# Mobile Robotics

## Introduction

Vitor Santos, Nuno Lau
Universidade de Aveiro - 2024

# Introduction

Concepts and definitions

# What is a mobile robot?

Physical device capable of autonomous locomotion and decision-making capacity based on the perception of the environment that surrounds it.

The definition is so vast that the number of applications and examples in the contemporary world is virtually unlimited.

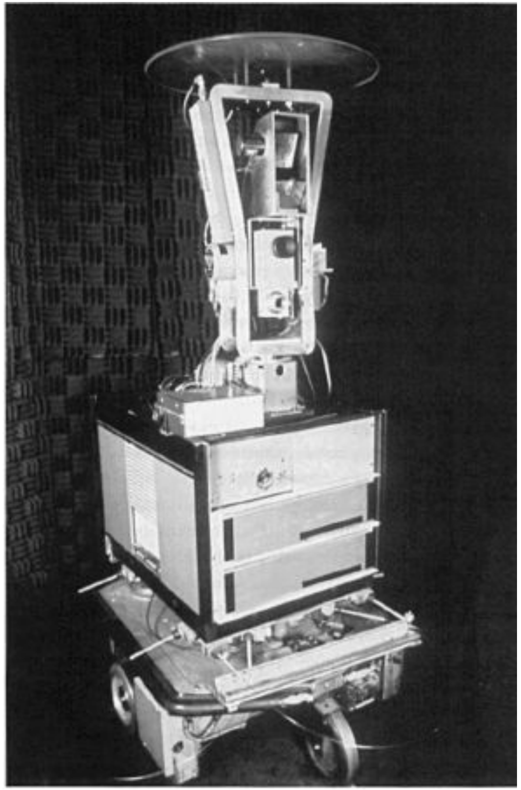
It is recommended to, for example, watch Durrant-White's Ted Talk in 2012:

What is a robot? [https://youtu.be/jElpEwMBNsQ](https://youtu.be/jElpEwMBNsQ)


# Introduction

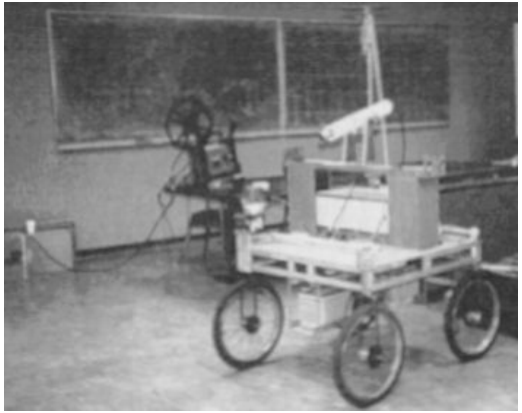
- Some distinctions of a mobile robot
    - Workspace: unlimited.
    - Autonomy: energy, of actions and decisions.
- Categories (in mobility)
    - Mobile robot
    - Automatically Guided Vehicle (AGV)
- Commercial applications
    - Transportation, surveillance, inspection, cleaning, ...
- Mobile how? Locomotion systems:
    - Wheels
        - Driving, steering, Both, Free (castors), Omnidirectional
    - Legs
    - Tracks (wheels on...)
- Other means of mobility (with specific challenges and complexities)
    - Air, water (above and below sea level)

# The beginnings

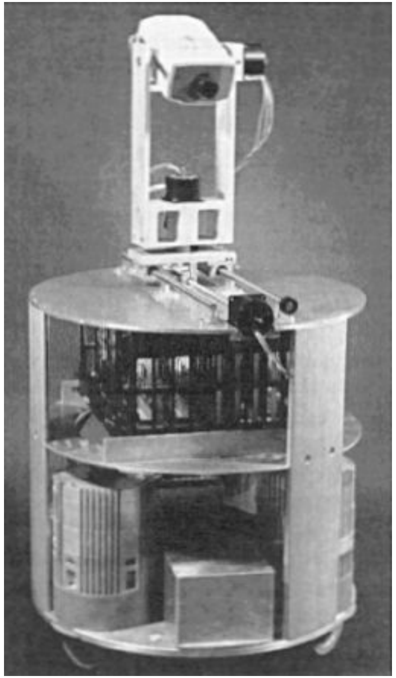


Shakey, Stanford (1966-1972)
- The first mobile robot with "intelligence".

Stanford Cart (1973-1979)



CMU Rover (1980)



# Areas involved in mobile robotics

- Mechanics
  - Structure, mechanisms, actuators, building materials,
- Electronics
  - Actuators, sensors and their connections and conditioning,
- Control
  - Processes and laws of motion generation in actuators
- Signal processing
  - Voice recognition, measurement interpretation, ...
- Image Processing (Vision)
  - Recognition of image properties
- Programming and Algorithms (computing)
  - Decision and decision management system, models, ...
- Psychology (synthetic)
  - Behavior models, ...

# Navigation: the heart of mobile robotics

“Navigation is the science (or art) of directing the course of a mobile robot as it traverses its environment (land, sea or air)”

*Philip McKerrow, 1993*

More specifically, navigation is a whole set of components (methodologies, actions, algorithms, etc.) that can include map building, path planning, localization, obstacle avoidance, leaving and recovering trajectories, etc., and which generally aims to take a robot to a certain point without getting lost and without colliding with anything.

