MODULE 2: AUTONOMOUS BEHAVIOR

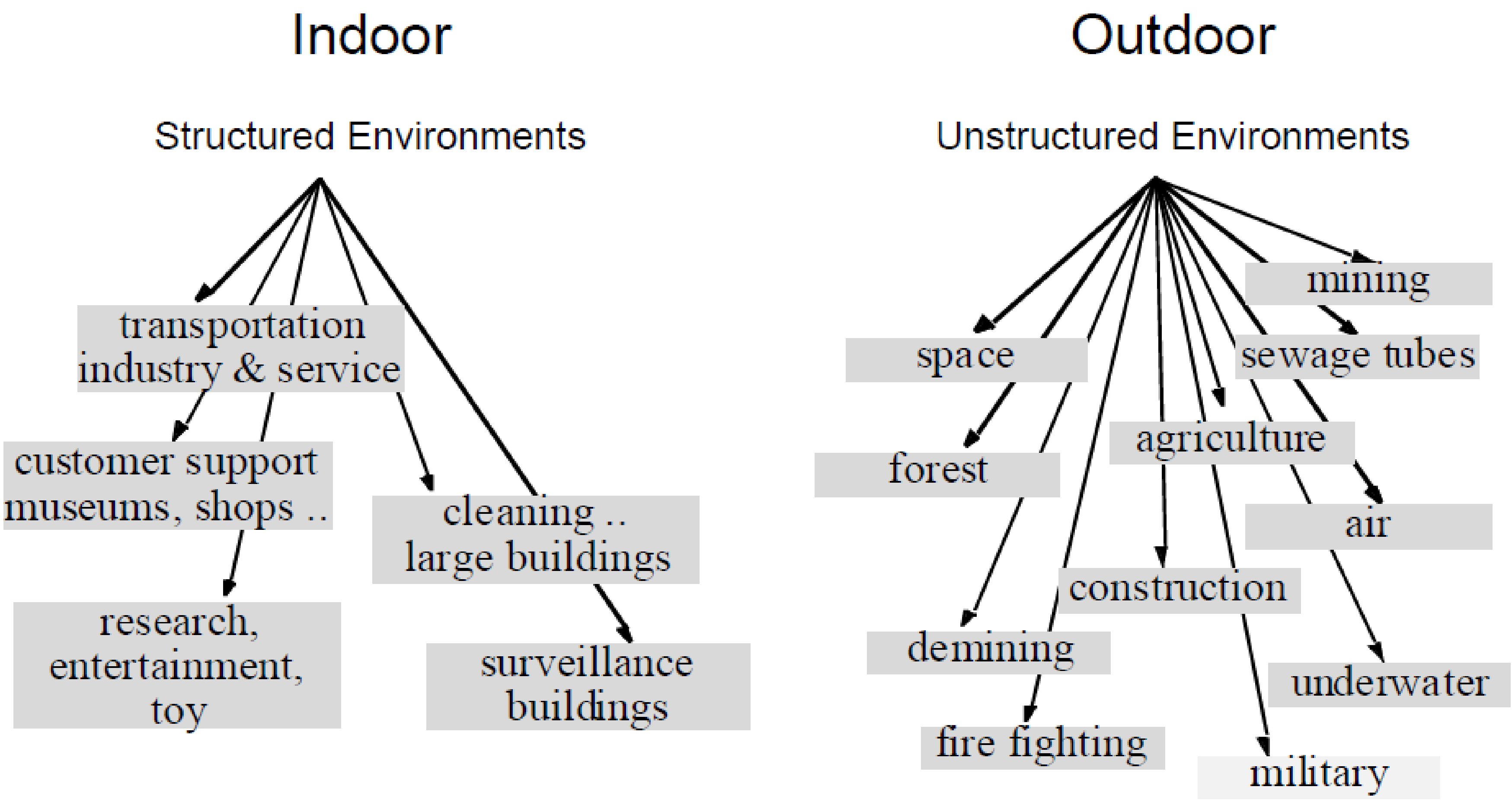
Nicholas HO, PhD

Institute of System Science, NUS

Topics

- Fundamentals of control techniques
- Localization schemes
- Robotic spatial mapping

Applications of Mobile Robots



Autonomous Mobile Robots

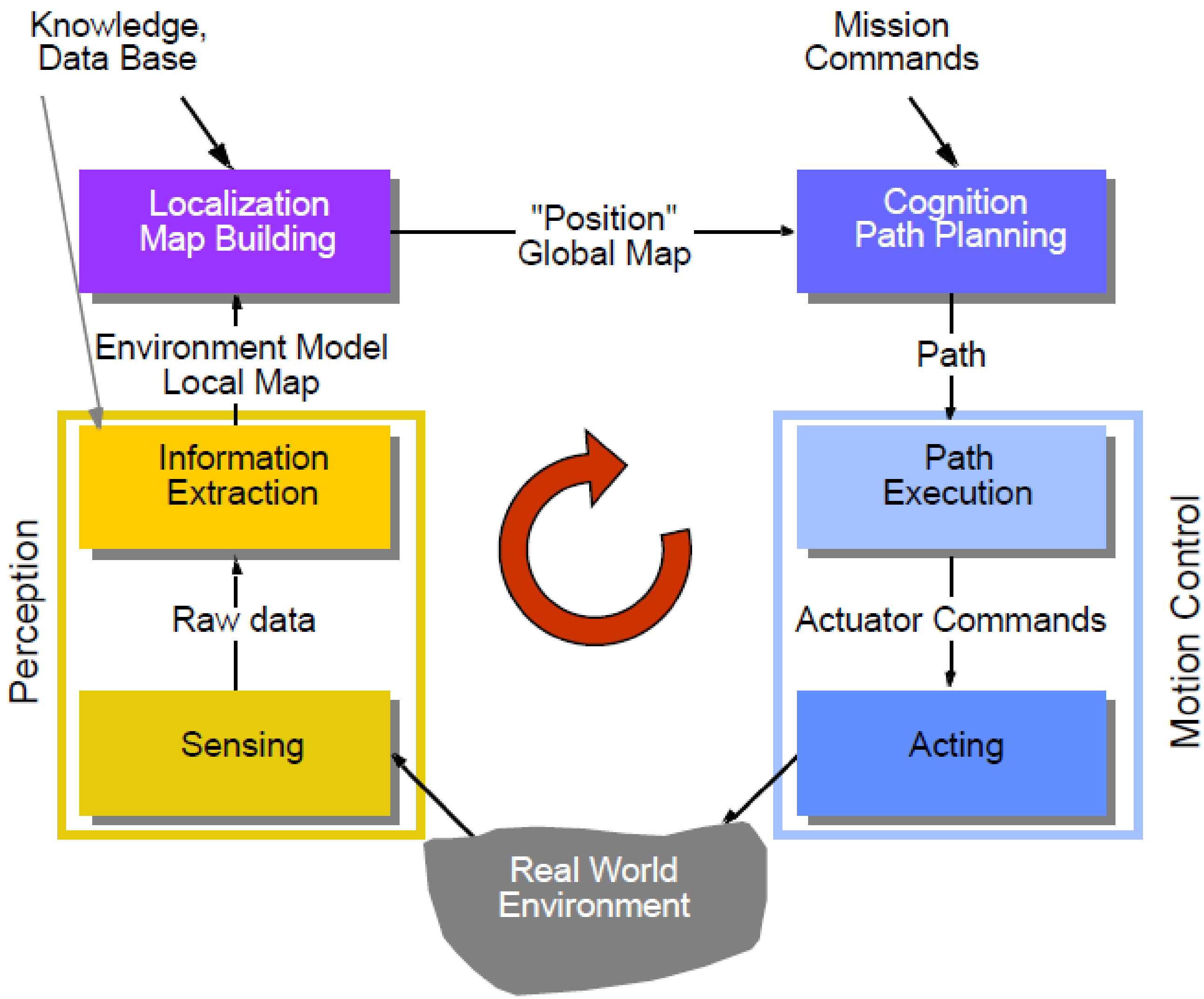
- **The 3 key questions in Mobile Robotics**
  - Where am I ?
  - Where am I going ?
  - How do I get there ?
- **To answer these questions the robot must:**
  - **Model the Environment** – have a model of the environment (given or autonomously built)
  - **Perceive and analyze the environment –** Utilize sensors and algorithms for perception
  - **Localize Itself Within the Environment –** Determine its position (aka localization)
  - **Plan and execute the movement** – Develop a path and navigate (aka navigation)

