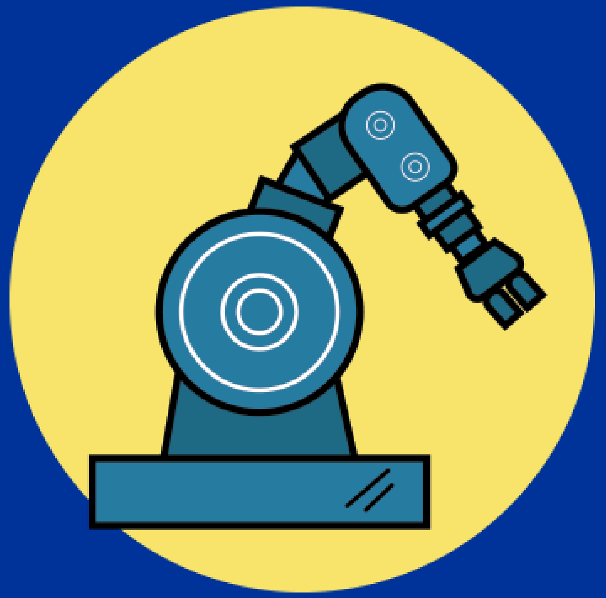


Introduction to Robotics

Manipulation and Programming

Unit 4: Motion Control

ROBOT CONTROL ARCHITECTURE

DR. ERIC CHOU

IEEE SENIOR MEMBER

Objectives

- What is a Robot Architecture?
- Categories of Robot Control
- Choice of Control Architecture

What is Robot Architecture?

SECTION 1

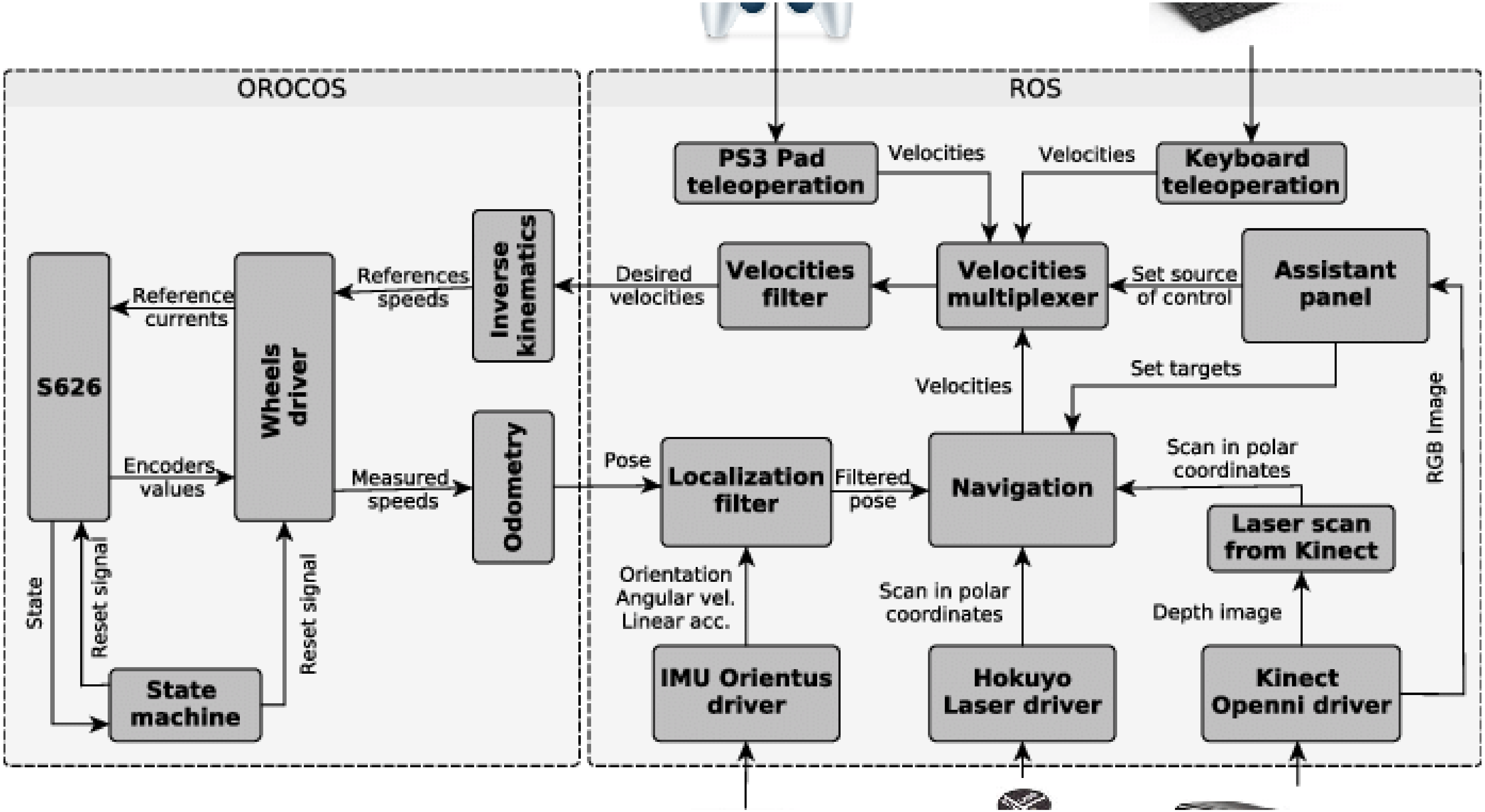
What is a Robot Architecture?