# General problem

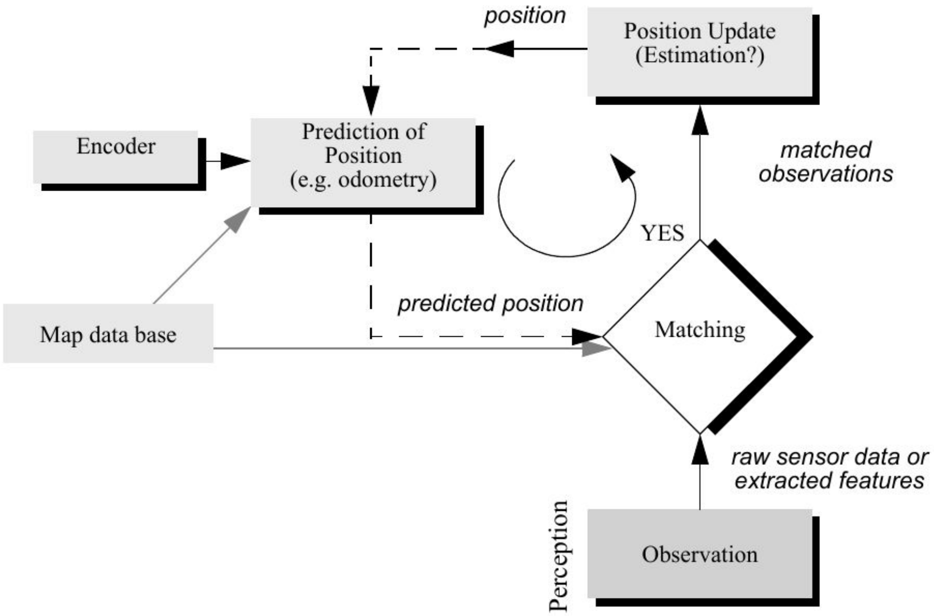
- Navigation - Durrant-White's 3 problems:
- Where am I? Where am I going? How do I get there?
    - Where am I? - Localization
    - Where do I want to go? - Mapping
    - How can I get there? - Planning (Global and Local)

# Specific problem

- Specific problem of mobile robots
  - When compared to manipulators
    - The need to calculate where it is in space as it is not always directly accessible...
    - Explicit need to obtain dynamic information from the environment
- Problems in the perception and mobility
  - Noisy and limited sensors
  - Limited actuators (poor control)
  - Robot state only partially known
  - Dynamic environment

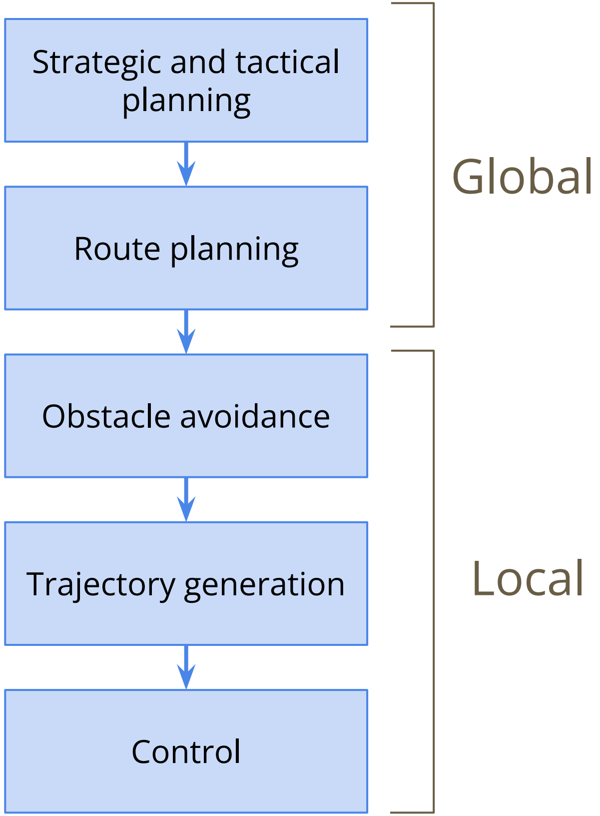
# The localization

- Localization can be a demanding challenge with varying accuracy
- It is a complex of estimation, observation and correspondence on maps.
- It uses relative localization methods (incremental), combined with external information (absolute) and mathematical estimation models in the face of noisy observations.



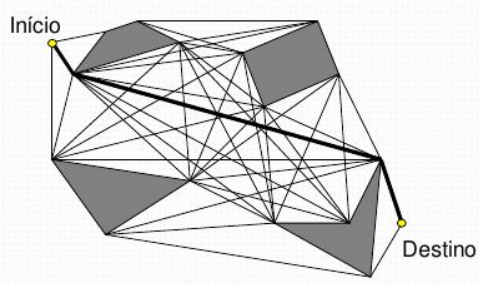
# Planning levels

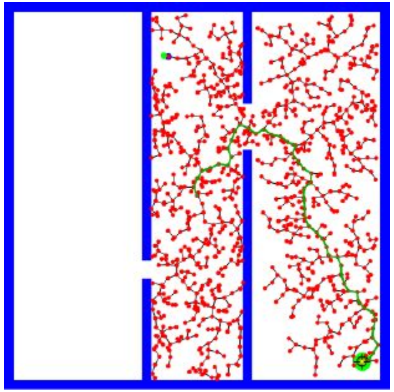
- Strategic and tactical planning
  - e.g. Visiting several places in a given region, for example in a given sequence.
- Route planning
  - e.g. Define which paths to take for each particular region, therefore, with a starting point and an ending point. Much is assumed at this level: paths are not blocked or limited by traffic, etc.
- Obstacle avoidance
  - Avoiding other vehicles, overtaking them, keeping distances, going around obstacles or blockages, etc.
- Trajectory generation
  - Definition of the instantaneous paths and the moments in which they must be taken (velocity, acceleration), respecting issues of safety and comfort for people and transported goods.
- Control
  - Instructions and laws for actuators to try to respect the planned trajectory by making the necessary corrections