Autonomous Mobile Robots

Key Components of Autonomous Behavior:

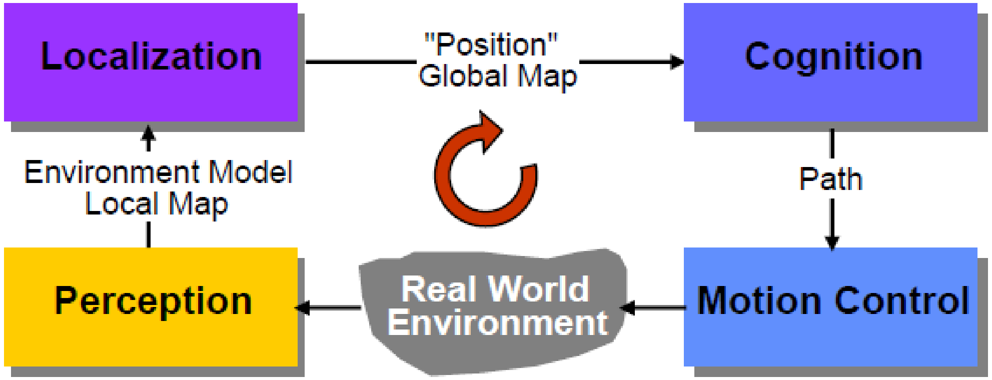
1. **Locomotion and Navigation**:
    - The robot's ability to move and traverse its environment
2. **Perception**:
    - The process of acquiring, interpreting, selecting, and organizing sensory information
3. **Localization**:
    - The method by which a robot determines its position within its environment
4. **Planning and Motion Generation**:
    - The creation of a plan or path to achieve a specific goal and the execution of this plan

General Control Scheme for Mobile Robot Systems



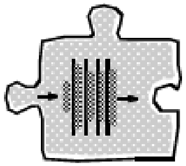
Control Architectures / Strategies

- Control Loop
  - dynamically changing
  - no compact model available
  - many sources of uncertainty

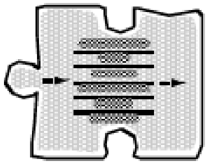


Two Approaches

- *Classical AI*
  - *complete modeling*
  - *function based*
  - *horizontal decomposition*

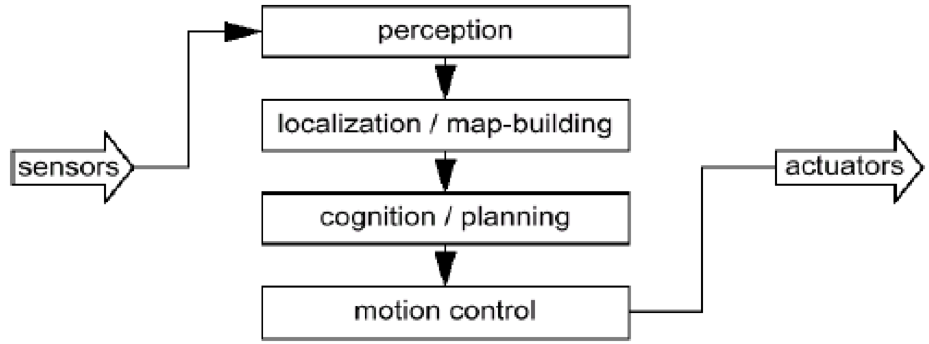


- *New AI, AL*
  - *sparse or no modeling*
  - *behavior based*
  - *vertical decomposition*
  - *bottom up*

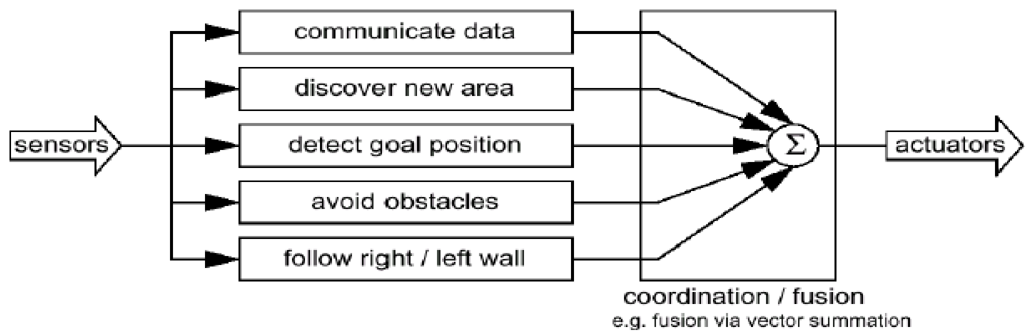


Control Architectures: Two Approaches

Classical AI (model-based navigation)



New AI, AL (behavior-based navigation)



Control Architectures:

Classical AI (model-based navigation)	New AI, AL (behavior-based navigation)
Complete Modeling – Requires a detailed and comprehensive model of the environment	**Sparse or No Modeling** – Operates with minimal or no detailed environmental representation
Function-Based – Uses predefined functions and rules to handle various tasks	**Behavior-Based** – Focuses on the emergence of intelligent behavior from simple, low-level interactions
Horizontal Decomposition – Divides the problem into sub-tasks at the same level, which are then solved sequentially	**Vertical Decomposition** – Breaks down tasks into layers, starting with simple behaviors that build up to more complex actions
Top-Down Approach – Starts with a high-level plan and decomposes it into smaller tasks	**Bottom-Up Approach** – Begins with basic behaviors and combines them to form higher-level functionality
Deliberative Planning – Involves extensive computation and planning before execution	**Reactive Planning** – Responds to the environment in real-time, with little to no prior planning
High Computational Requirements – Needs significant computational power and memory to handle complex models	**Low Computational Requirements** – Can function effectively with limited computational resources
Rigid and Less Adaptive – Can struggle with dynamic and unpredictable environments	**Flexible and Adaptive** – Well-suited for dynamic and changing environments

