State, probability, and actions

CS 3630

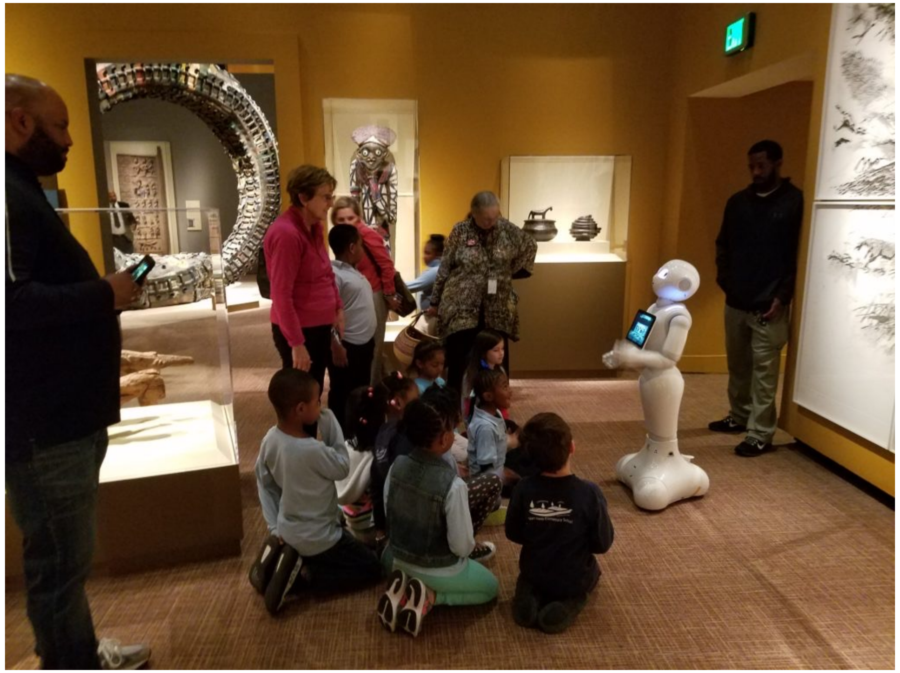
A Taxonomy of Robotics Topics

To develop a robot we must integrate six distinct aspects:

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?

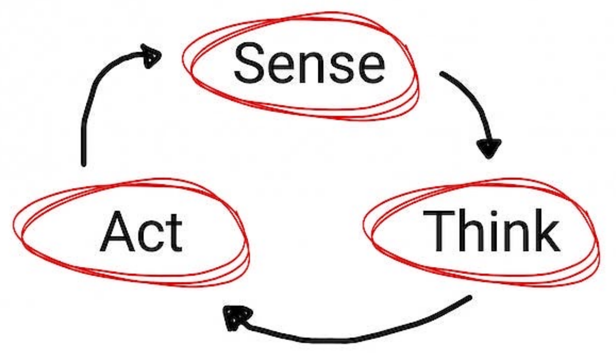
Museum Guide Robot example

- **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 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.

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.

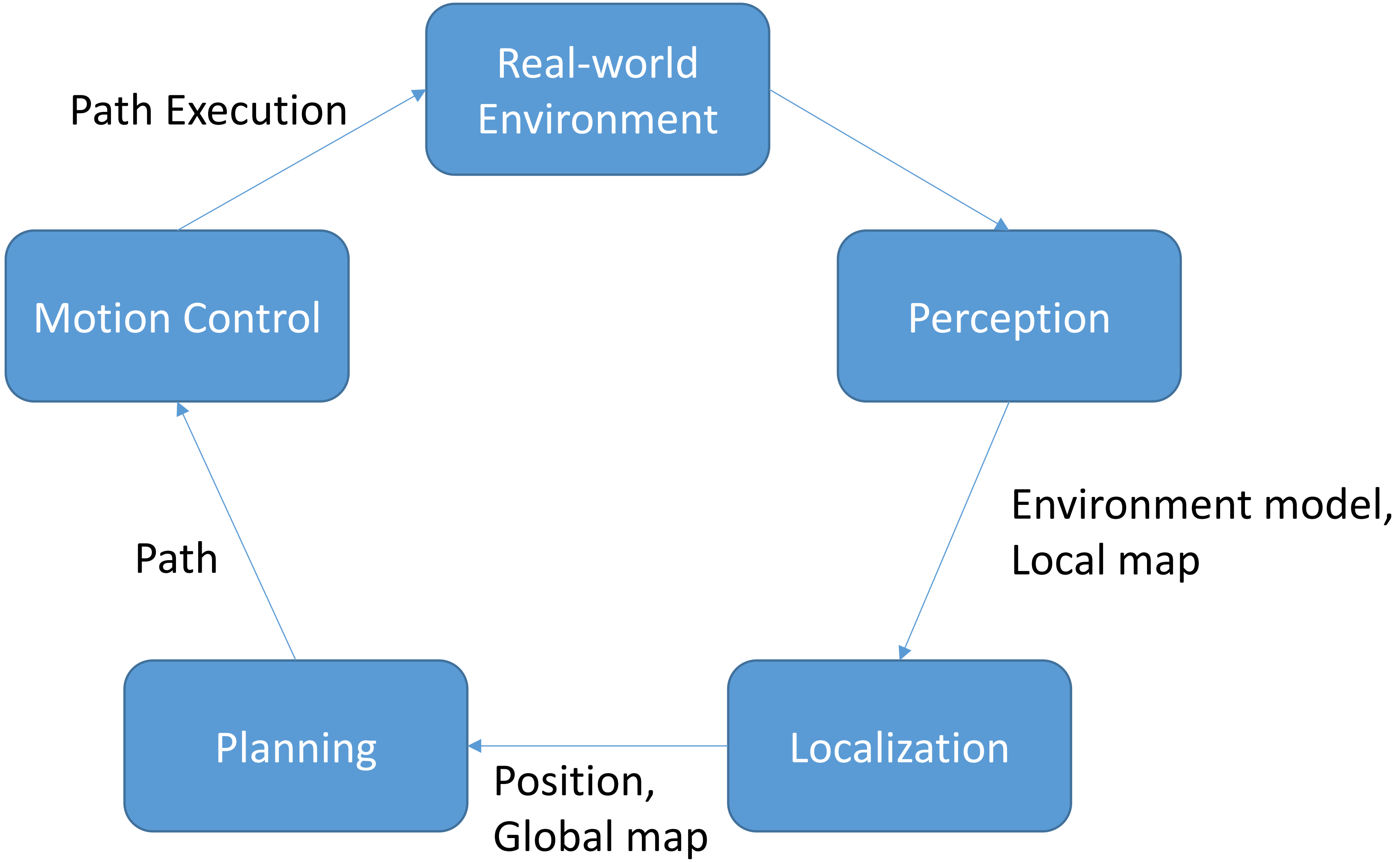
Sense, Think, Act

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

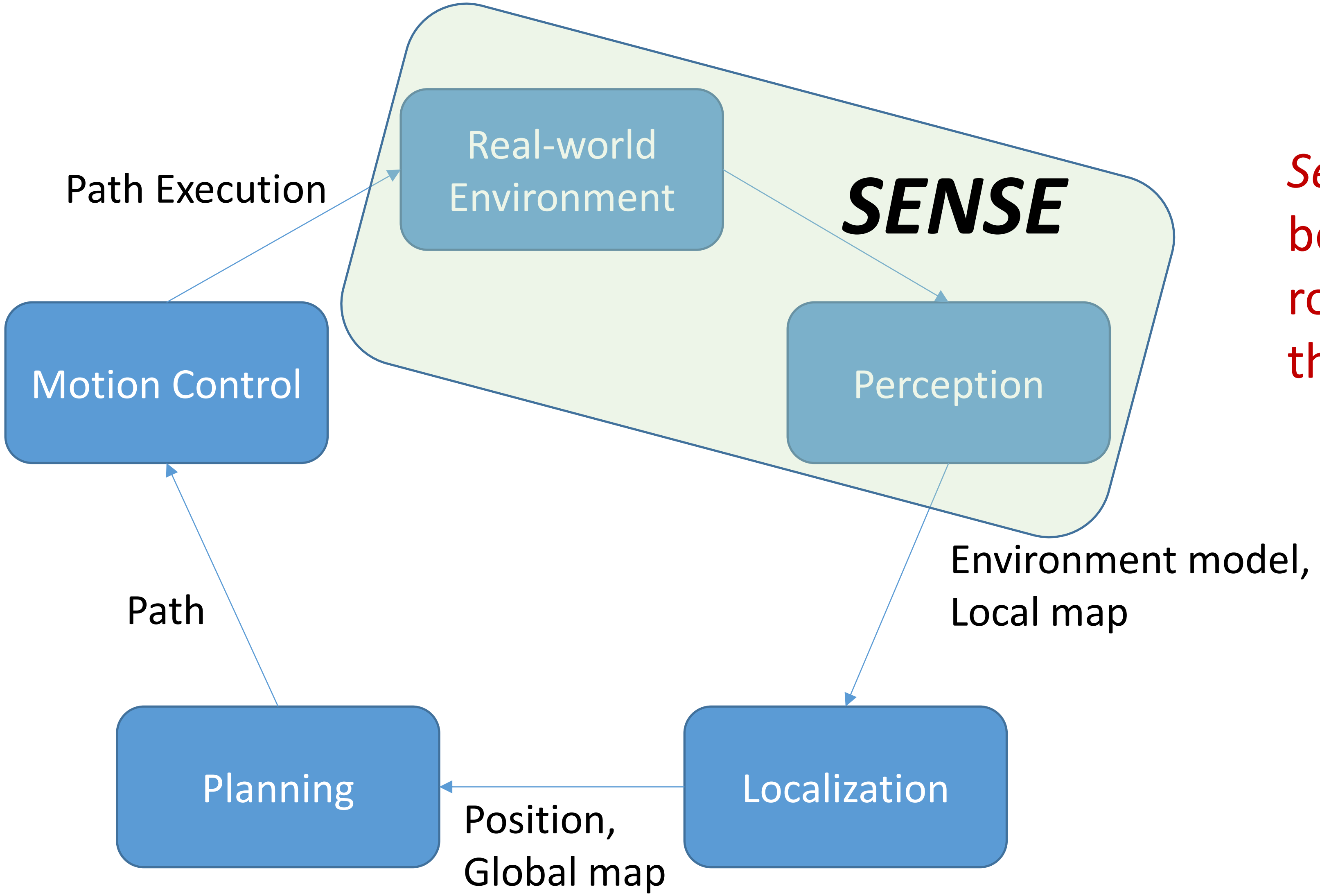
1. Look around the room and evaluate the situation. Where are the chairs? How many chairs are there?	Sense
2. Make a plan: 1. Go the first chair, pick it up, place it in the desired position 2. Repeat for all N chairs.	Think
3. Execute the plan.	Act