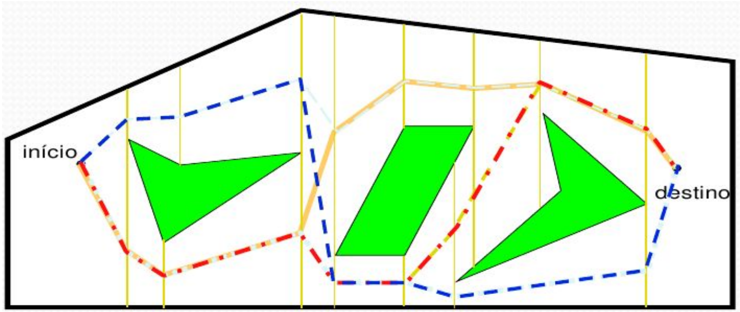


# Path planning (Global)

- Set of techniques for defining and choosing paths from a map.
- We need to have a final destination to calculate solutions.
- The solutions can be somehow optimized because there is a priori knowledge of the different possibilities, for example to avoid local minima.
- It includes several families of methods:
    - Cell decomposition
    - Road map
    - Potential fields
    - Sampling Methods

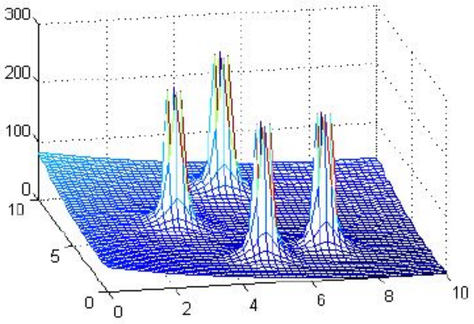


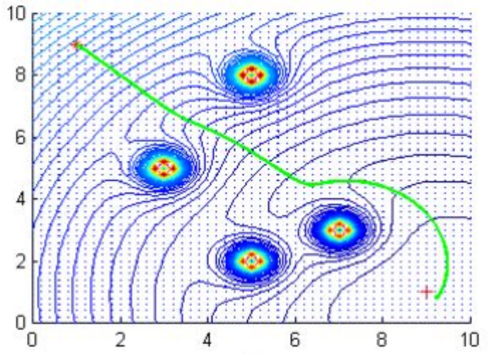



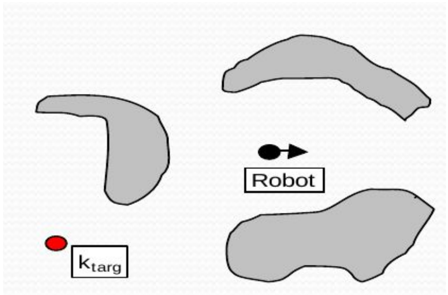


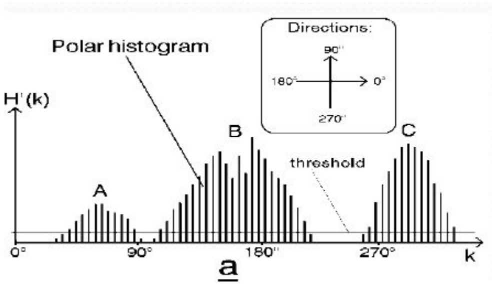
# Local planning and obstacle avoidance

- Local planning
  - The final destination is generally not needed or used in the calculation, only nearby or intermediate destinations are always updating.
  - Some solutions run the risk of having local minima
  - Requires sensors to assist motion and obstacle avoidance
  - Requires a control law for continuous path corrections
- Techniques based on Potential Fields
- Vector Field Histogram



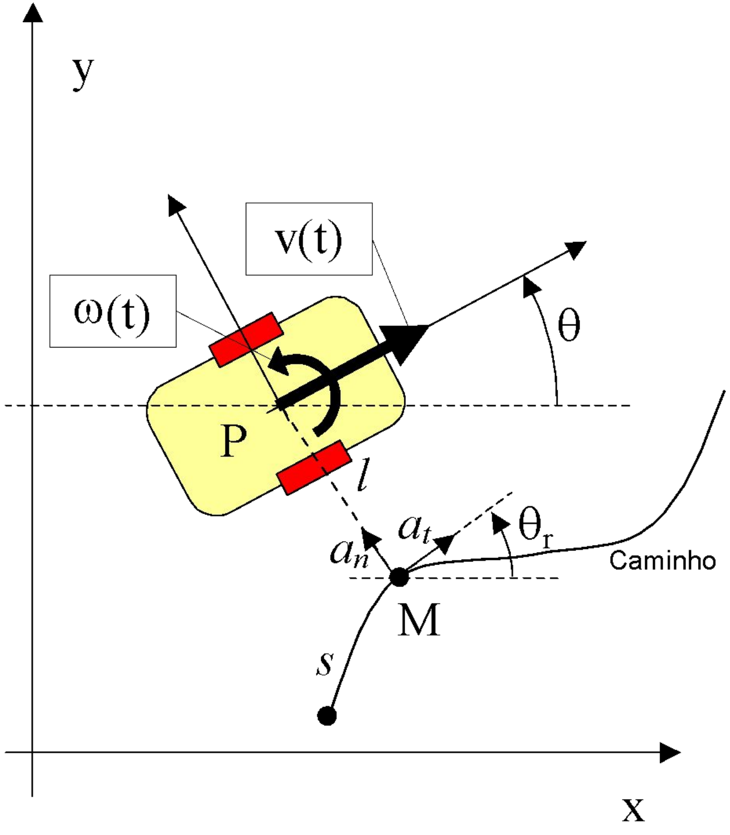




# Trajectory control

- Step that seeks to fulfill the planned motion.
- Three main problems define the issue of mobile robot control:
- Posture and trajectory tracking
  - Attempt to follow a real or pre-calculated reference to maintain posture and kinematic parameters (velocities) – for continuous moving systems.
- Path following
  - Try to minimize position and orientation errors in relation to a predefined path
- Posture stabilization
  - Problem of asymptotically reaching a given posture starting from any initial posture; it is assumed that the posture to be obtained is static



# Control architectures

# Definitions of control architectures

- An architecture provides a principled way of organizing a control system. However, in addition to providing structure, it imposes constraints on the way the control problem can be solved.

*Mataric, 1992*

- An architecture is a description of how a system is constructed from basic components and how those components fit together to form the whole.

*Albus, 1995*

- Robotic architecture usually refers to software, rather than hardware.