Control Architectures:

1. Classical AI (model-based navigation)

Advantages:

- **Precision and Accuracy**: Can achieve high levels of accuracy due to detailed modeling
- **Comprehensive Planning**: Thorough and deliberate planning ensures that all scenarios are considered
- **Predictability**: More predictable behavior since it follows predefined rules and functions

Disadvantages:

- **High Computational Requirements**: Needs significant computational power and memory
- **Inflexibility**: Less adaptable to dynamic and unpredictable environments
- **Complexity**: Creating and maintaining detailed models can be complex and time-consuming

2. New AI, AL (behavior-based navigation)

Advantages:

- **Adaptability and Flexibility**: Highly adaptable to changing and dynamic environments
- **Efficiency**: Requires less computational power and resources
- **Simplicity**: Easier to implement and scale, especially in real-time applications
- **Robustness**: Can handle unexpected situations better due to real-time reactive planning

Disadvantages:

- **Less Precision**: May not achieve the same level of precision as model-based approaches
- **Limited Long-Term Planning**: Focuses on short-term, reactive behaviors rather than long-term planning
- **Emergent Behavior**: Can sometimes exhibit unexpected or undesired behaviors due to the complexity of interactions

Control Architectures:

Combining Both Approaches: Hybrid AI

- **Integrate Model-Based and Behavior-Based Strategies** – Leverage the strengths of both approaches for a more robust and adaptable system
- **Balanced Computation and Adaptation** – Utilize detailed modeling where necessary while allowing for real-time behavioral responses to changes in the environment
- **Enhanced Performance** – Achieve better performance and reliability by combining deliberate planning with reactive adaptability

Advantages:

- **Best of Both Worlds**: Leverages the precision and planning of Classical AI with the adaptability and efficiency of New AI
- **Balanced Performance**: Achieves a good balance between computational efficiency and adaptability
- **Robust and Reliable**: More robust and reliable in a wider range of scenarios, including both predictable and dynamic environments

Disadvantages:

- **Integration Complexity**: Combining both approaches can be complex and may require sophisticated integration strategies
- **Resource Management**: Balancing the computational resources and ensuring smooth interaction between the two approaches can be challenging

Control Architectures:

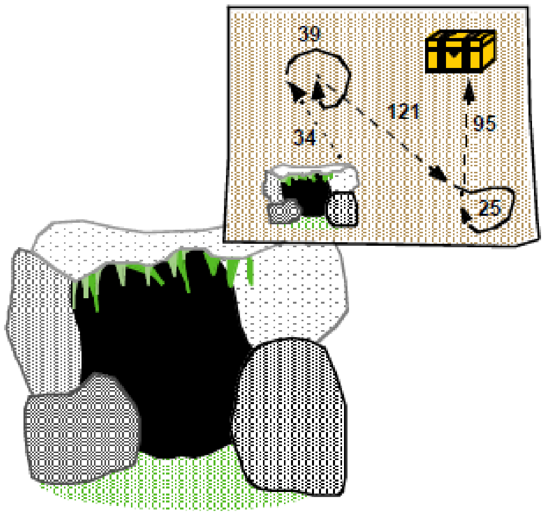
What to choose?

- **For Structured and Predictable Environments:** Classical AI is generally better due to its precision and comprehensive planning
- **For Dynamic and Unpredictable Environments:** New AI is preferable because of its flexibility and adaptability
- **For Complex and Varied Scenarios:** A Hybrid Approach is often the best solution, combining the strengths of both to handle a wide range of situations effectively

In many modern applications, a hybrid approach is favored to balance the strengths and weaknesses of both approaches

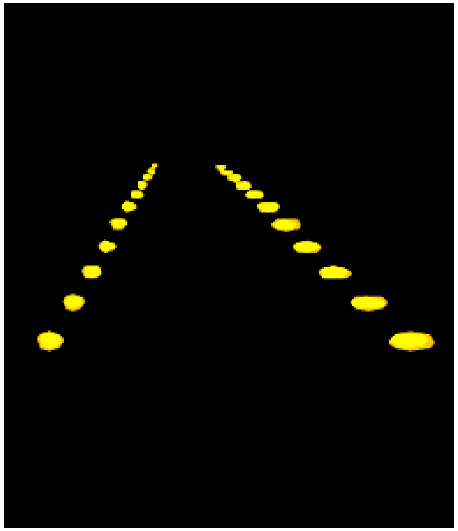
Environment Representation and Modelling

- Odometry



How to find a treasure

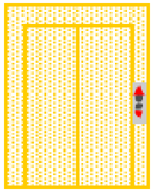
- Modified Environments



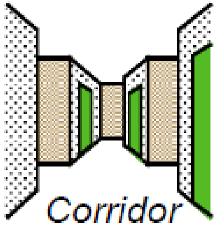
Landing at night

> ➢ *expensive, inflexible*

- Feature-based Navigation

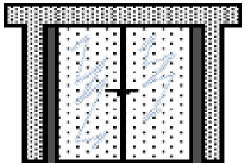


Elevator door



Corridor crossing

Eiffel Tower

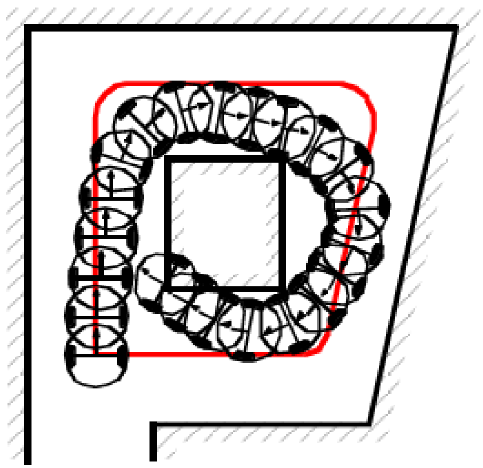


Entrance

> ➢ *still a challenge for artificial systems*

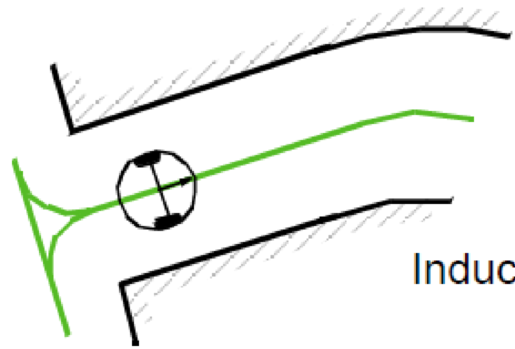
Methods for Navigation: Approaches with Limitations

- Incrementally
  (dead reckoning)