- There are many different ways in which a robot control program can be put together. In order to program a robot in a structured and principled fashion, we use an appropriate robot control architecture.
- System developers have typically relied upon robotic architectures to guide the construction of robotic devices and for providing computational services (e.g., communications, processing, etc.) to **subsystems** and **components**.

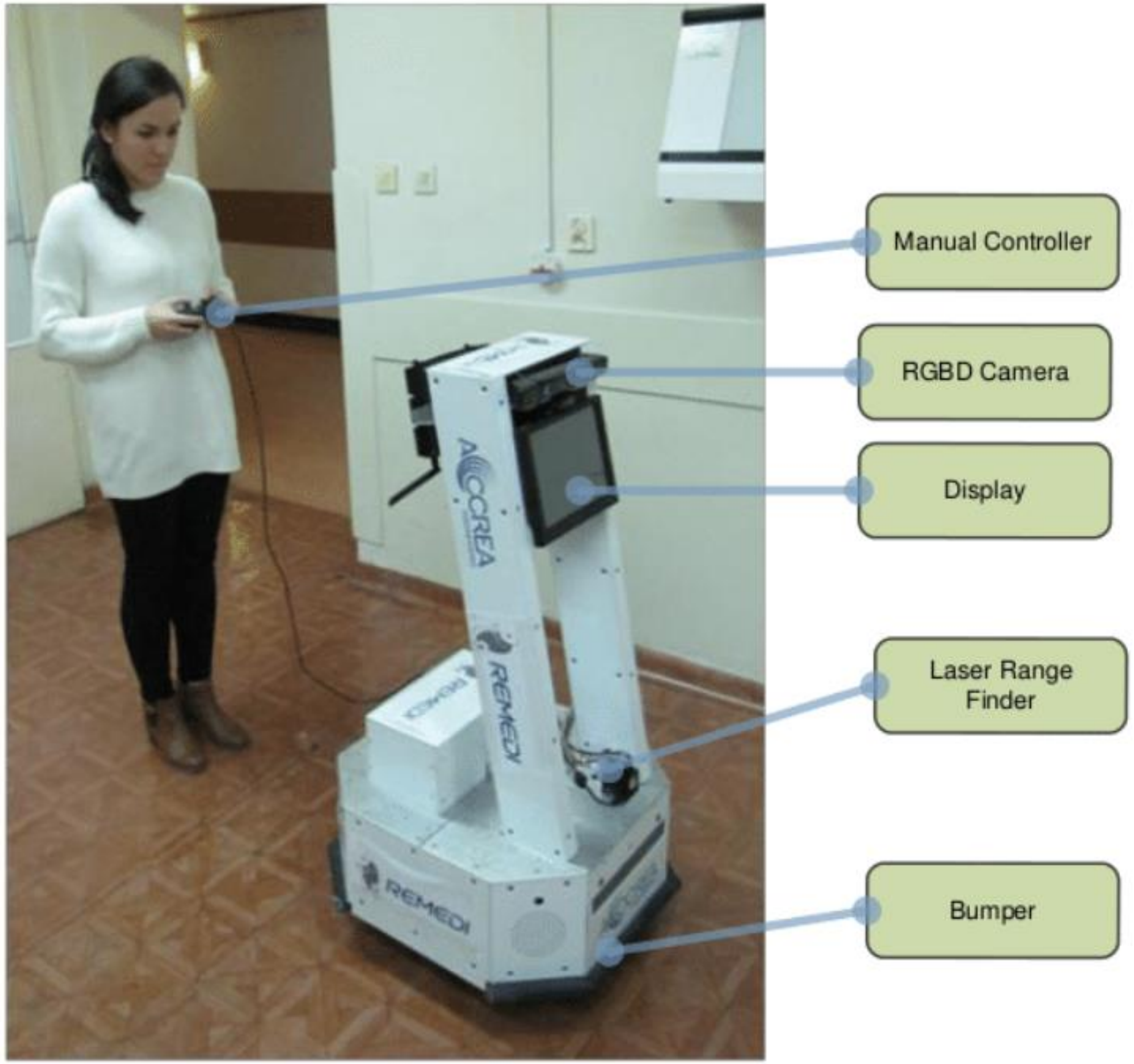
Robot Architecture

ASIC for Robots

Computer Architecture and Operation System (ROS) for Robots



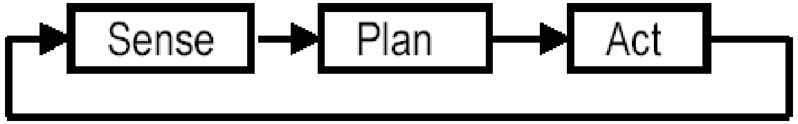
OROCOS
ROS
PS3 Pad teleoperation
Velocities
Velocities
Keyboard teleoperation
Inverse kinematics
References speeds