This is the basic strategy followed by almost all robots.

Example: Navigation in a Known Environment

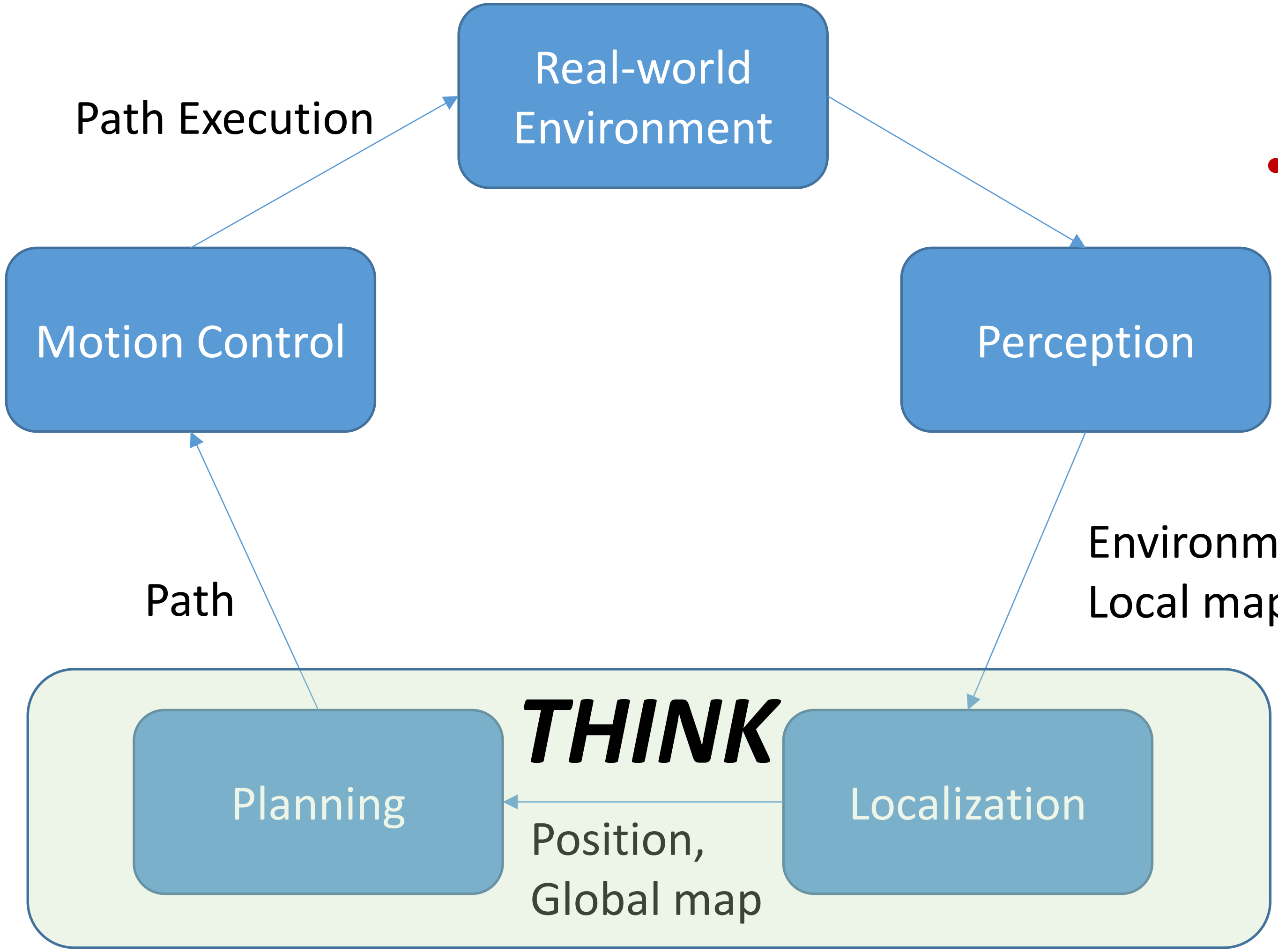


Example: Navigation in a Known Environment



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

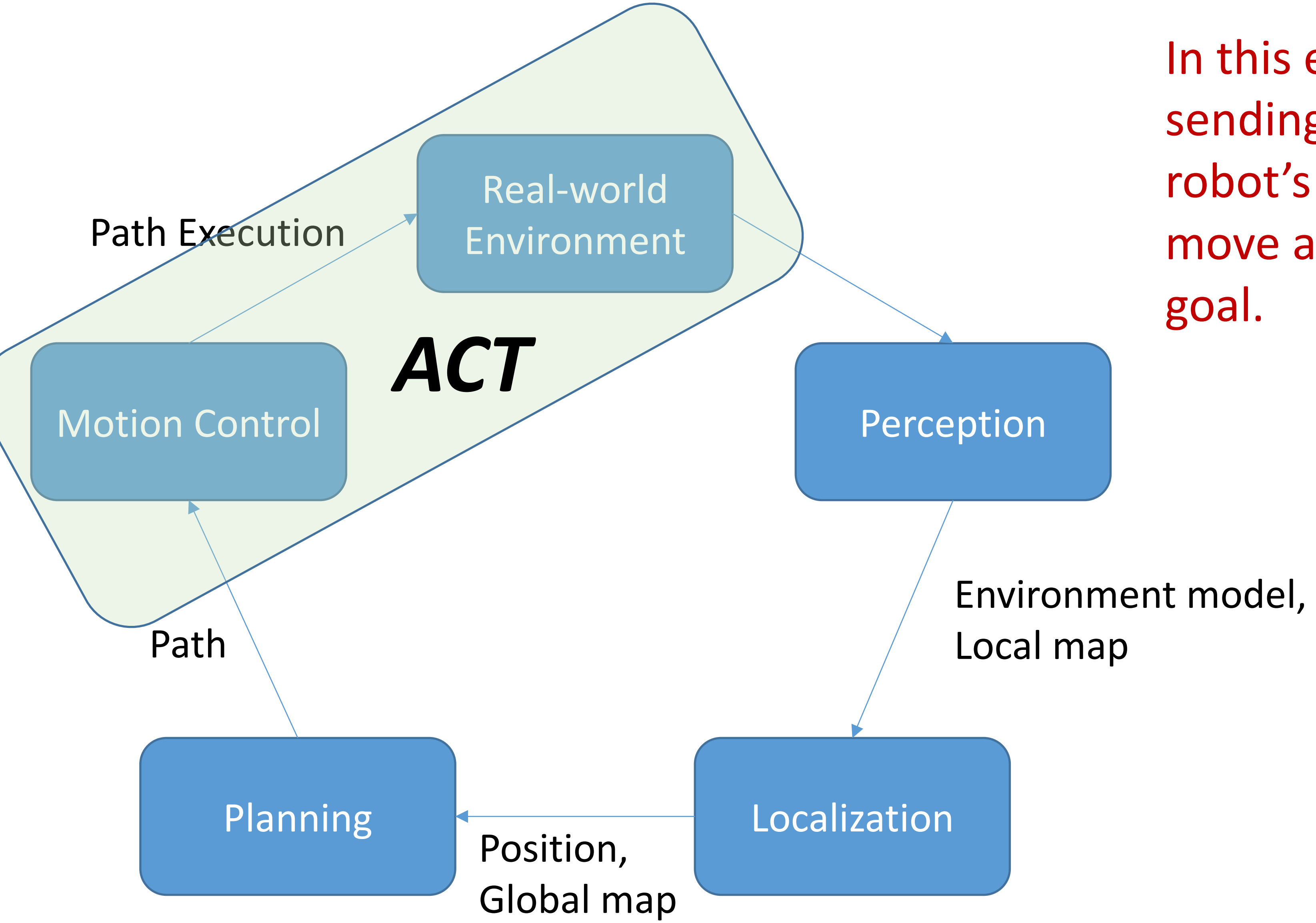
Example: Navigation in a Known Environment



