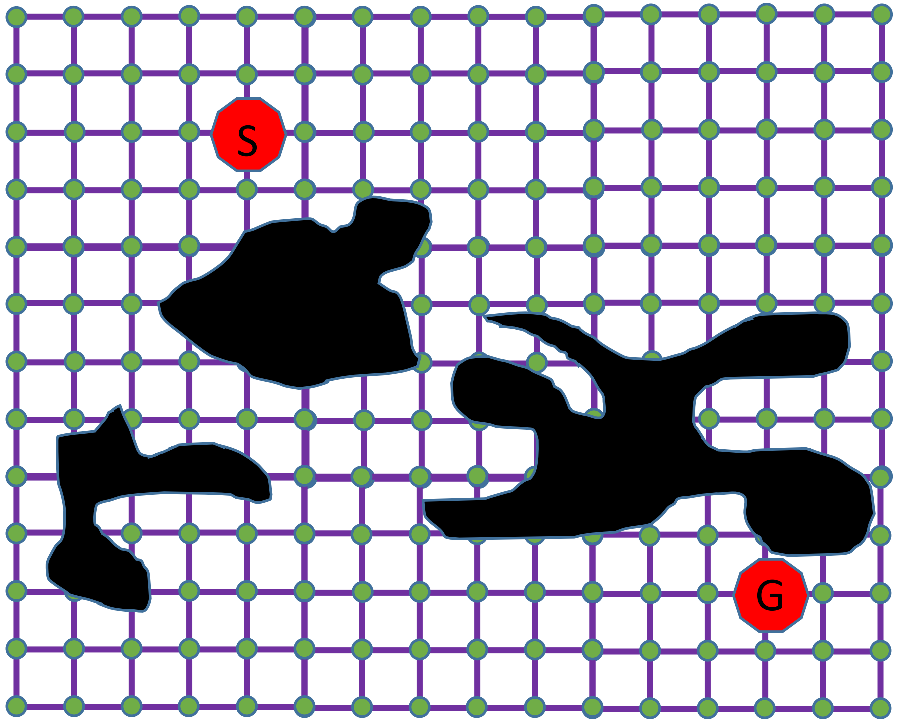
# Grid World




Define a Starting state and a Goal state, and use your favorite graph search algorithm to find a path.

When there are no obstacles, it's easy.

# Grid World

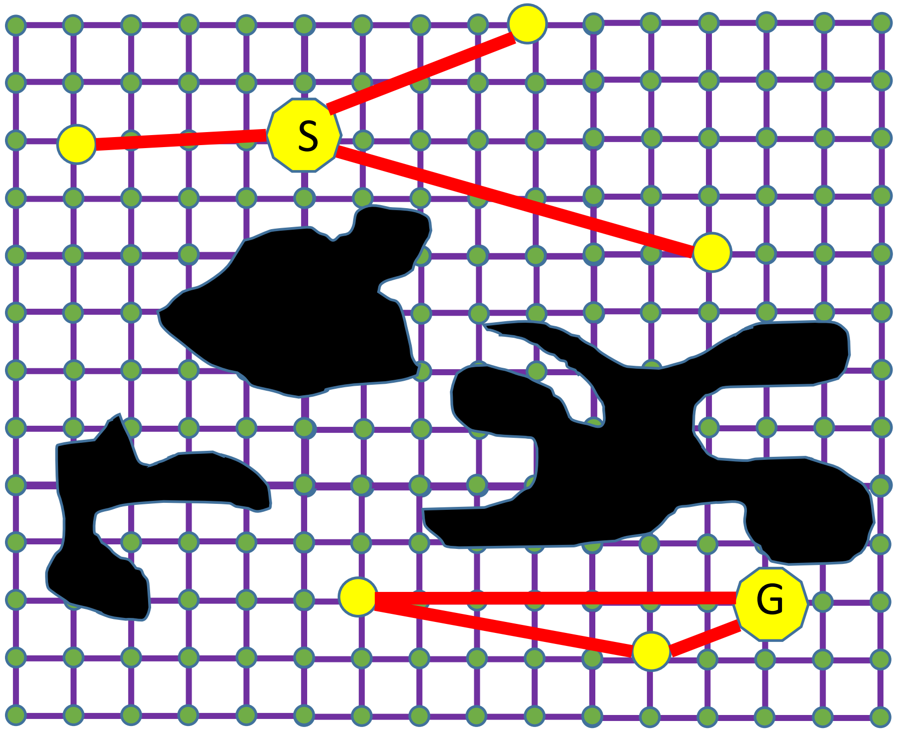



Define a Starting state and a Goal state, and use your favorite graph search algorithm to find a path.

When there are no obstacles, it's easy.

When there are obstacles, it becomes (only) slightly more difficult.