*Arkin, 1998*

- How the job of generating actions from percepts is organized.

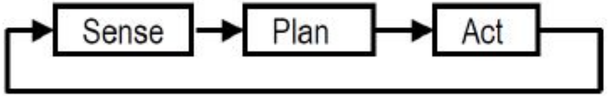
*Russell and Norvig, 2002*

# Architecture Classes

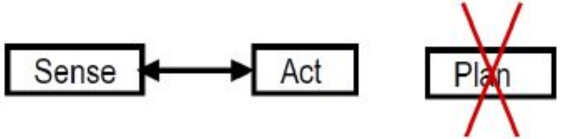
- 1- Deliberative (planner-based)
  - Sense-plan-act (SPA)
- 2- Reactive
  - Movement is an immediate and direct function of perception
- 3- Behaviour-based
  - Existence of independent “behaviors” where each dictates, in its own way, how the robot should act.
  - A system of arbitration should decide when there are contradictory actions (to move and where, for example)
- 4- Hybrid
  - Mixed characteristics of deliberative and reactive

# Typical organization of architectures

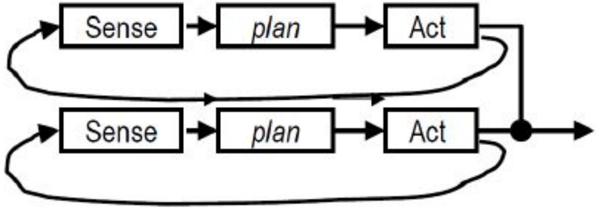
- Deliberative (or hierarchical)



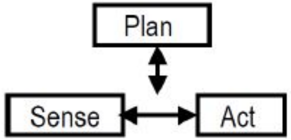

- Reactive




- Based on behaviors




- Hybrid




*RMI, 2019*

# Option for an Architecture

- It must support parallelism
  - simultaneous tasks
- Be able to be installed on hardware
  - Sensors, actuators, microprocessors available
- Have flexibility during the execution (run-time)
  - Adaptation during execution (change of processes/parameters).
  - It is important for cases with learning
- Modularity
  - Encapsulated control, multiple levels, software reuse
- Robustness
  - Failure of individual components, fault tolerance of multiple controllers
- Ease of use
- Performance (post-assessment)
  - How the robot behaves with the architecture. Prone to failure.

# Deliberative Architectures

- 1- Look ahead
- 2- Think/plan
- 3- Acts

- Planning: based on searching in a universe of potential solutions
- Sequential classical steps
  - Sensing, planning, acting (SPA)
- Disadvantages
  - Slow planning process because it's exhausting!
    - Many situations to go through
    - Plenty of memory to store situations/solutions
  - Stimulates open loop control

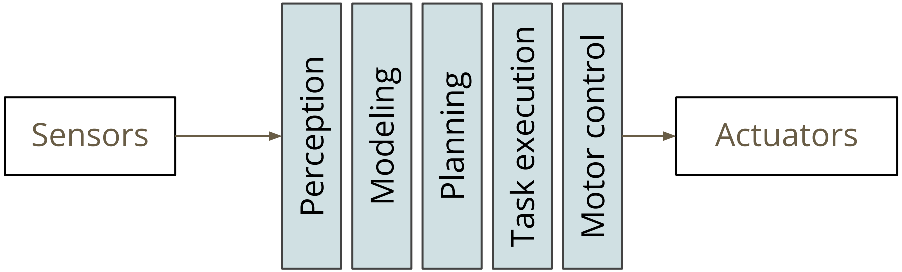
# Reactive Architectures

- 1-No look-ahead
- 2-React

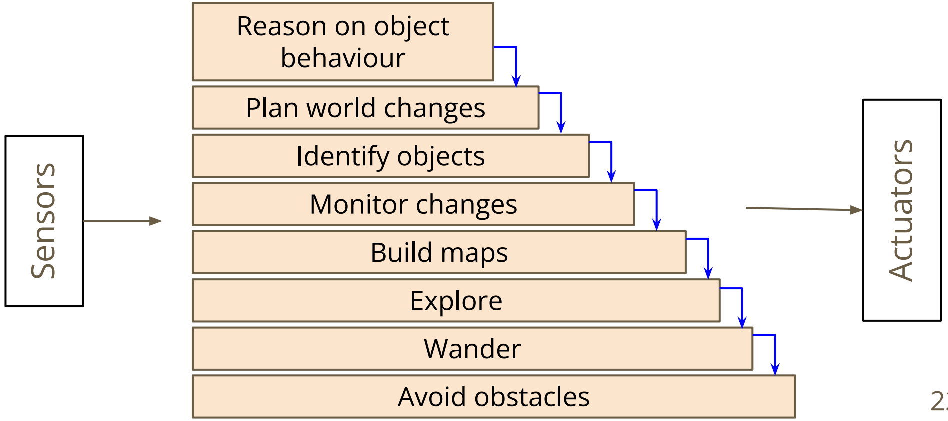
- Choice of reactions depending on the sensors
  - Function of perception uniquely
  - Function of arbitration, how is it done?
    - fixed hierarchy
    - dynamic hierarchy
    - Learning
- R. Brooks' Subsumption variant
  - Level 0
  - Upper levels that encompass the functionality of the levels below and inhibit them