In this example, *thinking* involves:

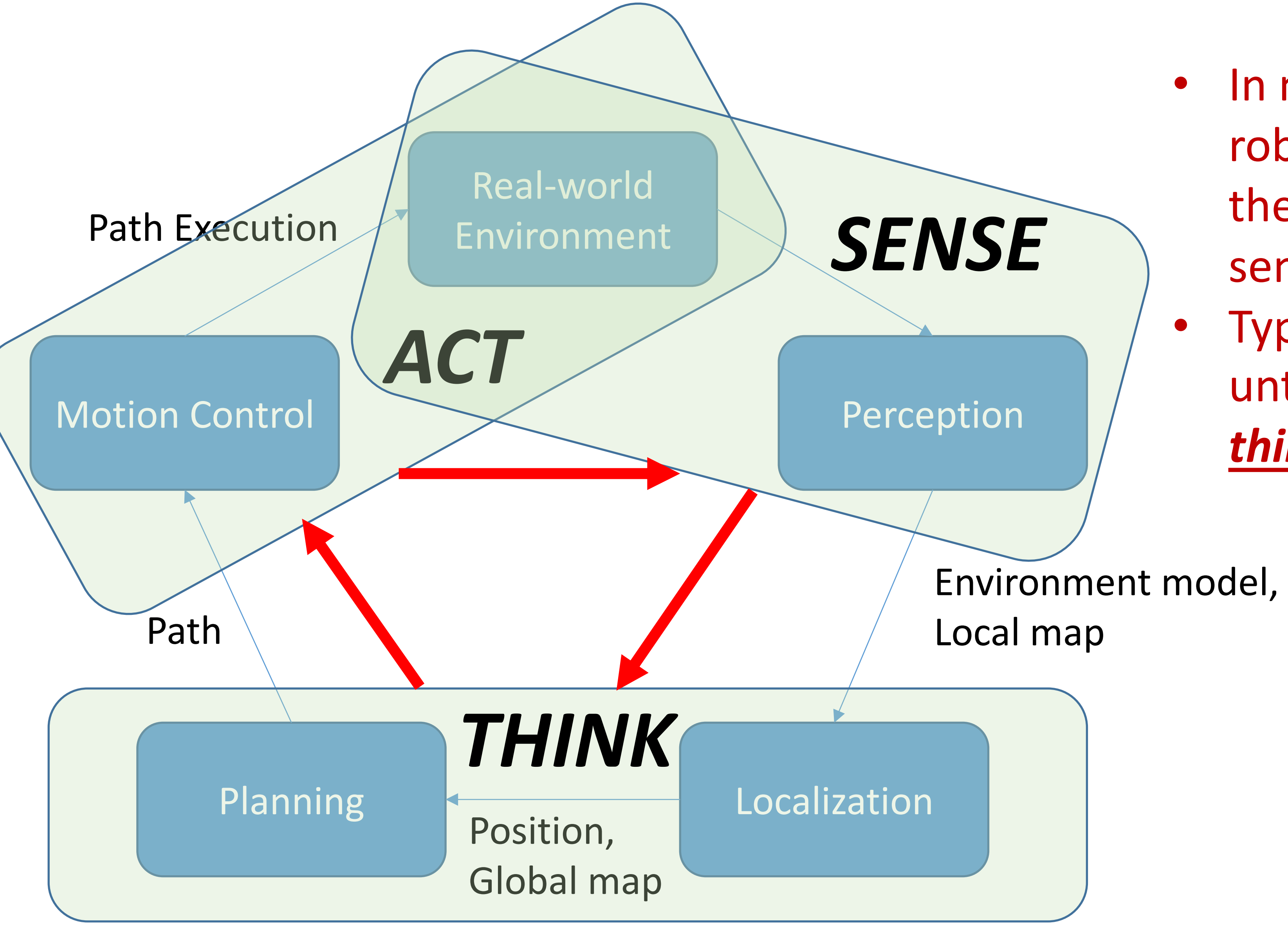
- Processing perceptual information to determine the position of the robot in its environment
- Constructing a motion plan to move from the current position to the goal position.

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

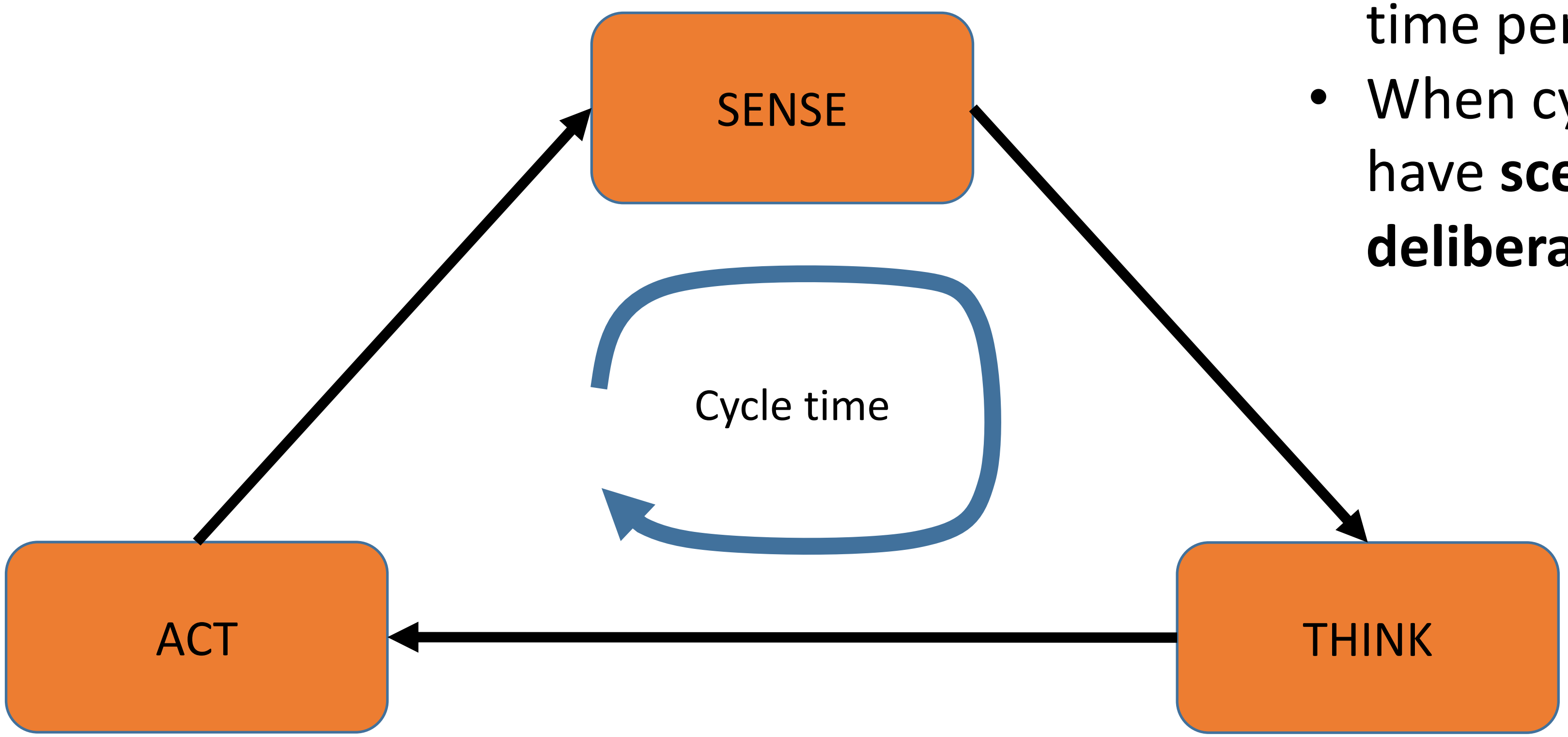


- In most robotics applications, the robot does not succeed to perform the task using a single episode of sense, think, act.
- Typically, these stages are repeated until the task is achieved: the ***sense, think, act loop.***

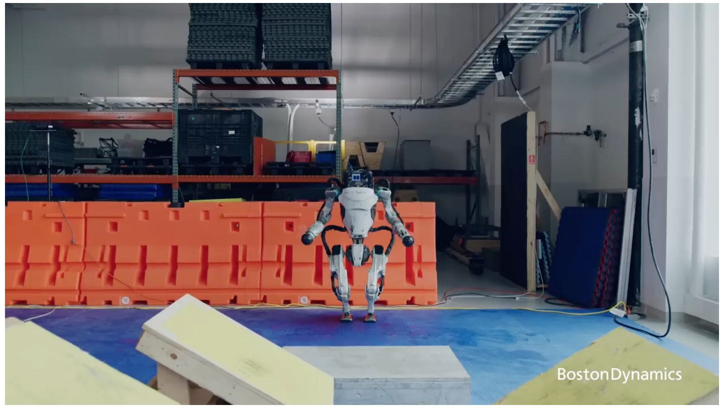
Sense, Think, Act at Different Time Scales

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**.