Reference currents
S626
Wheels driver
Desired velocities
Velocities filter
Velocities multiplexer
Set source of control
Assistant panel
Set targets
Velocities
RGB Image
Encoders values
Measured speeds
Odometry
Pose
Localization filter
Filtered pose
Navigation
Scan in polar coordinates
Laser scan from Kinect
State
Reset signal
Reset signal
State machine
Orientation
Angular vel.
Linear acc.
Scan in polar coordinates
Depth image
IMU Orientus driver
Hokuyo Laser driver
Kinect Openni driver


Manual Controller
RGBD Camera
Display
Laser Range Finder
Bumper
ACCREA
REMEDI
REMEDI

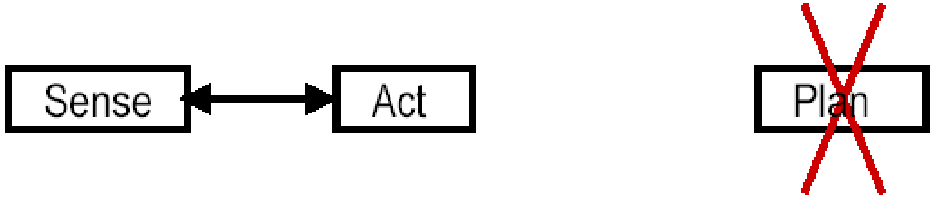
Categories of Robot Control

SECTION 2

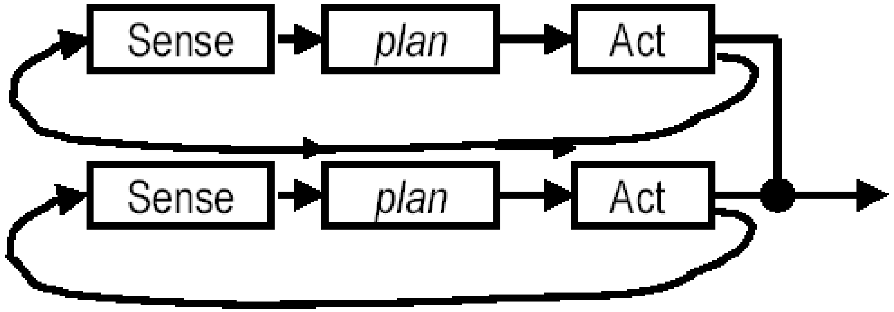
1. Hierarchical:



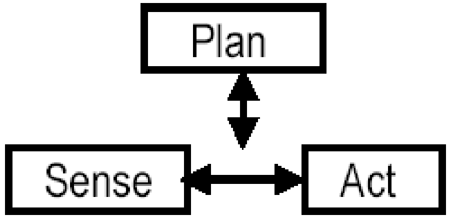
2. Reactive:



2+. Behavior-Based:



3. Hybrid deliberative/reactive:

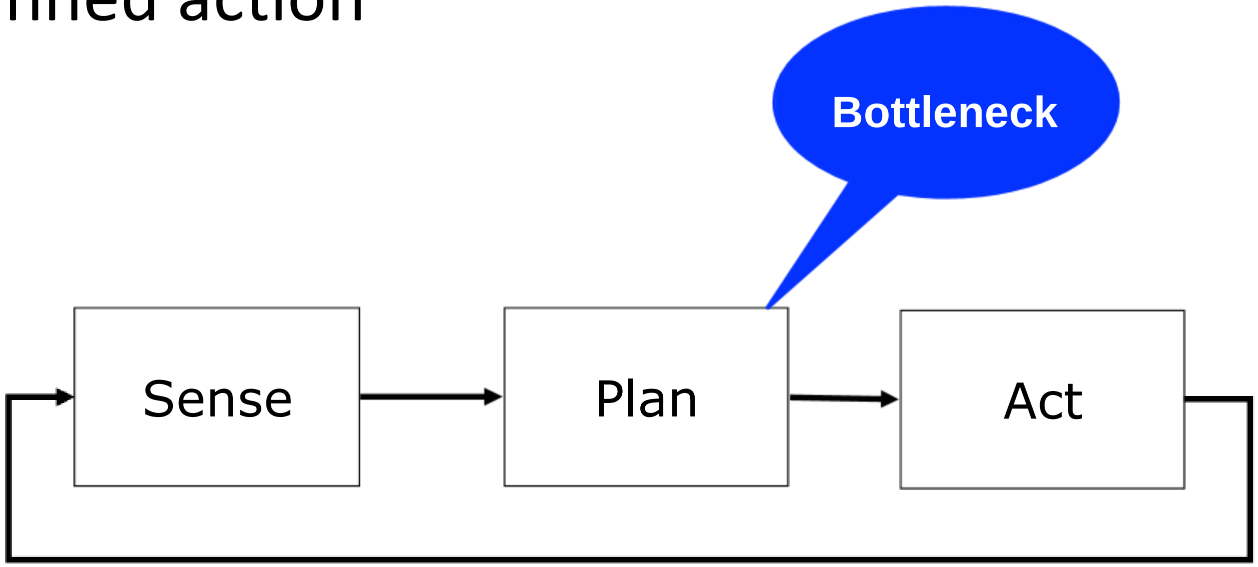


Categories of Control

- **Deliberative Control** : Think hard, act later.
  - SPA, serial, complete each step first – then proceed
- **Reactive Control** : Don’t think, (re)act.
  - Direct connection between perception to action, no memory, no planning.
- **Hybrid Control** : Think and act independently, in parallel.
  - Deliberative and Reactive modules run independently at different time scales
- **Behavior-Based Control** : Think the way you act.
  - Distributed by behavioral task decomposition
  - Each behavior has its restricted planning and execution capabilities

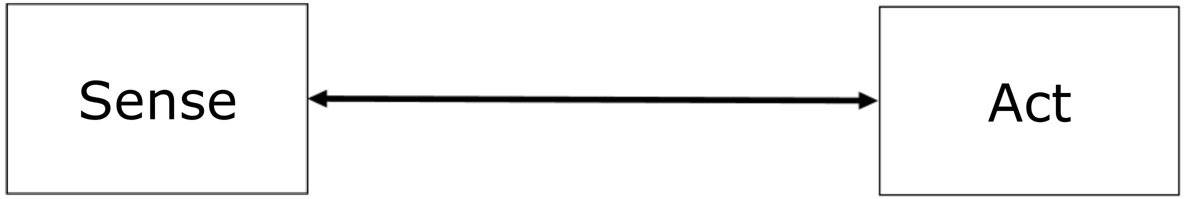
Deliberative Paradigms

- Sense the world
- Update world mode and generate plan
- Knowledge representation + Automated reasoning
- Execute planned action