# Deliberative and Reactive

- Classic deliberative approach




- Reactive approach – subsumption architecture
  - Hierarchical layer inhibition



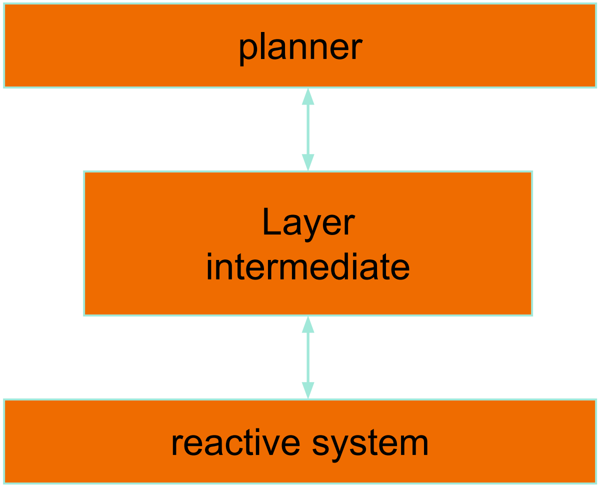
# Deliberative vs Reactive

- Neither approach is "best" for all robots; each has its strengths and weaknesses
- Control requires some unavoidable trade offs because:
  - Thinking is slow
  - The reaction must be fast
  - Thinking allows looking ahead (planning) to avoid bad actions
  - Thinking too much can be dangerous (e.g. falling off a cliff)
  - To think, the robot needs (lots of) accurate information
  - The world keeps changing while the robot is thinking, so the slower it thinks, the more imprecise its solutions.
- As a result of these exchanges, some robots do not think at all, while others think and act very little.
- It all depends on the robot's task and its environment!

*RMI, 2019*

# Hybrid Architectures

- Combination of Deliberative and Reactive architecture
  - “Think” Slowly
  - “React” quickly
- Known as “three-layer architectures”
- Middle layer role
  - Compensate limitations of other layers
  - Reconcile the different time scales
  - Reconcile possible decision contradictions between the layers



# Universal plans

- If every situation the robot faces…
  - … is known in advance, …
  - … it is possible to make an action plan for each one of them,
    - the robot is then said to have a universal plan
- Any system that uses the universal plan is reactive
  - but the reaction is not calculated in real time!
  - It is thoroughly planned before starting to execute
- Case of some AGVs
  - Knowledge of the list of all possible paths and stops, as well as forced stop situations due to a predetermined sensorial situation

# Behavior Based Architecture

- General motto:
  - "Think" in a distributed way during the action.
- Modularity issue:
  - It is the way in which architecture breaks down into its elements:
    - SPAs break down functionally:
      - Independent sequential actions (**S**ense the world, generate a **P**lan, transform plan into **A**ctions)
  - Reactive ones like Subsumption use task-oriented decomposition, parallel modules running concurrently (i.e. alternatively) for certain tasks:
    - obstacle avoidance or contouring, follows walls, ...
- In behavior-based architectures, behaviors are used as building blocks...

# The Behaviors...

- ...are control laws.
- They accomplish specific tasks/objectives
  - avoid others, look for a friend, go home...
- Typically run in parallel/concurrently
- Can store the state to build representations or models
- Directly connect sensors and actuators
- Take information from some behaviors and send it to other behaviors
- Higher level than action (go-home instead of turning 35.5º!)
- Typically a closed-loop subsystem but with longer cycles than low-level ones (such as PIDs)

# Example of a behavior-based architecture

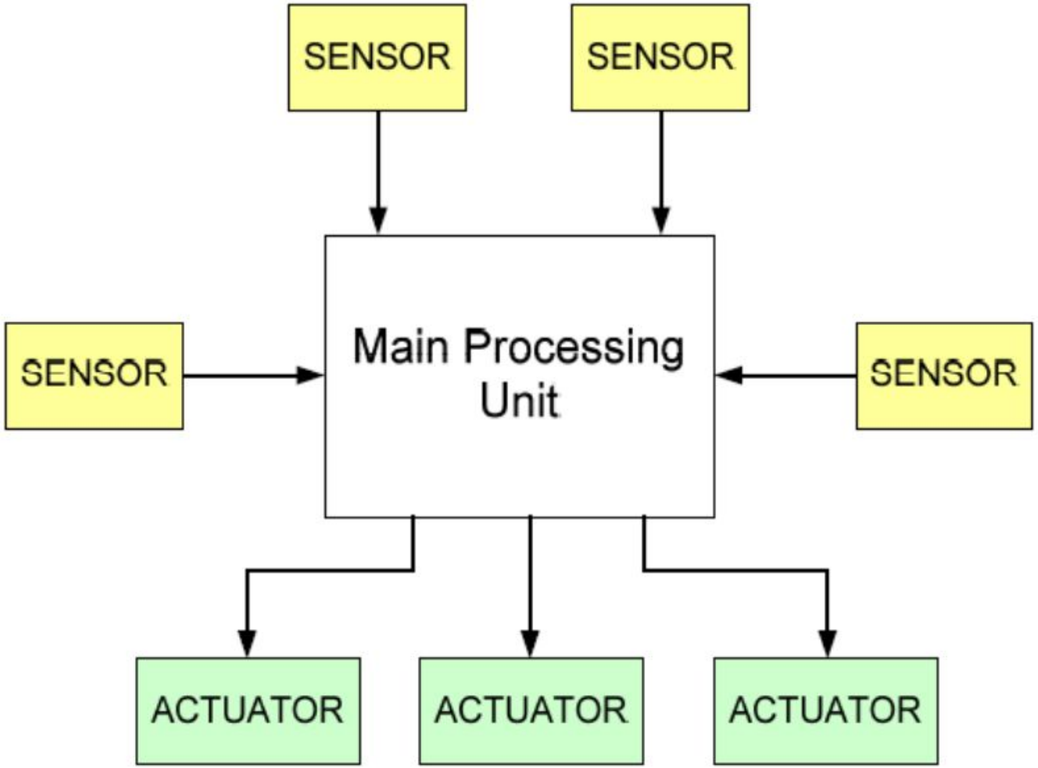
- A robot with sonar and compass
- Navigation layer
  - Behavior1: movement without collision
  - Behavior2: Stay close to walls or objects
- Landmark detection layer
  - Behavior1: landmark detection
    - If it keeps moving straight for a certain time, then it concludes that there is a wall
- Mapping layer and path search
  - Behaviors created to associate with each new landmark (e.g. north left wall...)
  - Two landmarks found in sequence are topologically linked
  - Use odometry to estimate its position (metric complements topology)