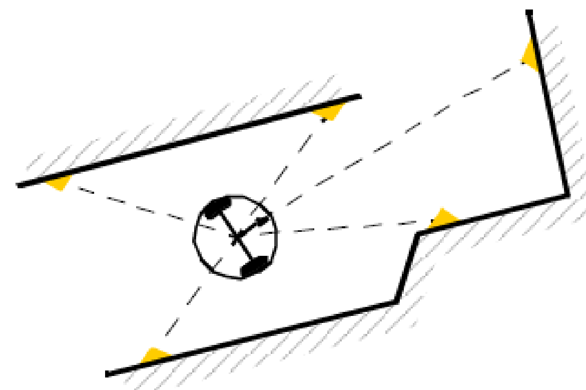


Odometric or initial sensors
(gyro)

- Modifying the environments
  (artificial landmarks / beacons)



Inductive or optical tracks (AGV)

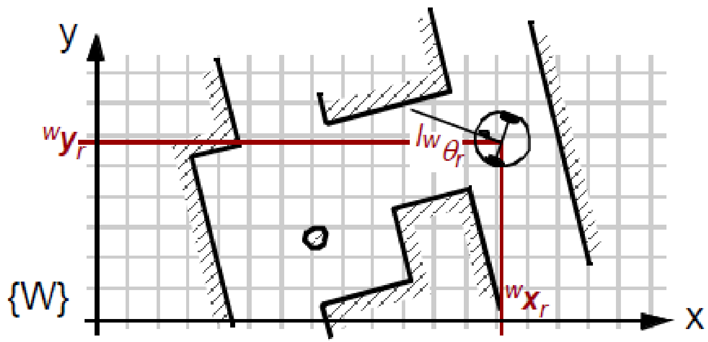


Reflectors or bar codes

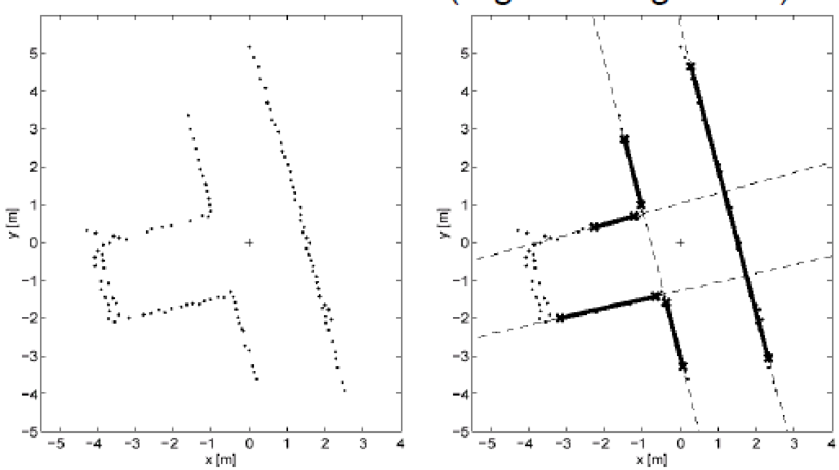
➢ *expensive, inflexible*

Method for Localization: The Quantitative Metric Approach

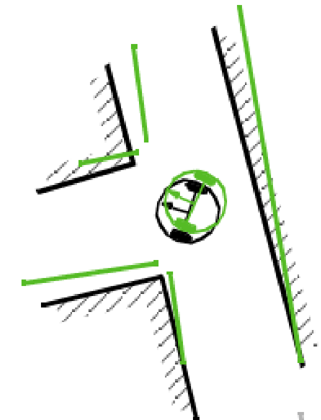
1. A priori Map: Graph, metric



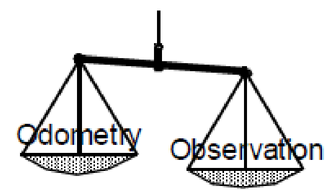
2. Feature Extraction (e.g. line segments)

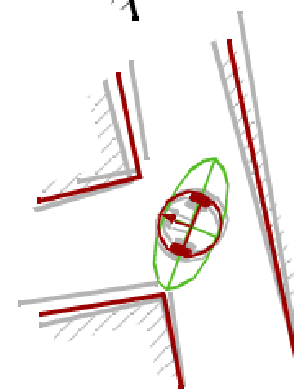


3. Matching:
Find correspondence of features



4. Position Estimation:
e.g. Kalman filter, Markov

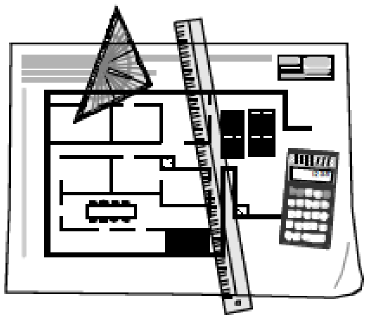




- representation of uncertainties
- optimal weighting according to a priori statistics

Map Building: How to Establish a Map

1. By Hand



2. Automatically: **Map Building**

The robot **learns** its environment

Motivation:

- by hand: hard and costly
- dynamically changing environment
- different look due to different perception

3. Basic Requirements of a Map:

- *a way to incorporate newly sensed information into the existing world model*
- *information and procedures for estimating the robot's position*
- *information to do path planning and other navigation task (e.g. obstacle avoidance)*

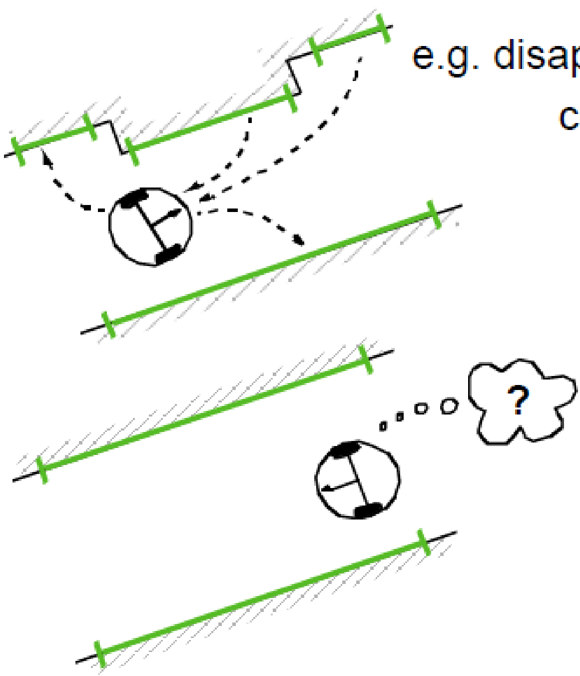
predictability

- Measure of Quality of a map
  - *topological correctness*
  - *metrical correctness*
- But: Most environments are a mixture of predictable and unpredictable features → hybrid approach