Boston Dynamics Atlas

Boston Dynamics Spot

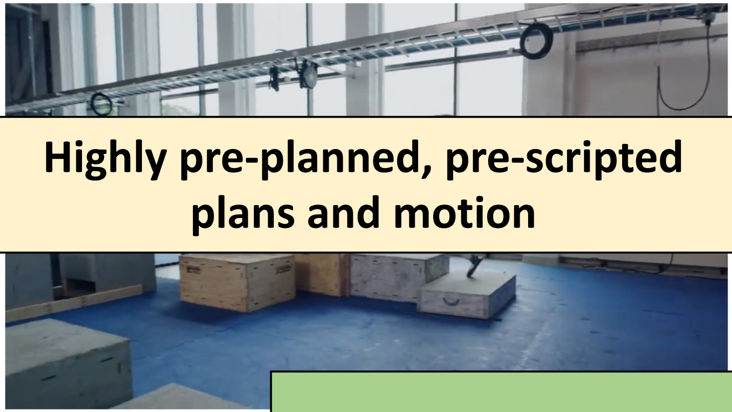
Ocado

Hello Robot Stretch

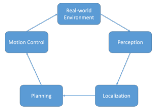
Boston Dynamics Spot
(visual navigation @GT/Google)

Starship

The Sense-Think-Act loop can take many forms!



Highly pre-planned, pre-scripted plans and motion

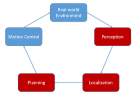


Boston

Full Sense-Think-Act loop

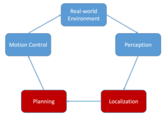
Centralized planning system, individual robots make few decisions

Teleoperated by a human!

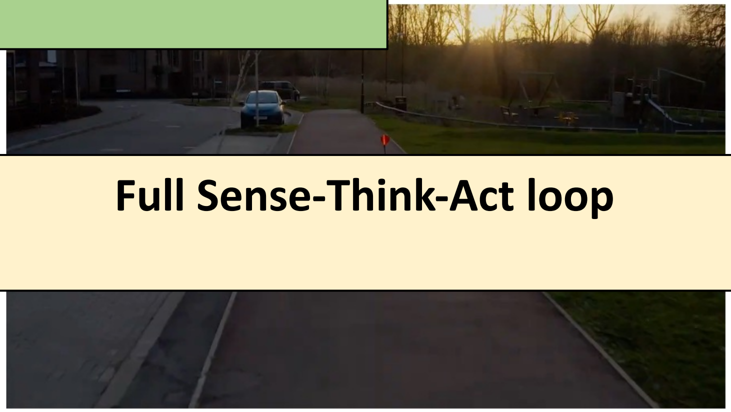


Hello Robot Stretch

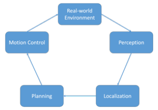
No explicit localization and planning, vision→ motion



Boston Dynamics Spot
(visual navigation @GT/Google)



Full Sense-Think-Act loop



Starship

State

Representing the Robot and the World

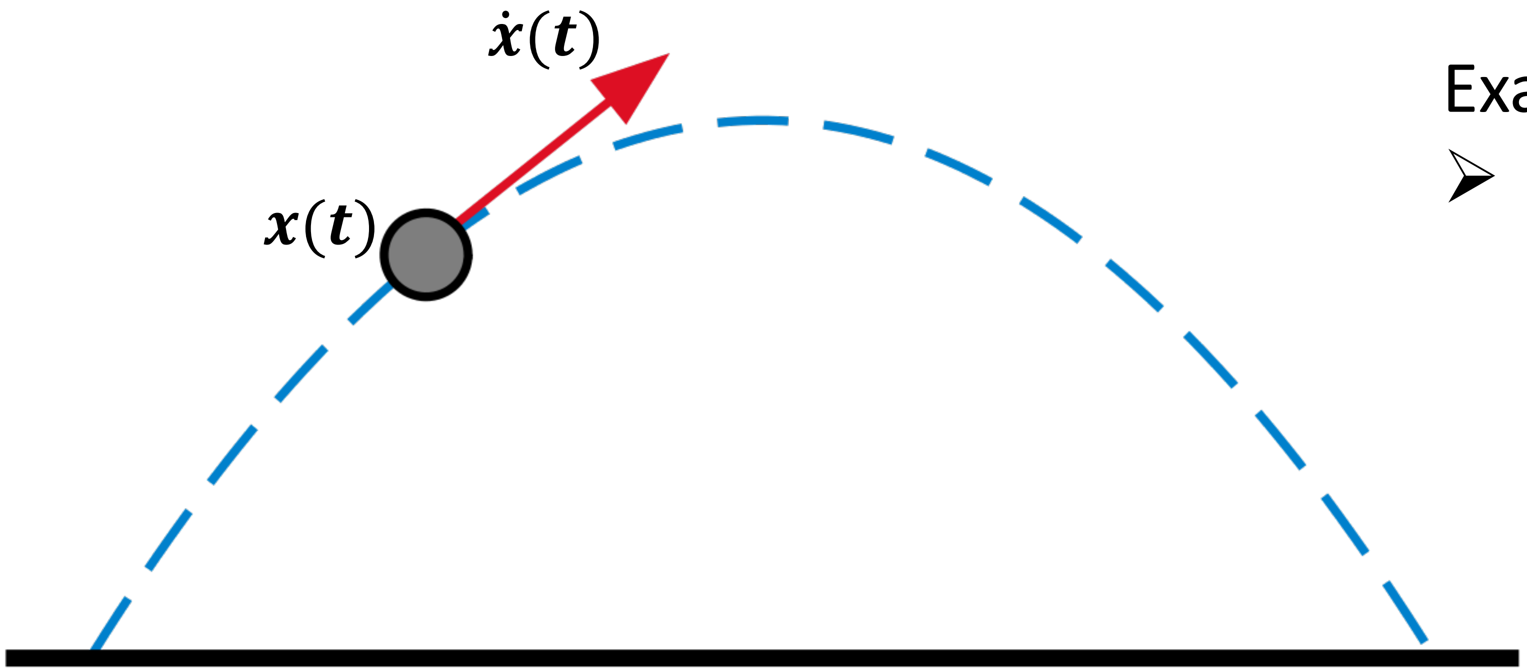
- Perception has the responsibility of converting sensor measurements into a representation of the world and of the robot's current situation.
- Planning uses these representations to reason about the effects of actions in the world.

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

State

The term ***<u>state</u>*** 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.

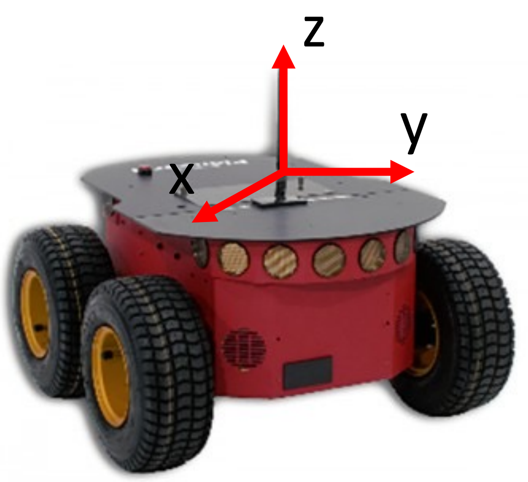
Geometric Representations

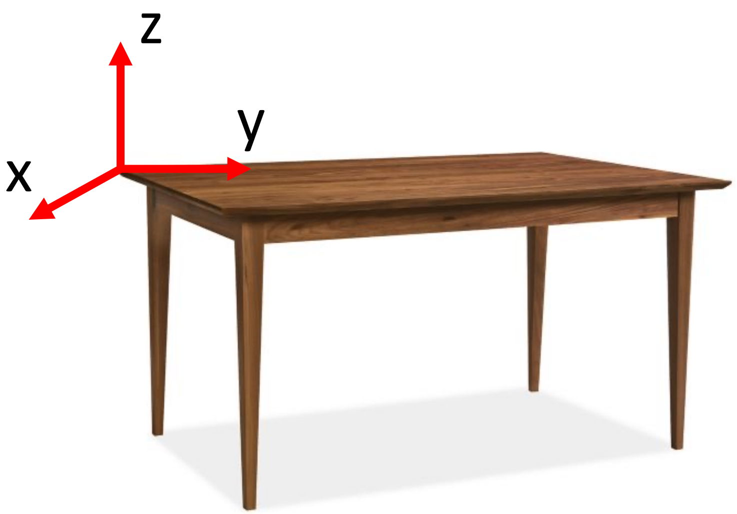
In robotics, we often require specific geometric information.

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!