# Computational models

Software/hardware architectures to support control architectures

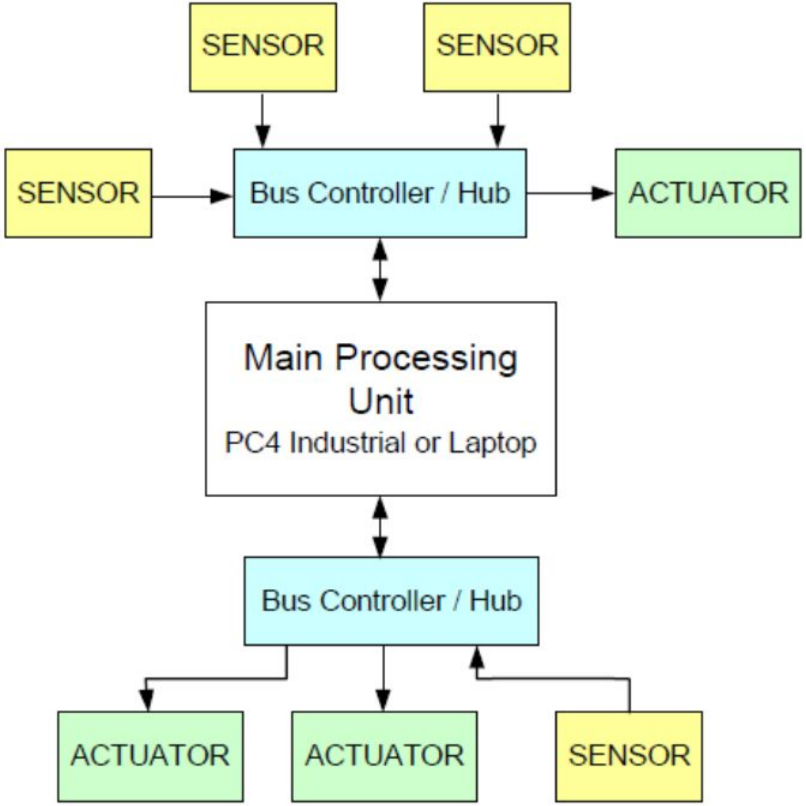
# Computational models in mobile robotics

Fully centralized architecture



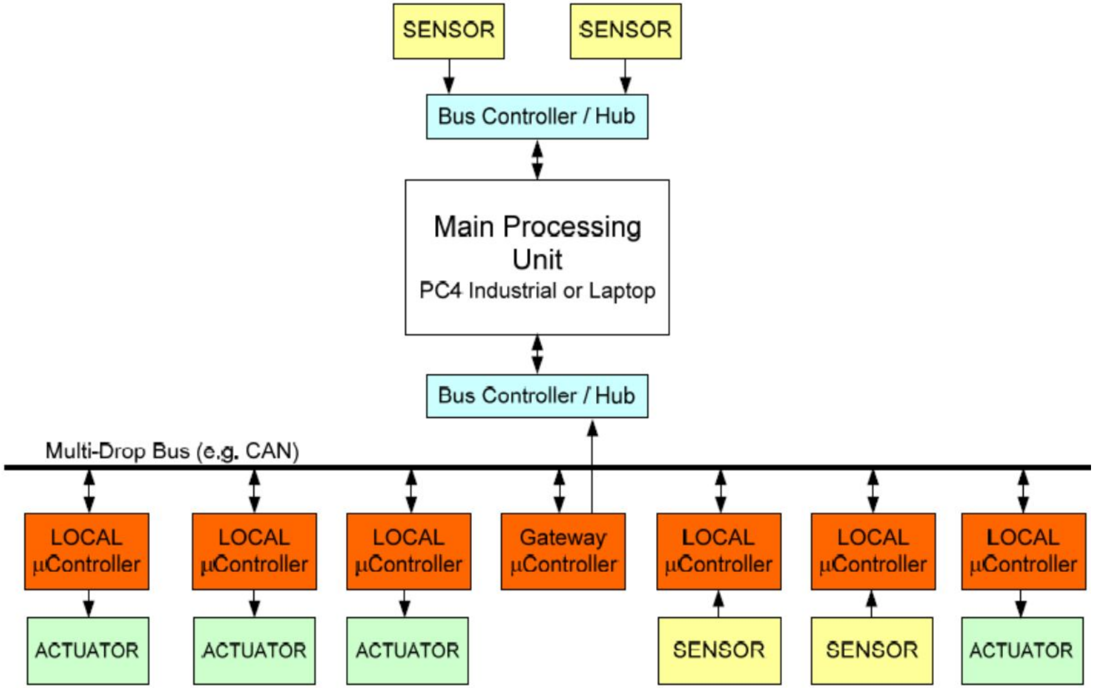
# Computational models in mobile robotics