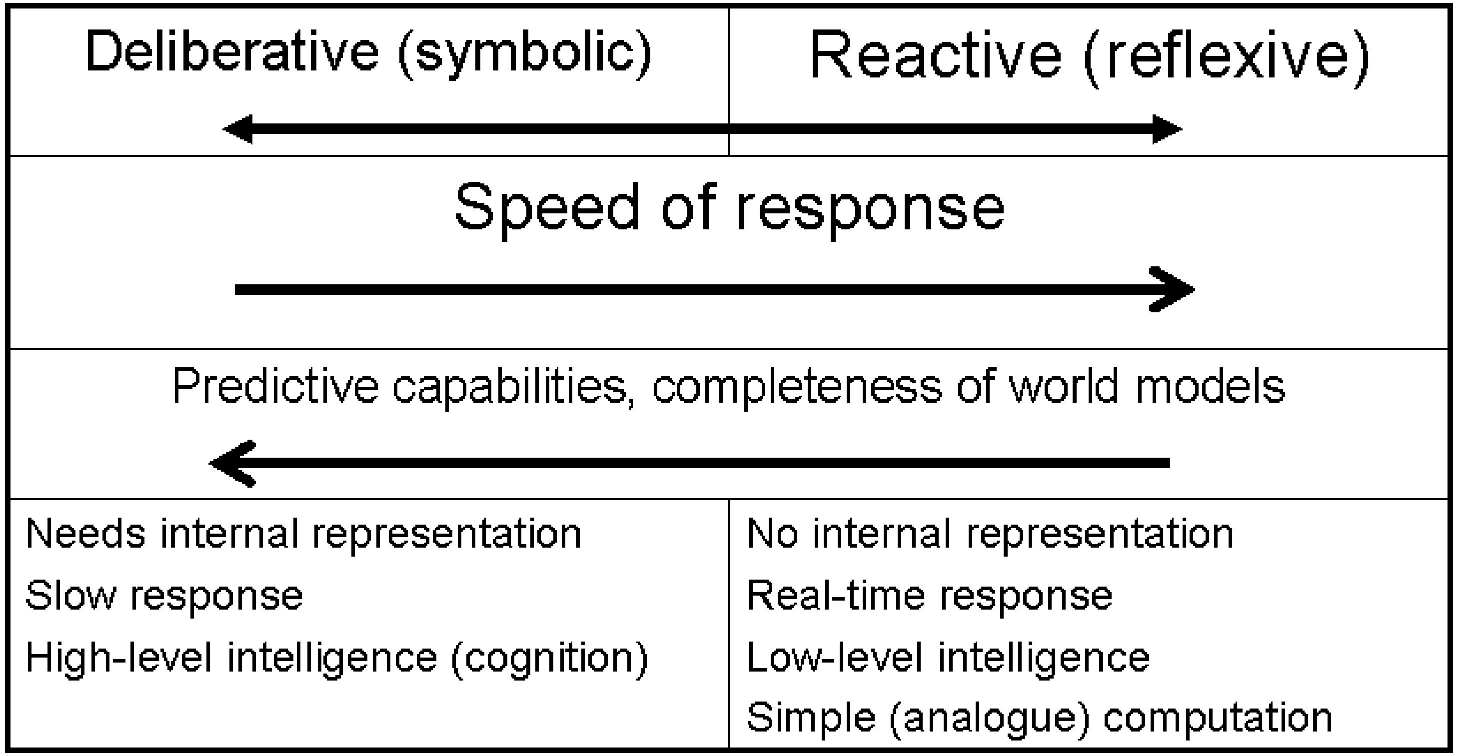


Reactive Paradigm

- No world model; no planning
- Direct mapping between sensor input and actuators output
- Very reactive to changes in sensor readings
- (Radical) answer to shortcomings of deliberative paradigm