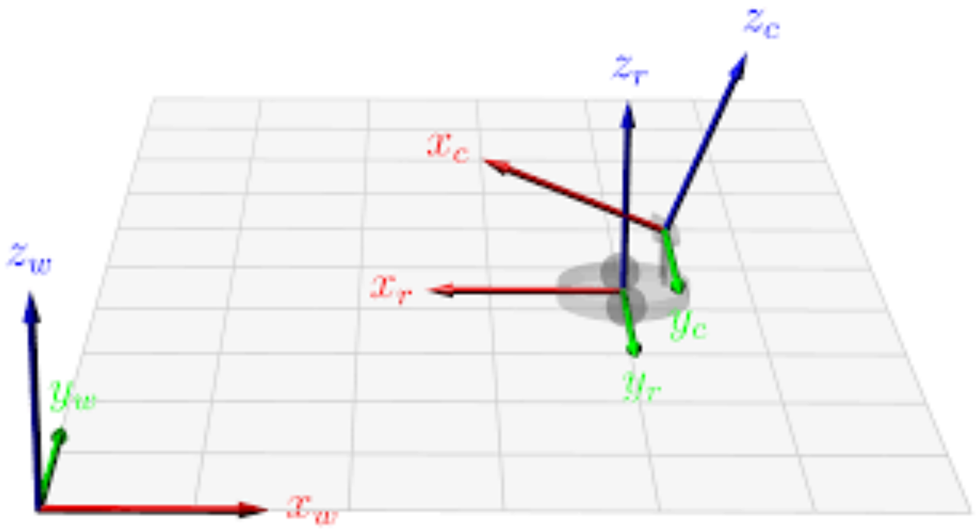
Geometric Representations

In robotics, we often require specific geometric information.

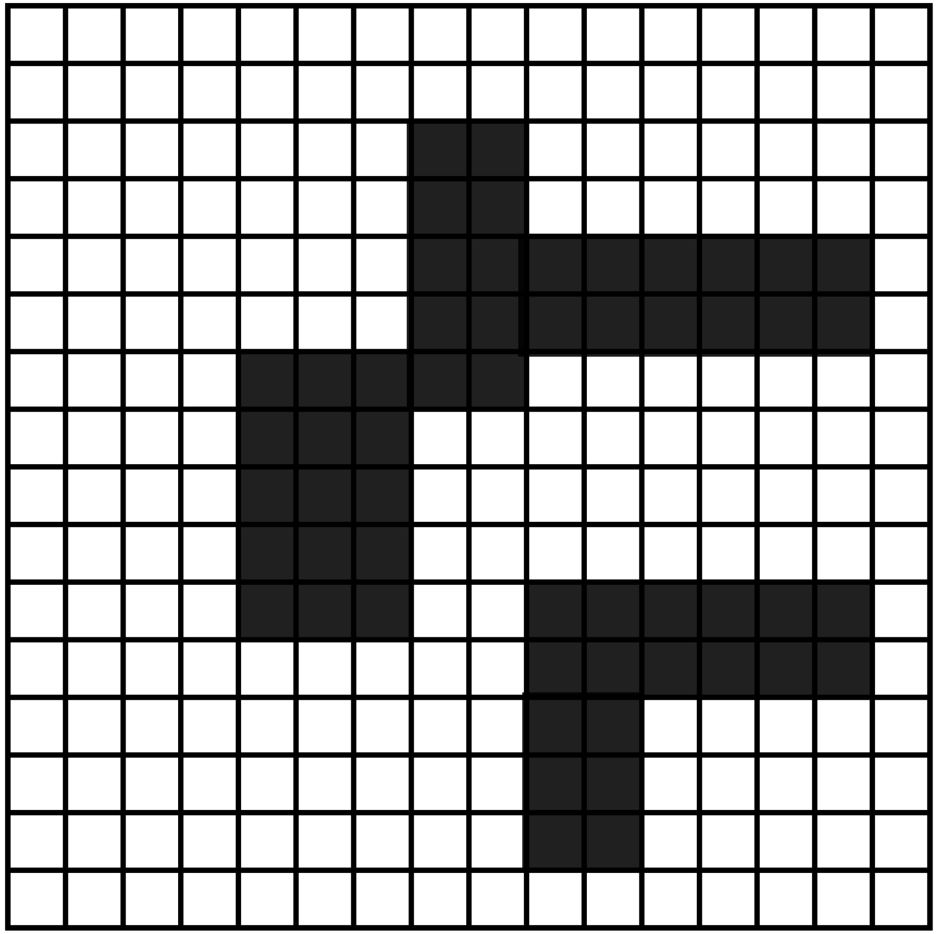
To describe an object's position:

- 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.

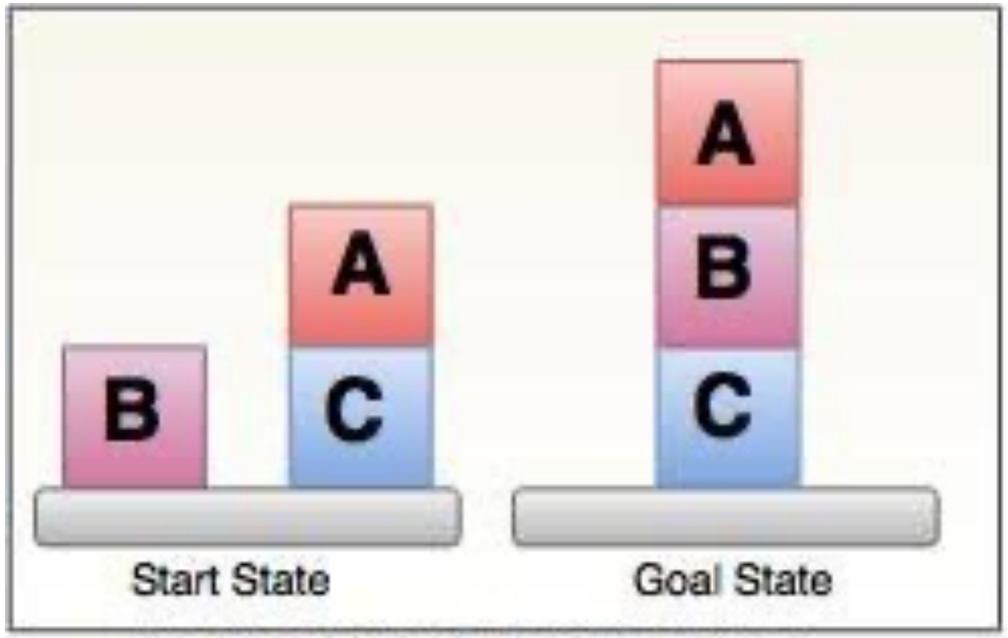


Fig: Blocks-World Planning Problem

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:

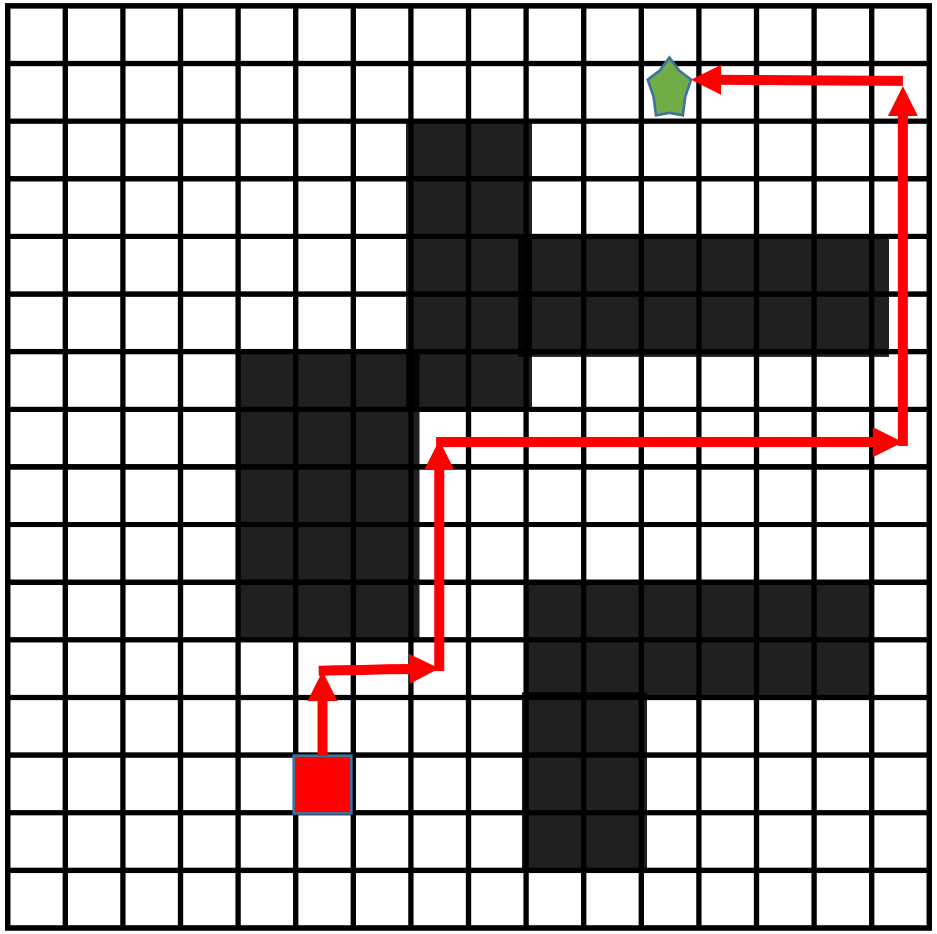
- Preconditions: what must be true in the world before the action is applied?
- Effects: what changes occur in the world after the action occurs?