Star configuration architecture



# Computational models in mobile robotics

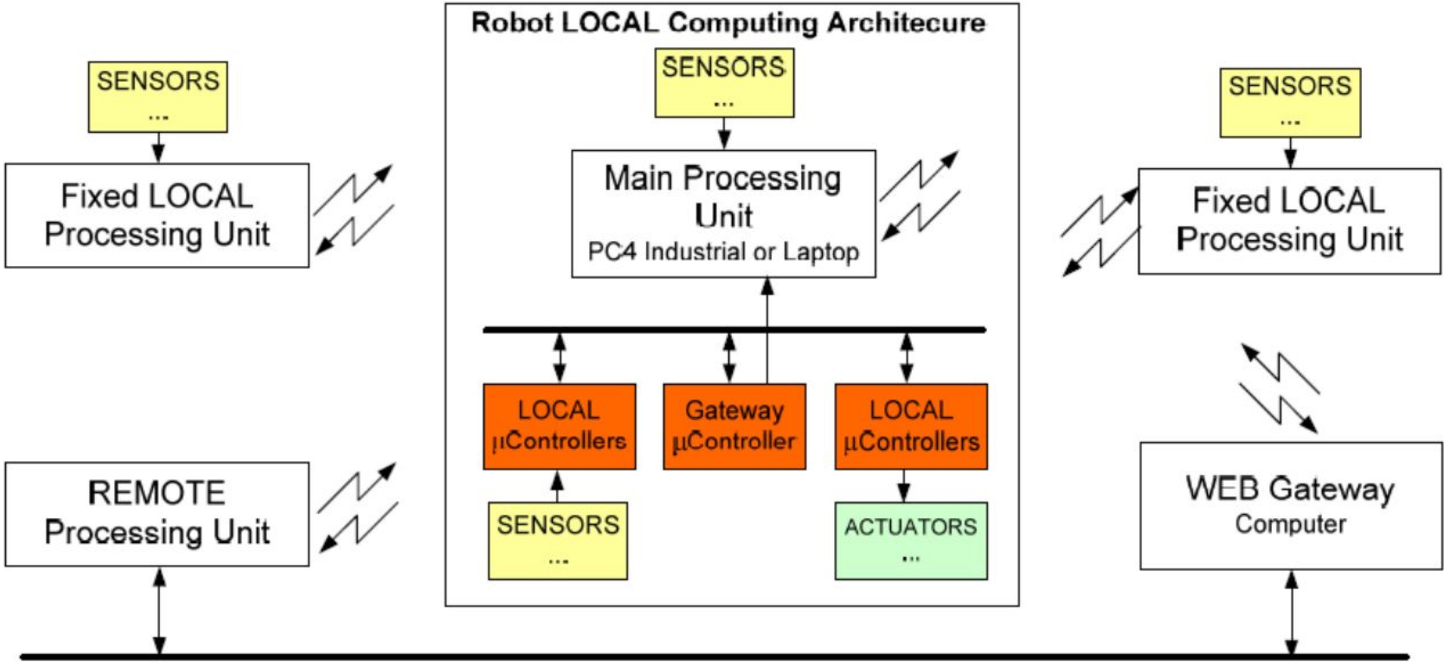
Distributed hierarchical architecture




*RMI, 2019*

# Computational models in mobile robotics

Fully distributed architecture



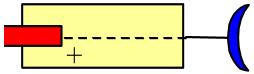
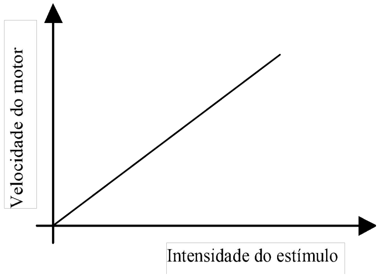

*Kivi, 2019*

# Braitenberg vehicles

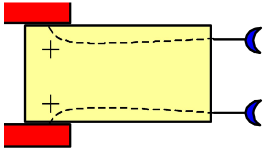
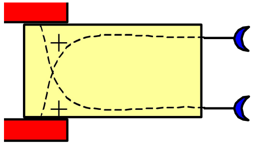
# Braitenberg vehicles

- They are perception ⇔ action models that...
- ... result in certain behaviors.
- They consist of...
  - Sensors
  - Actuators (typically two motors driving independent parallel wheels)
  - A controller that affects motors based on sensors and a given transfer function
- Assumptions in vehicles
  - Actuators have variable velocity
  - Sensors provide information continuously
  - There may be elementary processing of sensorial data before using it to “command” actuators in an excitatory or inhibitory manner.
- Reference:
  - Vehicles: Experiments in Synthetic Psychology, Valentino Braitenberg, MIT Press, 1984-1998.

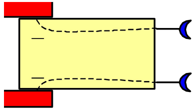
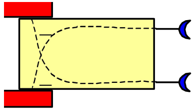
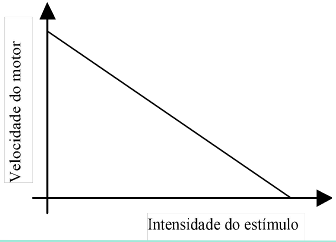
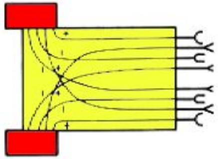
# Braitenberg's vehicle 1

| Keywords | Illustration | Main features | Comments |
|---|---|---|---|
| Gettting Around |  | 1 motor – 1 sensor<br>The speed is proportional to the intensity of the sensor measurement. In the example it is excitatory. To be inhibitory, the slope of the line would be negative!<br> | Simple movement. Only the (non-modeled) irregularities of the system can lead to a non-linear movement. |

# Braitenberg's vehicle 2

| Keywords | | Illustration | main features | Comments |
|---|---|---|---|---|
| Fear (Coward) | 2A |  | 2 motors + 2 sensors connected similarly to vehicle 1 (excitatory) | Walks towards the source (of light, for example) if it is in front of it, but moves away from it if it is not in front of it |
| Aggression | 2B |   | | Walk towards the source if it is straight ahead. Look for the direction of the source if it is not in front of it. |

# Braitenberg's vehicle 3

| Keywords | | Illustration | main features | Comments |
|---|---|---|---|---|
| Love | 3A |  | Similar to Vehicle 2 but with inhibitory connection to motors | Tendency to approach and stop at the source |
| | 3B |  |  | Unless the source is straight ahead, tendency to move away from it, but slowing down as it approaches it |
| | 3C |  | Several pairs of sensors (stimuli of different nature – light, sound, $O_2$ concentration, ...) with inhibitory and excitatory connections cross-linked | Illusion of the existence of knowledge due to the enormous variety of behaviors that the system can exhibit: fleeing from one element and attracted to another, ... |