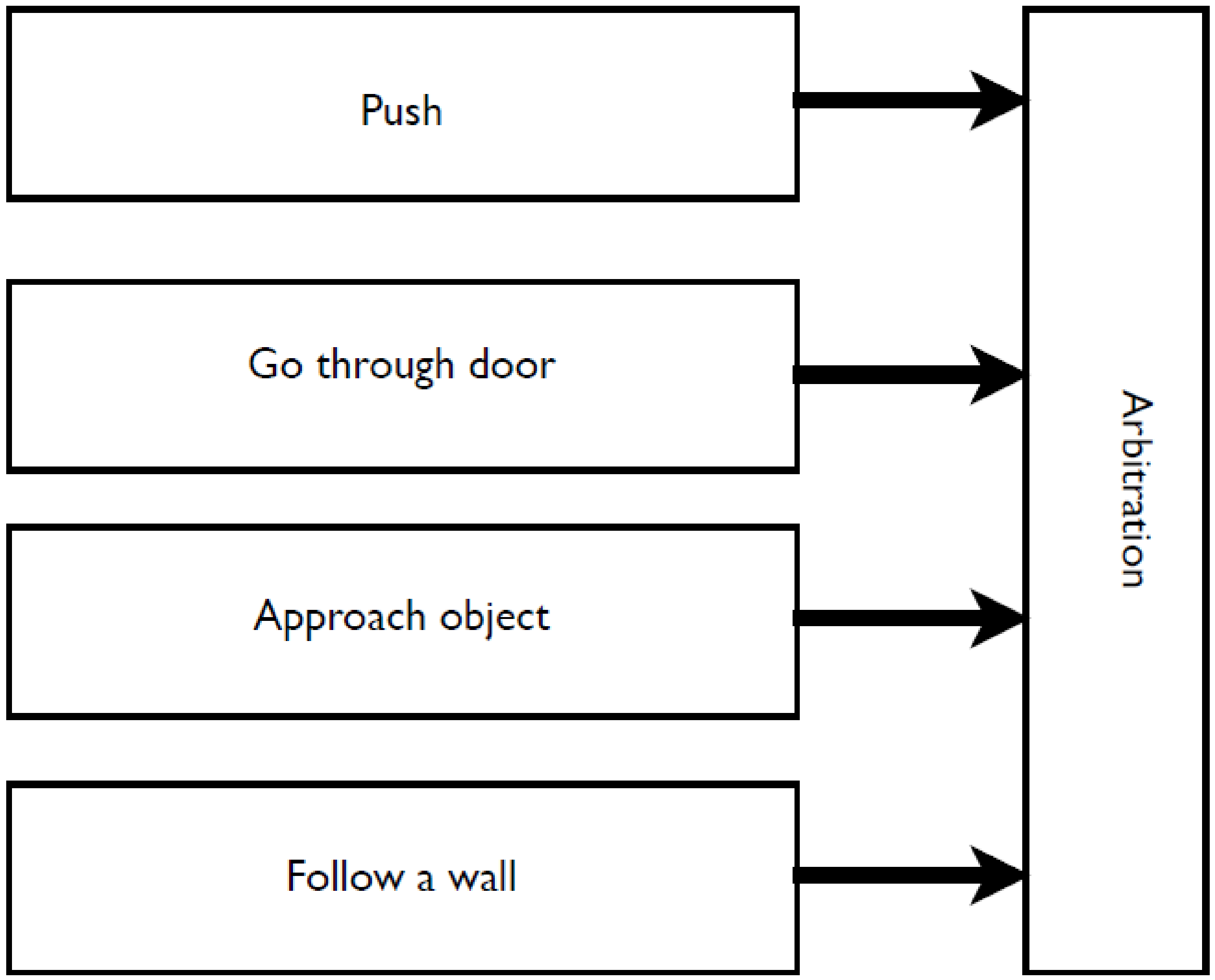
Comparison



Deliberative (symbolic)
Reactive (reflexive)
Speed of response
Predictive capabilities, completeness of world models
Needs internal representation
Slow response
High-level intelligence (cognition)
No internal representation
Real-time response
Low-level intelligence
Simple (analogue) computation