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

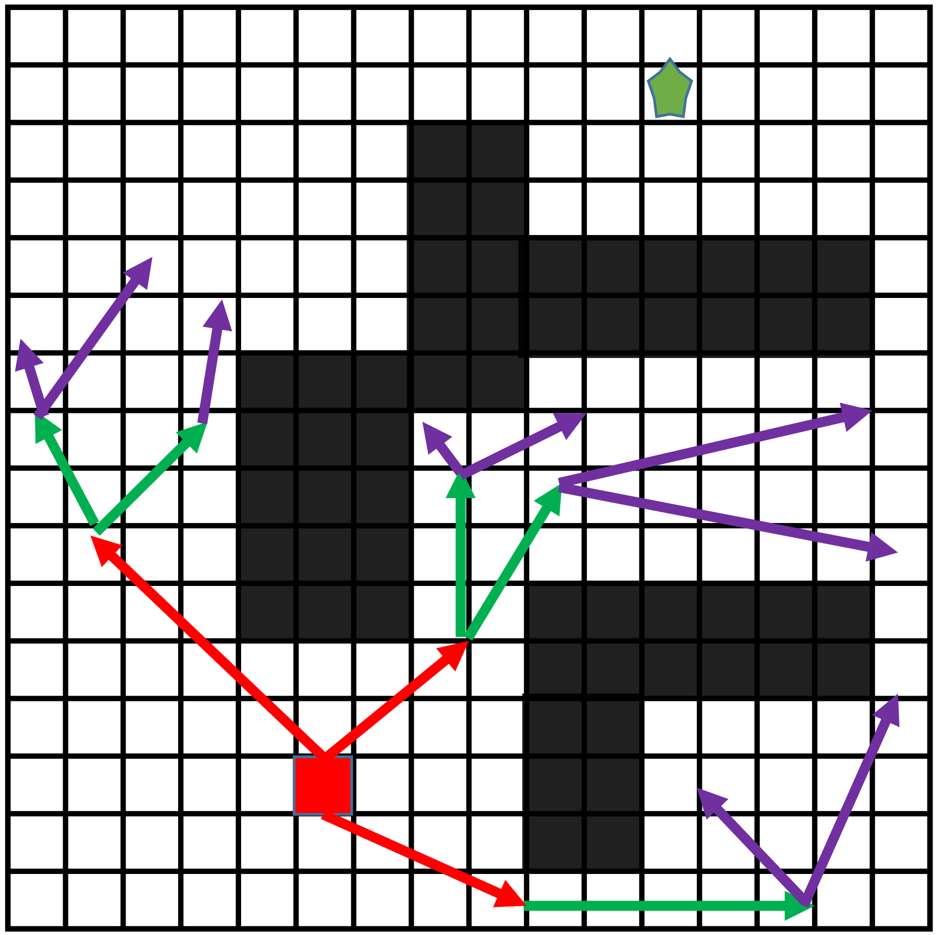
The path planning problem is to find a free path from start to goal.

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

Start position

Goal position

Grid World



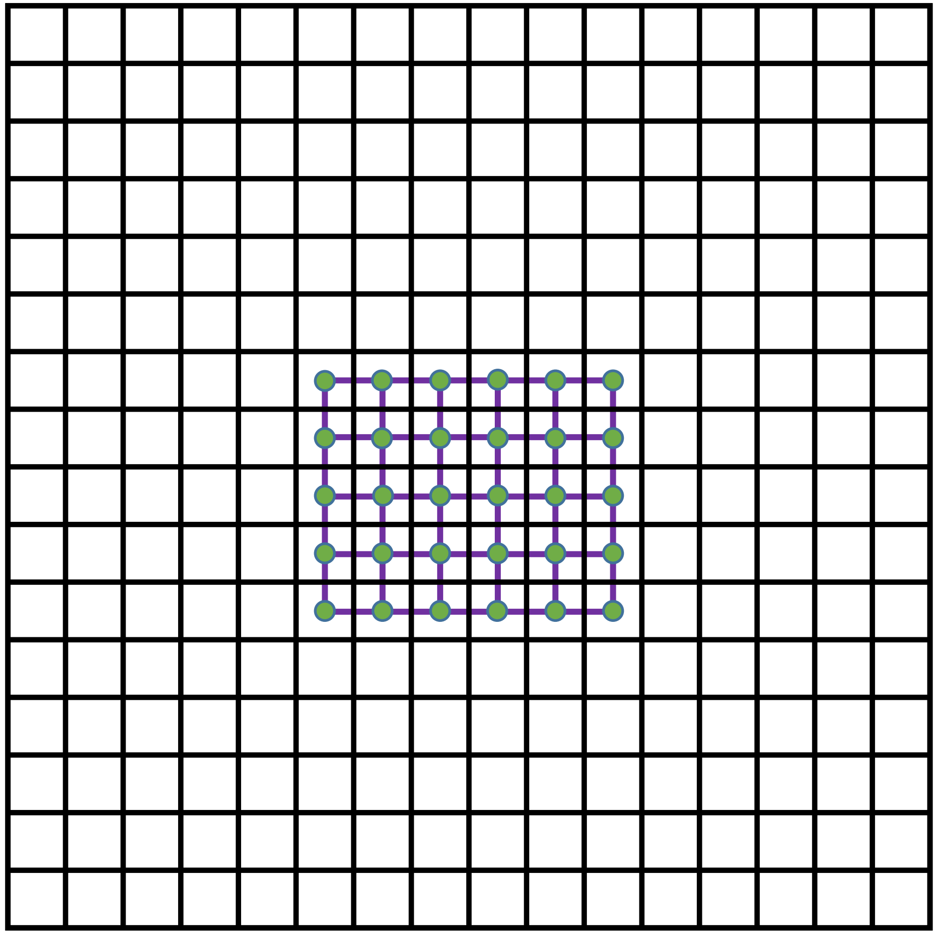
Start position

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?

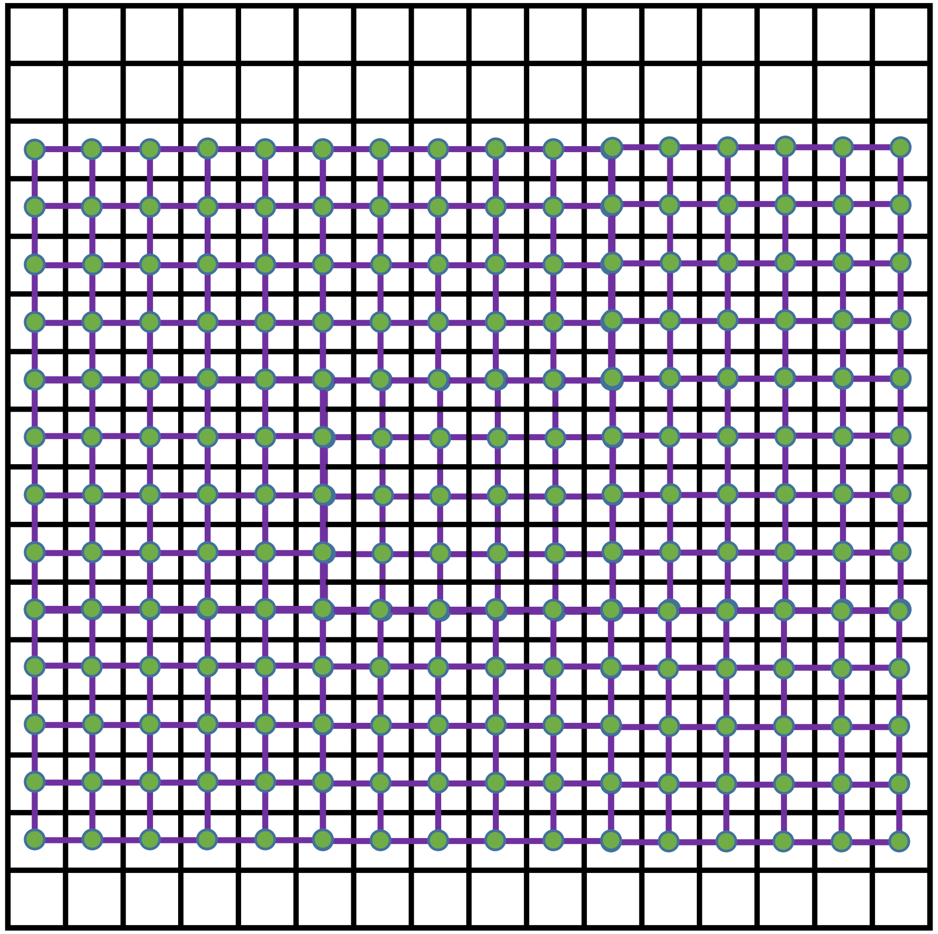
Grid World



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.