model-based vs. behaviour-based

Map Building: The Problems

1. Map Maintaining: Keeping track of changes in the environment

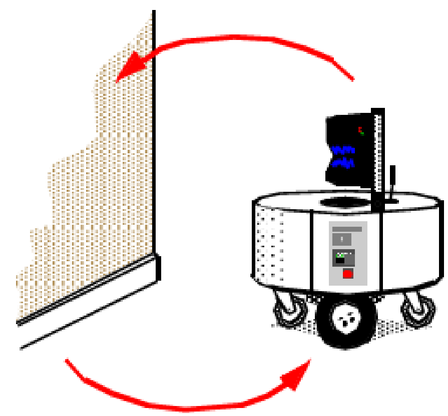


e.g. disappearing cupboard

- e.g. measure of belief of each environment feature

2. Representation and Reduction of Uncertainty

position of robot -> position of wall

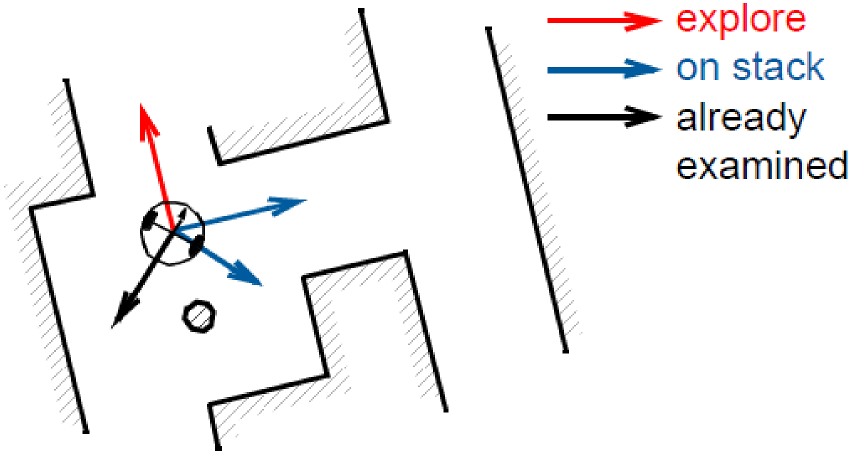


position of wall -> position of robot

- probability densities for feature positions
- additional exploration strategies

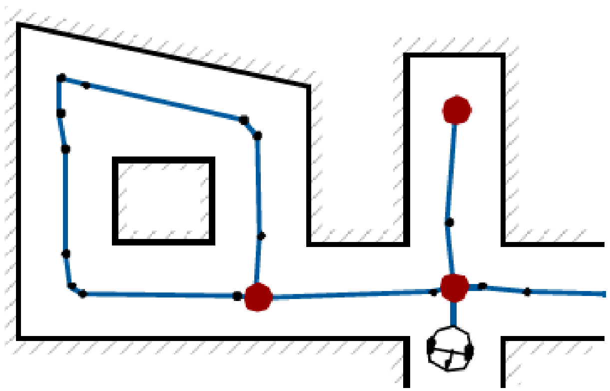
Map Building: Exploration and Graph Construction

1. Exploration



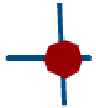
- provides correct topology
- must recognize already visited location
- backtracking for unexplored openings

2. Graph Construction



Where to put the nodes?