# An example with vehicle 3C

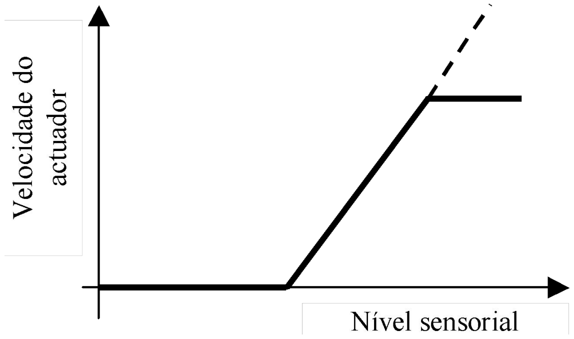
- It has sensors of 4 different natures
  - Light, temperature, oxygen, organic matter
- It has the following connections and control strategies:

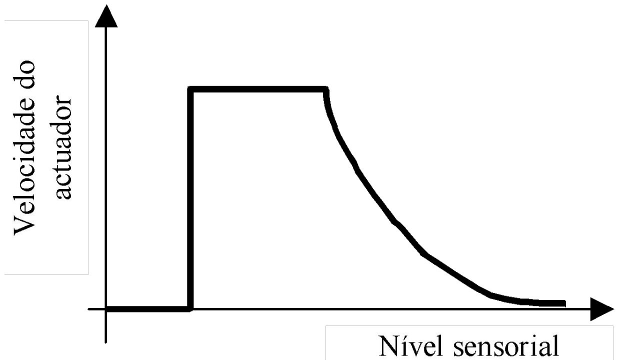
| Sensor Connection | **Light** | **Temperature** | **Oxygen** | **Organic matter** |
|---|---|---|---|---|
| **direct** | excitatory (+) | | Inhibitory (-) | |
| **cross** | | excitatory (+) | | Inhibitory (-) |

- Comments
  - Heads towards the light but moving away from it while heading towards higher temperatures
  - “Likes” the abundance of $O_2$ and stops at high concentrations, as well as for organic matter, but from the latter it can even deviate
  - There seem to be the concepts of “Values”, “Knowledge” and “Preferences”
  - Challenge: what is the arbitration rule between the various controllers...!

# Braitenberg's vehicle 4

- Introduction of non-monotonic control
  - not just excitatory or inhibitory
- Use of non-linear functions in the relationship between the sensor and the actuator
- The number of thresholds and the respective sensorial levels allow for a wide range of behaviors
- Keyword associated with vehicle 4:
  - Decision
- Examples of control functions:

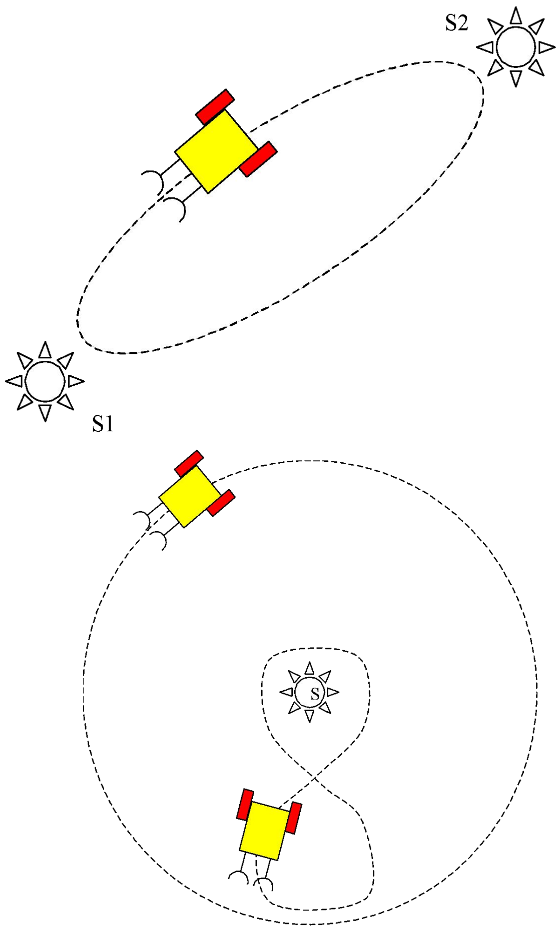



- Interpretation
  - Dead zones (no activity); fixed zones (unchanging activity); ...

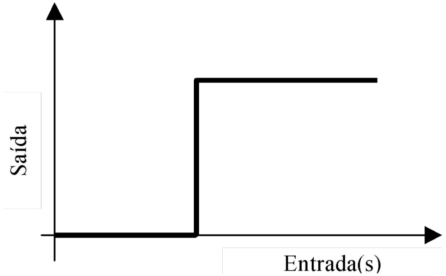
# Behavior of vehicles type 4

- Moving closer and further away from the sources as the stimulus becomes stronger or weaker.
- It can result in more complex behaviors like those illustrated: between two sources, or even with a single source.
- In case 1: the sensitivity to a source can be inverted when the signal increases “too much” or decreases “too much”!



# Braitenberg's vehicle 5

- Dynamic variation of vehicle 4 decision thresholds
- More complex control functions using step functions (binary action).
  - Possibility of implementing a binary system of any complexity: any logical system, even with memory.
  - Several inputs can be combined (added) and their output will be active only if the sum reaches a given threshold
- Keywords: logic, memory, names



# Braitenberg's vehicle 6

- Natural selection
- Construction of a large number of vehicles of the previous types with various sensors and control functions
- Put them in competition (for example see who gets to the light the fastest, etc...)
- Select the most capable according to a criteria to be defined.
- “Errors” in the construction of new vehicles simulate genetic mutations

# Other vehicles - summary

- Vehicle 7
  - The imaginary linker: “mnemotrix”
  - Learning
- Vehicle 8
  - Space, things and movement
  - One image sensor – lateral inhibition
- Vehicle 9
  - Shapes: symmetry and periodicity
- Vehicle 10
  - New knowledge. The deduction.
- Vehicle 11
  - The unidirectional imaginary linker: “ergotrix”. Causality.
- Vehicle 12
  - Automatic adjustment of decision thresholds
  - Look-ahead. Deliberation.
- Vehicle 13
  - Predicting and updating the internal models
- Vehicle 14
  - Bias in decisions – adjustable preferences