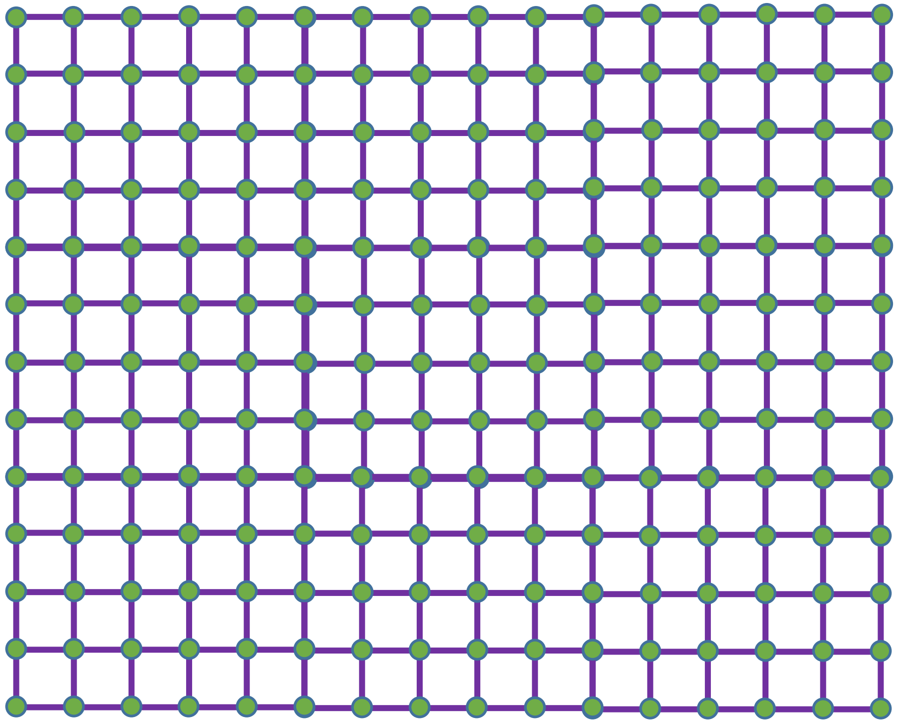
Grid World



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

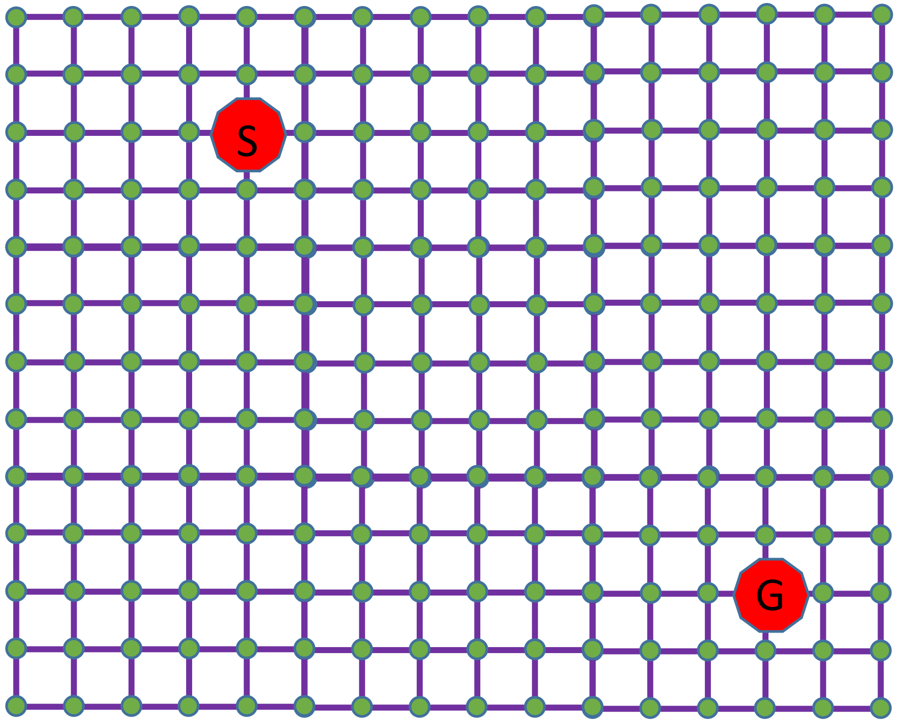


A grid can be represented as a graph:

- 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!

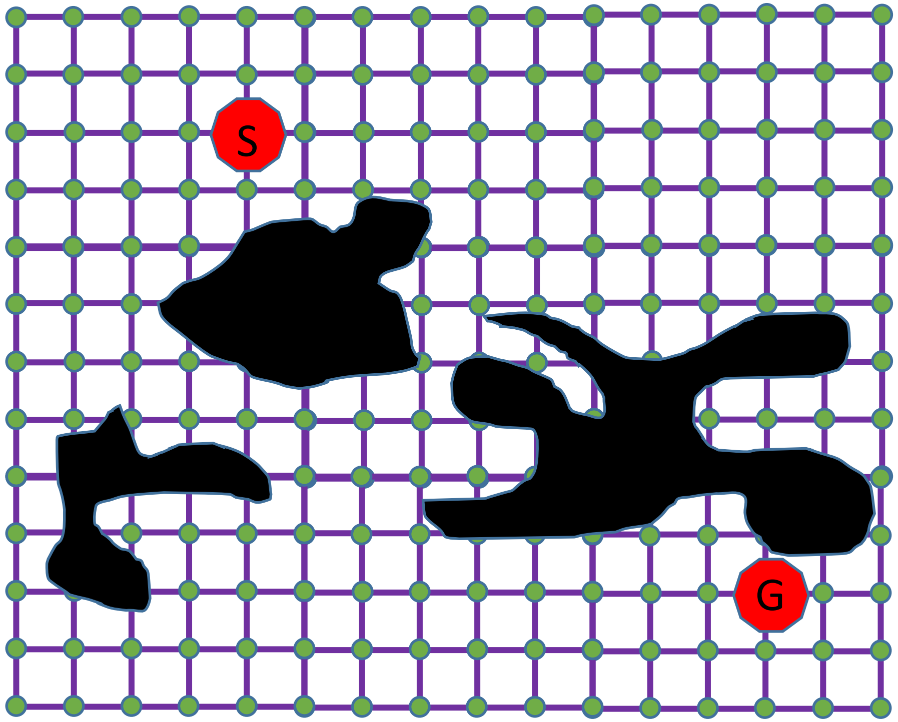
Grid World



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

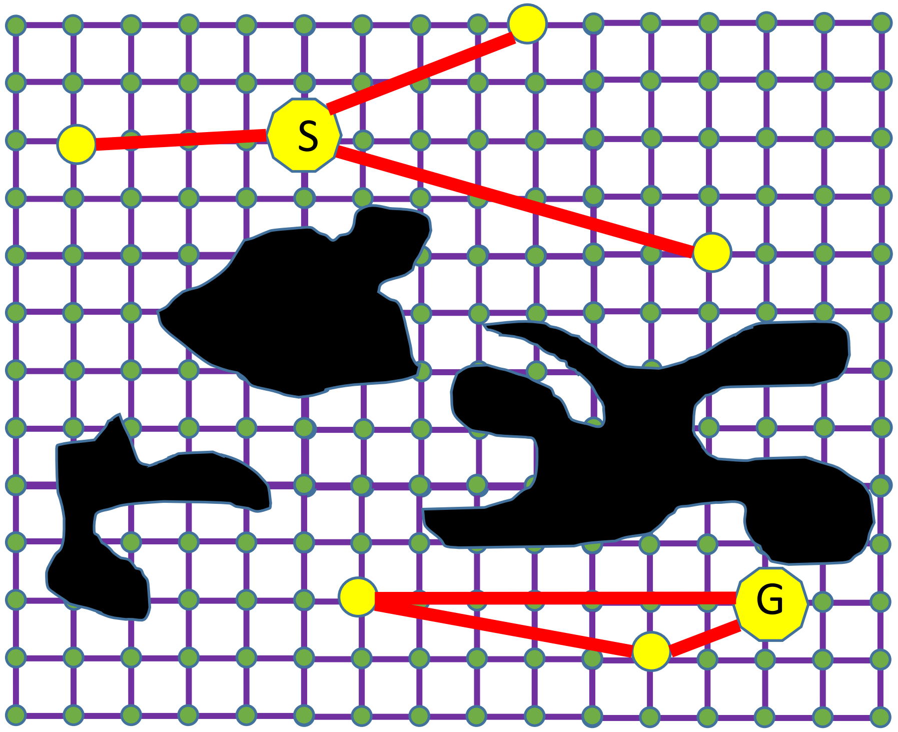


Define a Starting state and a Goal state, and use your favorite graph search algorithm to find a path.

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

- The world state is often 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.