- Topology-based: at distinctive locations

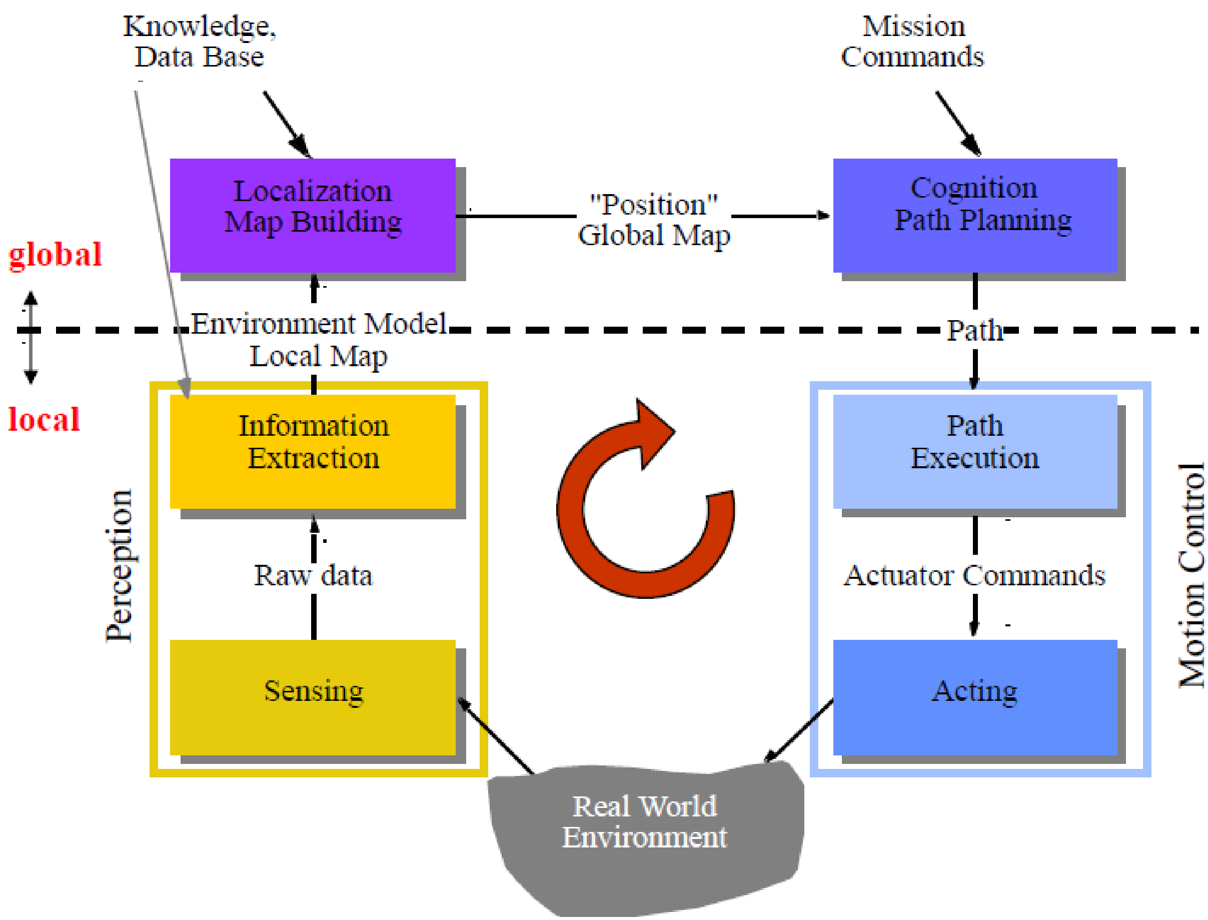


- Metric-based: where features disappear or get visible

State-of-the-Art:
Current Challenges in Map Representation

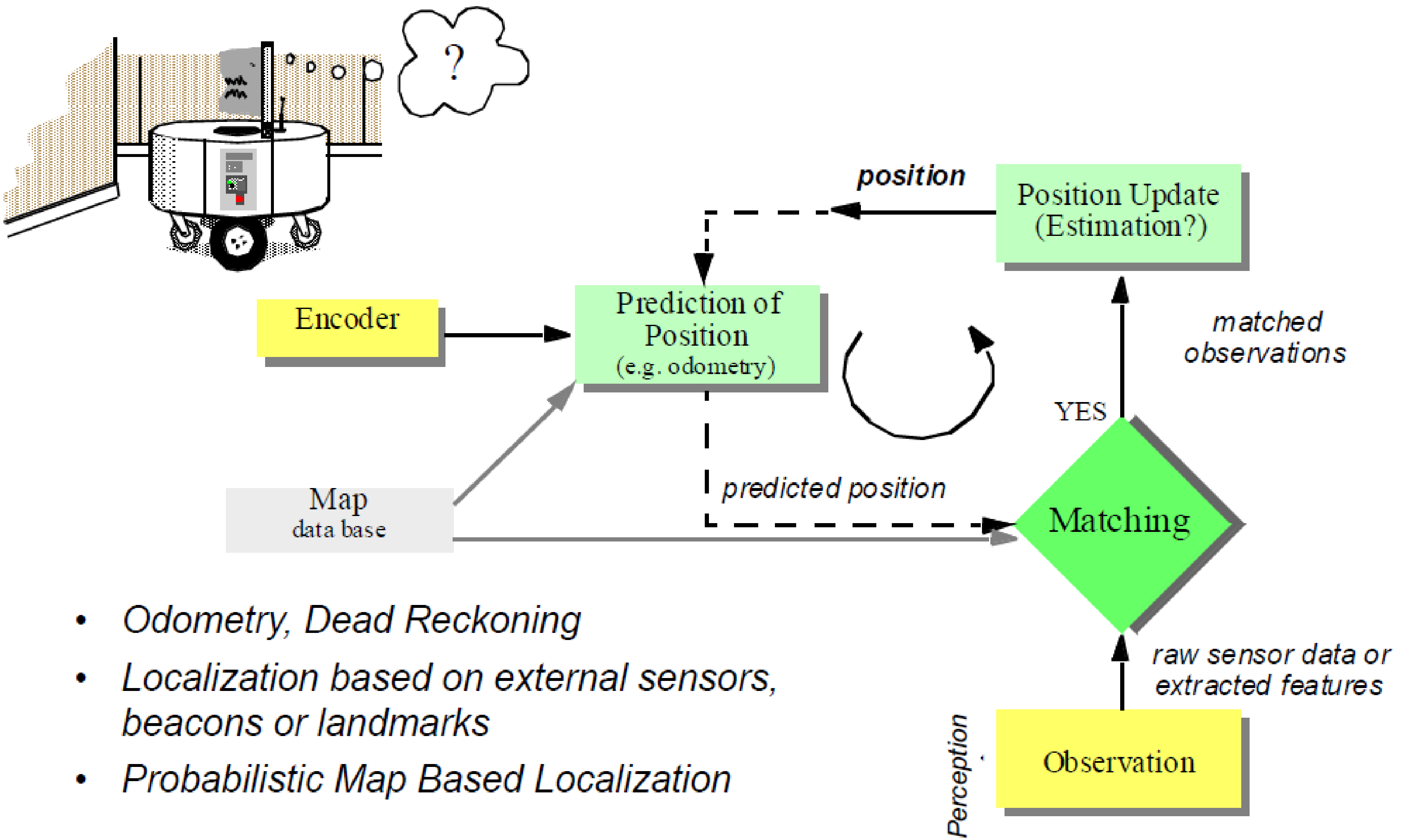
- **Real world is dynamic**
- **Perception is still a major challenge**
  - Error prone, e.g. environmental factors like lighting changes or obstacles can interfere with sensor readings
  - Difficulty of extracting useful information, e.g. identifying and classifying objects or landmarks from a cluttered background.
- How to **build up topology** (boundaries of nodes)
- Travelling across **open space (i.e. without any surrounding landmarks)**
- **Sensor fusion** – Integrating data from multiple sensors to create a coherent and accurate map, e.g. Combining LIDAR, camera, GPS, and IMU data to enhance localization and mapping

Control of Mobile Robots



- Most functions for safe navigation are ’local’ not involving localization nor cognition
- Localization and global path planning ➔ slower update rate, only when needed
- This approach is pretty similar to what human beings do

Localization, Where am I?

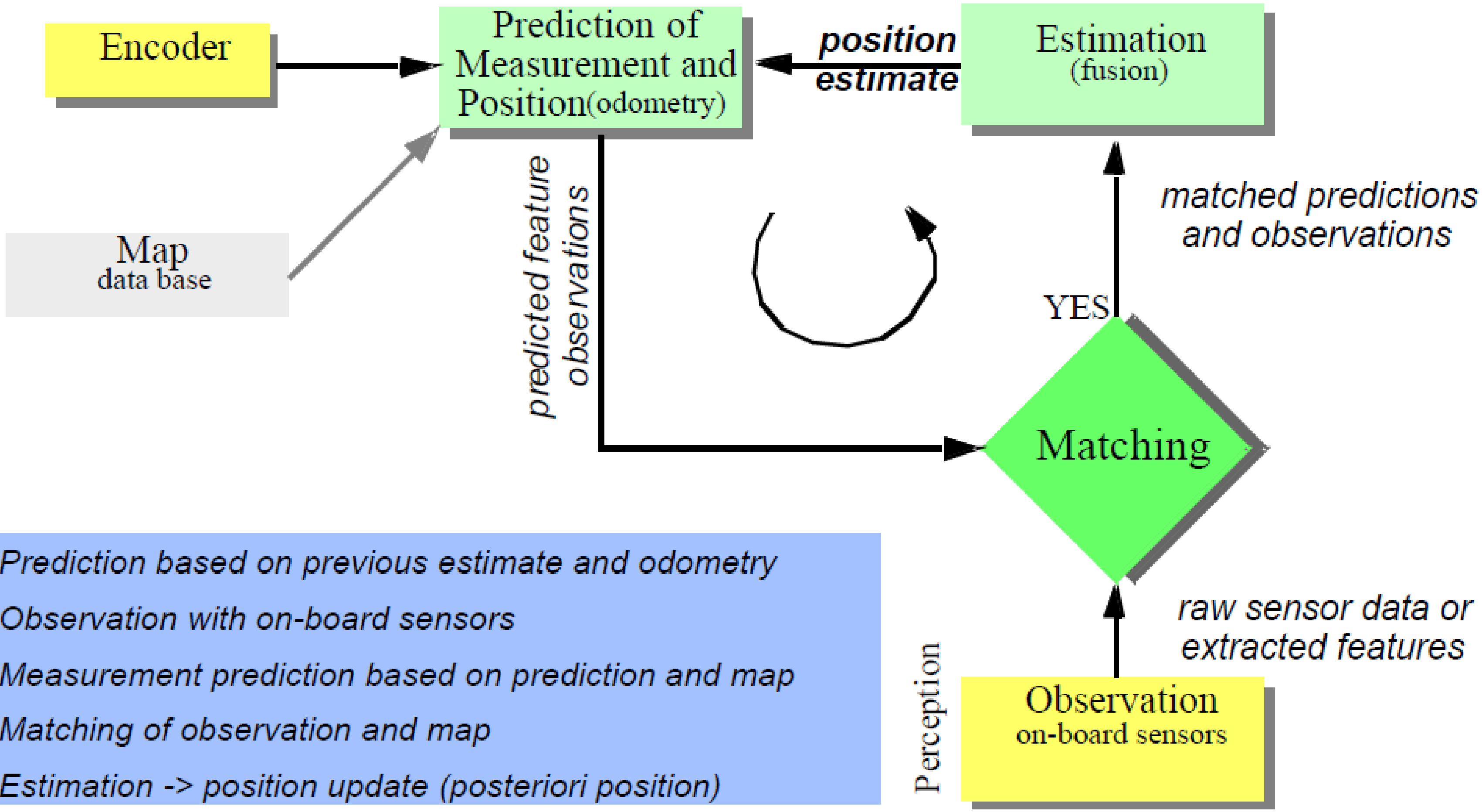


- *Odometry, Dead Reckoning*
- *Localization based on external sensors, beacons or landmarks*
- *Probabilistic Map Based Localization*