# Sampling-based algorithms




- Don’t build the entire grid a priori.
- Build the grid incrementally by generating random grid samples.
- Connect near-by samples when a collision-free path exists.
- No need for paths to stay on the grid.
- Stop sampling when we can find a path that satisfies the problem

# Planning under uncertainty

- For many robotics applications, the world state is not known with certainty.
- In such cases, we use **probability theory** to characterize uncertainty.
- Planning typically involves maximizing some reward, or minimizing some cost, on average, over many trials.
- If we don’t know the relevant probabilities, we can often apply machine learning techniques to develop good estimates for these.
- Planning can look like: optimization, theorem proving (logic), geometry, stochastic control, optimal control, etc. – depending on the problem to be solved, the representation of state, and the nature of uncertainties.

# *Sensing and Perception*

# Some Sensors


Pan-Tilt Camera

Intel's RealSense
Depth Camera

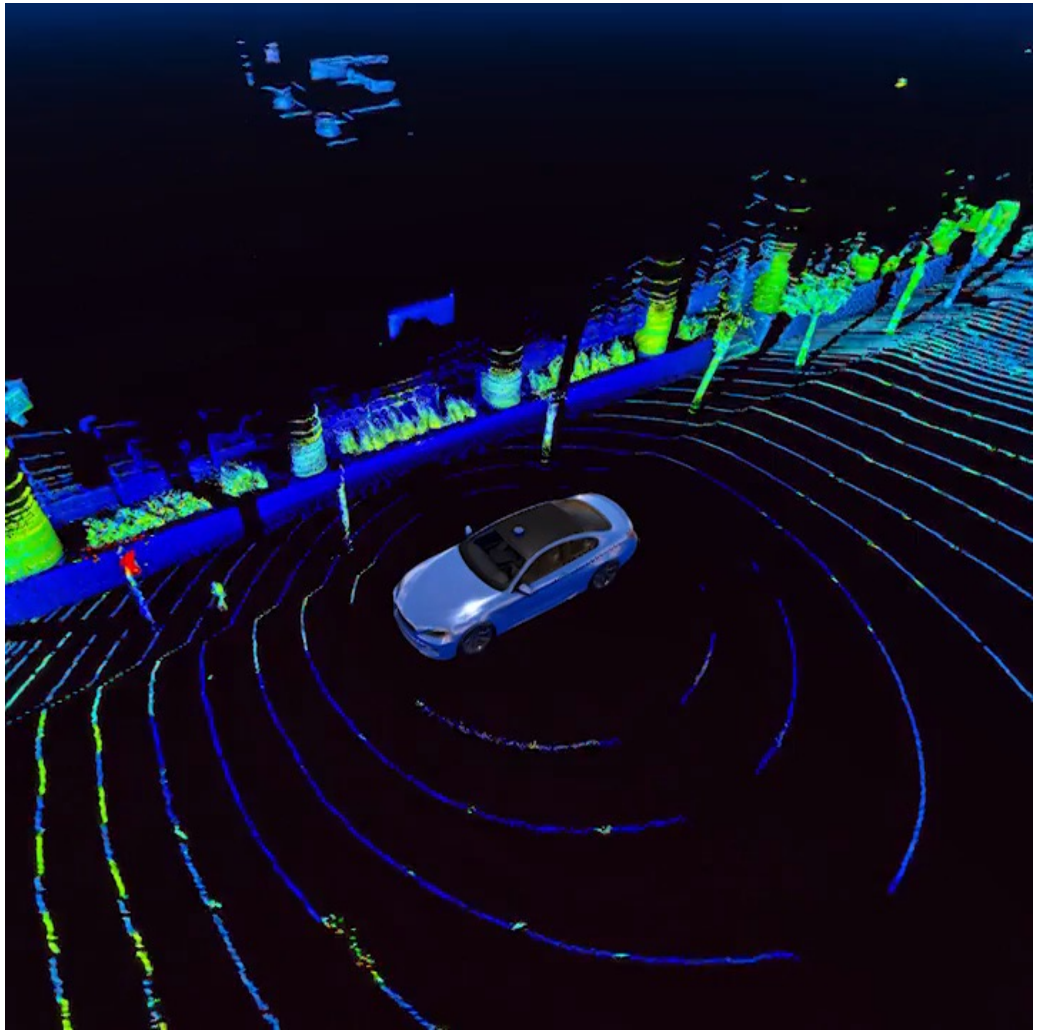

Velodyne LIDAR

# LIDAR

LIght Detection And Ranging
aka Laser Scanning
aka 3D scanning

1. Emit light wave pulse
2. Measure time to return
3. Compute distance

Do this a few million times per second, and voila!



# Perception

- Sensor readings are subject to noise and other errors.
- Sensor readings alone are not sufficient to reconstruct the state of the world:
  - A depth sensor reads 10m… what does that imply about the world?
  - Along a corridor there are many office doors. How can we know where we are when all doors look the same?
- Perception uses contextual information (e.g., maps, other sensor readings) to reason about state using sensor data as input.
- Bayesian inference is a key tool for this.

# *Learning*

# Machine Learning

- Maybe the hottest topic in robotics, and all of AI, today.
- Many methods have been developed:
  - Simple parameter estimation.
  - Reinforcement Learning (RL)
  - Deep Learning (using convolutional neural nets)
  - Deep RL

We won’t go into great depth with ML, but we’ll look at a few methods for specific cases.