Sensing and Perception

Some Sensors

Pan-Tilt Camera

Intel's RealSense
Depth Camera

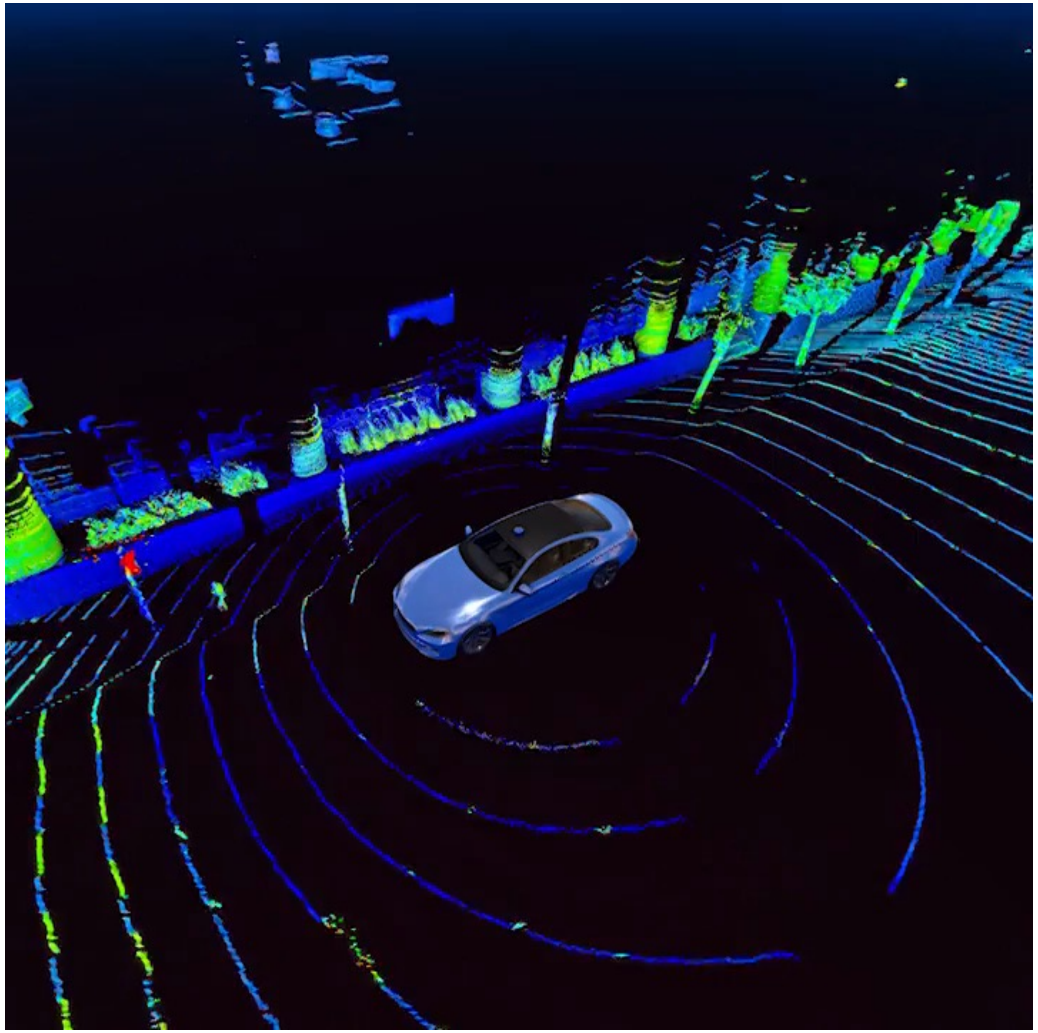
Velodyne LIDAR

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.

Let’s work on a detailed example…

A Trash Sorting Robot

Our first example is a trash sorting robot.

Individual pieces of trash arrive to the robot's work cell on a conveyor belt.

The robot's task is to place each piece of trash in an appropriate bin:

- Glass
- Mixed paper
- Metal
- Nop (Do nothing!)

Sensors measure various characteristics of the trash, which are used to make inferences about the object type (perception).

We assume sensor uncertainty, but perfect execution of actions.

Over time, sensor models can be refined using machine learning methods.

Modeling the World State

For this problem, the only interesting aspect of the world state is the specific **material composition** of the item of trash

We consider five possibilities:

- Cardboard
- Paper
- Cans
- Scrap Metal
- Bottles

For now, we assume that there are no other possibilities.

Modeling Uncertainty in Sensing

We assume that there is uncertainty in sensing.

We consider the state to be a random quantity, with five possible outcomes:

$$\Omega = \{\text{cardboard, paper, cans, scrap metal, bottle}\}$$

In probability theory,

- The set Ω is called the sample space.
- Each $\omega \in \Omega$ is called an outcome.
- A subset $A \subset \Omega$ is called an event.

Denote by $\mathfrak{B} = \{A | A \subset \Omega\}$ the set of all events.

Probability distributions map events to probabilities, $\boldsymbol{P}\colon \boldsymbol{\mathfrak{B}} \rightarrow [\mathbf{0}, \mathbf{1}]$

Examples

Suppose the probabilities associated with the five outcomes are given as:

Category (ω)	$P(\{\omega\})$
Cardboard	0.20
Paper	0.30
Cans	0.25
Scrap Metal	0.20
Bottle	0.05

Compute the following:

- The probability that an item is a paper product: $P(\{A_1\})$
- The probability that an item is a metal product: $P(\{A_2\})$

Answers:

- $P(\{A_1\}) = P(\{cardboard\}) + P(\{paper\}) = 0.5$
- $P(\{A_2\}) = P(\{cans\}) + P(\{scrap\ metal\}) = 0.45$

Define two events

A_1 = {cardboard, paper}

A_2 = {cans, scrap metal}

Some properties of probability distributions

Three Axioms of Probability Theory:

1. For A $\subset \Omega$, P(A) ≥ 0
   - *There's no such thing as negative probability.*
2. P(Ω) =1
   - *The probability that something happened is 1.*
3. For $\mathrm{A_i}, \mathrm{A_j} \subset \Omega$, if $\mathrm{A_i} \cap \mathrm{A_j} = \emptyset$, then $P\left(A_i \cup A_j\right) = P(A_i) + P\left(A_j\right)$
   - *If two events are disjoint (aka mutually exclusive), then the probability that one of the two events occurred equals the sum of the probabilities for the two events.*
   - *The second and third axiom immediately imply that* P($\emptyset$) =0.

A handy relationship:

Since $\boldsymbol{A \cap \bar{A} = \emptyset}$ ***and*** $\boldsymbol{A \cup \bar{A} = \Omega}$***, we can conclude that***

$$\boldsymbol{P(A) + P(\bar{A}) = 1}$$

which implies

$$\boldsymbol{P(\bar{A}) = 1 - P(A)}$$

Proof:

1. $\boldsymbol{A \cap \bar{A} = \emptyset}$ **implies** $\mathbf{P}(\boldsymbol{A \cup \bar{A}}) = \mathbf{P(A) + P}\overline{(\boldsymbol{A})}$ **[Axiom 3]**
2. $\boldsymbol{A \cup \bar{A} = \Omega}$ **impies** $\mathbf{P}(\boldsymbol{A \cup \bar{A}}) = \boldsymbol{P(\Omega) = 1}$ **[Axiom 2]**
3. **Together, 1 and 2 imply** $\mathbf{P(A) + P}\overline{(\boldsymbol{A})} \mathbf{= 1}$

Examples

Suppose the probabilities associated with the five outcomes are given as:

Category (ω)	$P(\{\omega\})$
Cardboard	0.20
Paper	0.30
Cans	0.25
Scrap Metal	0.20
Bottle	0.05

Compute the following:

- The probability that an item is a paper product: $\boldsymbol{P(\{A_1\})}$
- The probability that an item is a metal product: $\boldsymbol{P(\{A_2\})}$
- The probability that an item is not a paper product $\boldsymbol{P(\{\bar{A}_1\})}$

Answers:

- $\boldsymbol{P(\{A_1\}) = P(\{cardboard\}) + P(\{paper\}) = 0.5}$
- $\boldsymbol{P(\{A_2\}) = P(\{cans\}) + P(\{scrap\ metal\}) = 0.45}$
- $\boldsymbol{P(\{\bar{A}_1\}) = P(\Omega - \mathrm{A}_1) = \mathrm{P}(\{\mathrm{cans, scrap\ metal, bottle}\}) = 0.5}$

Define two events

A_1 = {cardboard, paper}

A_2 = {cans, scrap metal}

Prior Probability Distributions

What can we say about the probabilities of various outcome before we even invoke the robot's sensors?

- Our beliefs about the probabilities of various outcomes can be encoded in a ***prior distribution*** --- i.e., the a priori belief about the world.
- Priors can be estimated using data, or can be inferred using domain knowledge (e.g., a fair coin should land on heads 50% of the time).

We estimate prior probabilities using *observed data*:

- Cardboard occurs about 200 times for each 1000 item of trash.
- Paper occurs about 300 times for each 1000 item of trash.
- Cans occur about 250 times for each 1000 item of trash.
- Scrap Metal occurs about 200 times for each 1000 item of trash.
- Bottles occur about 50 times for each 1000 item of trash.

Is there any reason to believe that this approach should work in practice?

Borel's law of large numbers

- Let $A \subset \Omega$ be an event with probability $\mathrm{P}(A) = p$.
- Suppose we run our experiment n times, and we observe that event A occurs $N_n(A)$ times.
- Then, with probability one

$$\frac{N_n(A)}{n} \to p \text{ as } n \to \infty$$

- ***As the number of trials goes to infinity, the proportion of times that an event occurs approaches the probability of that event.***
- ***If we make enough observations, we can start to trust that we have good estimates of prior probabilities!***

Shamelessly taken from Wikipedia.

Machine Learning

In fact, we have just seen a first, simple example of machine learning:

1. ***Count the number of occurrences of each category.***
2. ***Use their relative proportions as an estimate of the prior probability distribution.***

We'll go a bit deeper later...

Simulation by sampling

- Often useful to simulate robot systems. In our case, we might like to simulate the arrival of trash to our sorting system, such that it accurately reflects the prior distribution?

- More on that next time…

shutterstock.com · 622462496

References

Definitions, descriptions and sample code available in the
Introduction to Robotics and Perception book at http://roboticsbook.org

- *Chapter 1: Introduction*
- *Chapter 2: Trash Sorting Robot*