Probabilistic, Map-Based Localization

- Consider a mobile robot moving in a known environment
- As it start to move, say from a precisely known location, it might **keep track of its location using odometry**
- However, **after a certain movement the robot will get very uncertain about its position**
  - update using an observation of its environment.
- **observation also leads to an estimate of the robots position which data can than be fused with the odometric estimation** to get the best possible update of the robot’s actual position

The Five Steps for Map-Based Localization



1. Prediction based on previous estimate and odometry
2. Observation with on-board sensors
3. Measurement prediction based on prediction and map
4. Matching of observation and map
5. Estimation -> position update (posteriori position)

Challenges of Localization

- **Knowing the absolute position (e.g. GPS) is not sufficient**
  - Environmental Dynamics: GPS provides absolute coordinates but lacks context about the immediate surroundings, obstacles, and changes in the environment
  - Indoor Limitations: GPS signals are often unreliable or unavailable indoors, making it inadequate for indoor localization
- **Localization in human-scale context**
  - Relative Positioning: It's essential to understand the robot's or device's position relative to objects, landmarks, and humans in its environment
  - Context Awareness: Accurate localization must consider the spatial relationships and interactions with dynamic elements, such as people moving around
- **Cognition and planning require more than just position**
  - Environmental Understanding: Effective planning and decision-making require a comprehensive understanding of the environment, including the layout, objects, and potential hazards
  - Sensor Fusion: Combining data from various sensors (e.g., LIDAR, cameras, IMUs) is necessary to create a detailed and reliable environmental model
  - Behavioral Insights: In addition to positional data, understanding the behaviors and patterns of movement within the environment is crucial for effective navigation and interaction

Challenges of Localization

- **Perception and motion** are crucial for the effective functioning of autonomous systems, enabling them to understand and navigate their environments safely and efficiently
- **Key Challenges**
  - **Sensor noise** - Random inaccuracies in sensor data degrade the robot's perception accuracy
  - **Sensor aliasing** - Similar sensor readings from different environmental states cause confusion in robot perception
  - **Effector noise** - Inaccuracies in actuator outputs lead to deviations from the robot's intended movements
  - **Odometric position estimation** - Cumulative errors in position estimation from wheel encoders cause navigation drift over time

Sensor Noise

- Sensor noise in **mainly influenced by environment factors** e.g. surface, illumination …
- **or by the measurement principle itself**. E.g. interference between similar types of sensors (e.g., ultrasonic sensors), inherent design and functioning of the sensor can introduce noise
- Sensor noise can drastically reduce the reliability and accuracy of the sensor readings, leading to incorrect perception of the environment

Sensor Noise (Cont)

The solution is:

1. to take **multiple reading** into account
   - **Averaging:** Take multiple readings and average them to reduce the impact of random noise
   - **Outlier Detection:** Identify and discard outliers to improve accuracy
2. employ **temporal fusion**
   - **Time-Based Analysis:** Combine sensor data over time to filter out noise and enhance signal quality
   - **Kalman Filtering:** Use algorithms like Kalman filters to predict and correct sensor readings based on past data
3. employ **multi-sensor fusion**
   - **Complementary Sensors:** Use different types of sensors to complement each other's strengths and weaknesses
   - **Data Integration:** Employ techniques to integrate data from multiple sensors to create a more accurate and reliable perception of the environment

Sensor Aliasing

Imagine a robot is navigating through a house using a LIDAR sensor, which measures distances to objects by bouncing laser beams off them

Different Environmental States:

- **Living Room**: The robot is in the living room, and there's a large, flat wall in front of it
- **Kitchen**: The robot is in the kitchen, and there's a large, flat refrigerator in front of it

Similar Sensor Readings:

- **LIDAR Readings**: In both the living room and the kitchen, the LIDAR sensor will detect a large, flat surface at a certain distance. The wall and the refrigerator reflect the laser beams similarly, producing similar distance readings

Ambiguity in Perception:

- **Confusion**: Because the LIDAR readings are similar, the robot might get confused and think it's in the living room when it's actually in the kitchen, or vice versa

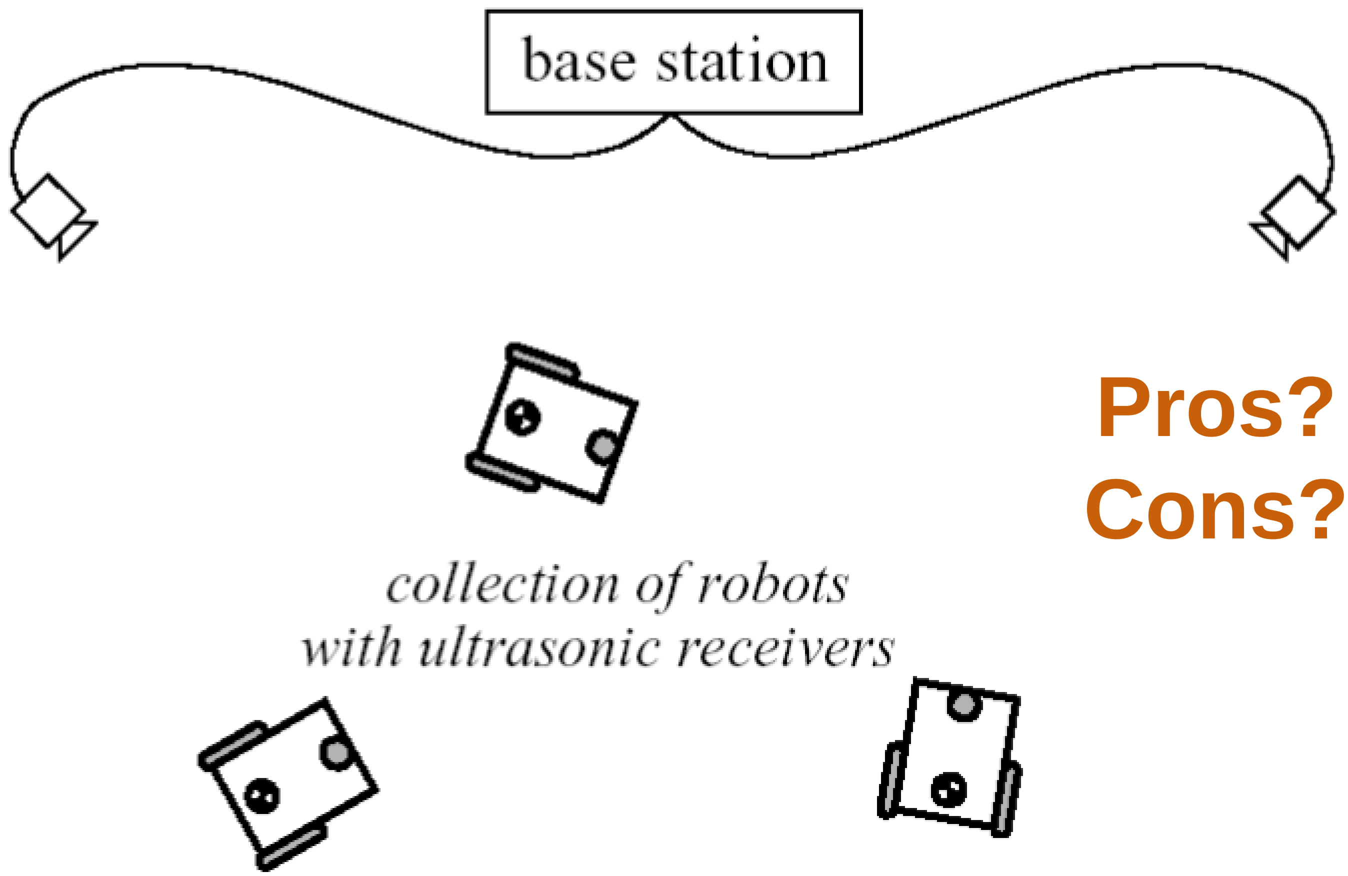
Resolving the Ambiguity:

- **Use Multiple Sensors**: Combine data from other sensors, like a camera that can see the color and texture differences between a wall and a refrigerator
- **Collect More Readings Over Time**: Move around and take more readings from different angles to gather more information and distinguish between the two locations
- **Compare with Map Data**: Use a pre-built map of the house to compare and determine the most likely location based on additional context

Sensor Aliasing (Cont)

- **Non-Uniqueness of Sensor Readings**
  - **Common Occurrence:** In robots, it is common for sensors to produce similar readings for different states of the environment
  - **Many-to-One Mapping:** Even with multiple sensors, there is often a many-to-one mapping from environmental states to the robot's perceptual inputs
- **Information Insufficiency**
  - **Single Reading Limitation:** Amount of information perceived by the sensors is insufficient to accurately identify the robot's position or state from a single reading
  - **Need for Multiple Readings:** To overcome aliasing, robots rely on a series of sensor readings over time
- **Localization Strategy**
  - **Series of Readings:** Robot localization is typically based on analyzing a sequence of readings, rather than a single snapshot
  - **Temporal Information Recovery:** By continuously gathering data, the robot can accumulate sufficient information to resolve ambiguities and accurately determine its position
  - **Algorithmic Solutions:** Algorithms like particle filters or extended Kalman filters can help in integrating multiple readings to improve localization accuracy

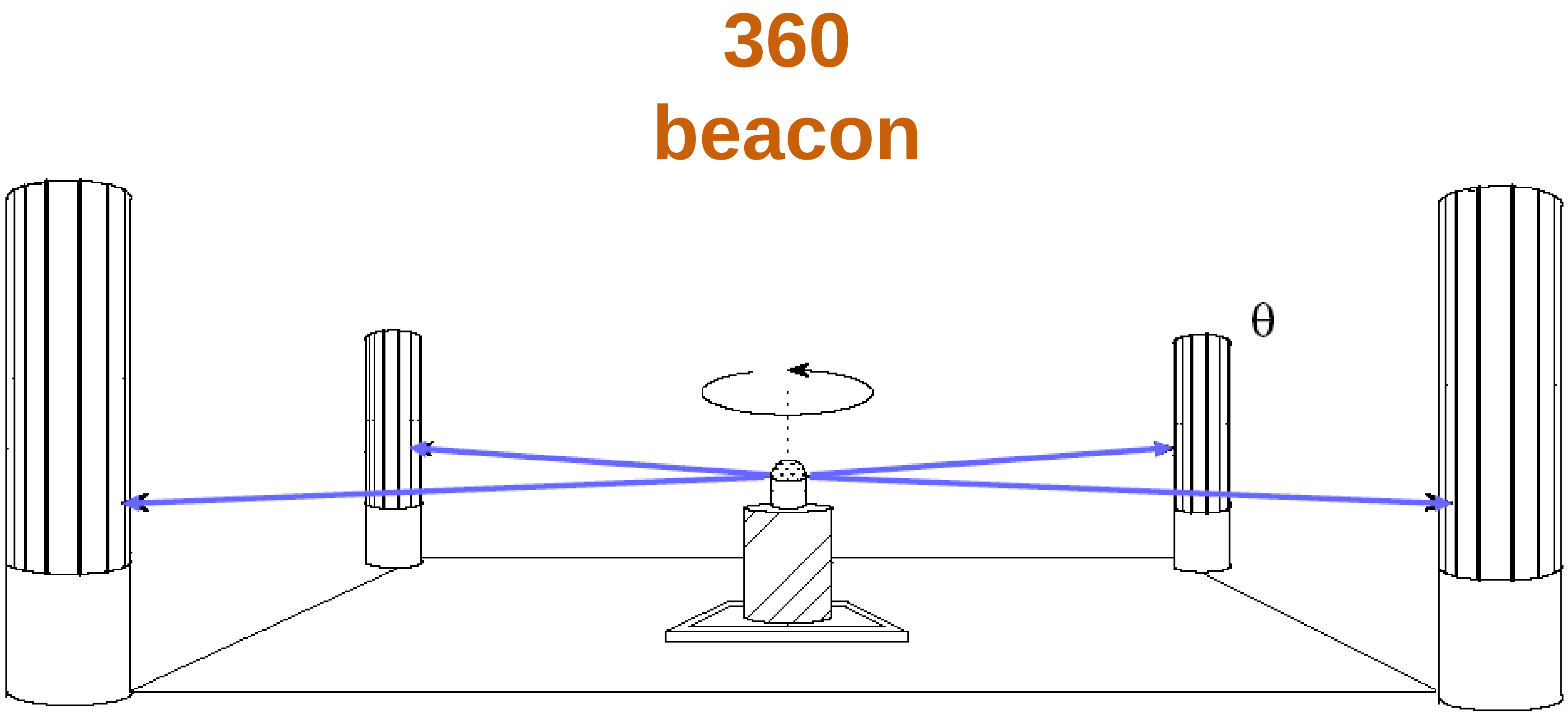
Positioning Beacon Systems: Triangulation



Pros?

Cons?

Positioning Beacon Systems: Triangulation



Autonomous Map Building

- Starting from an **arbitrary initial point**,
- a mobile robot should be able to **autonomously explore** the environment with its onboard sensors,
- **gain knowledge** about it,
- **interpret the scene**,
- **build an appropriate map**
- and **localize itself** relative to this map

SLAM

The Simultaneous Localization and Mapping Problem

End of Module 2