Behavior-Based Architectures

- Overall controller composed of two parts
  - Task achieving controllers (e.g., simple state machines)
  - Arbitrating controller (also task specific)
- Controllers:
  - Are reactive (map perception to action)
  - Do not include world models, no planning
  - Are only capable of performing one task each
    - Should perform it very efficiently

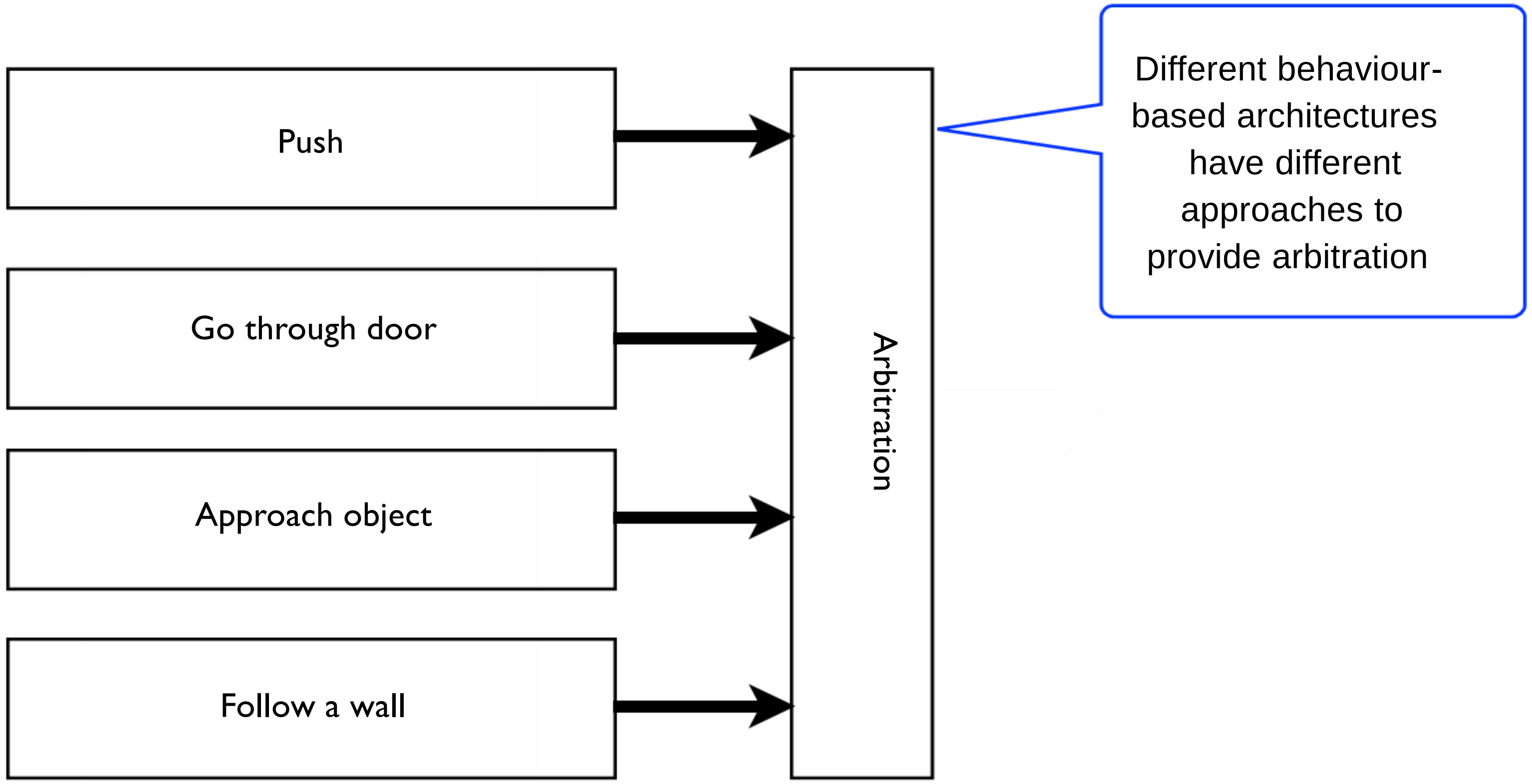


Behavior-Based Architectures

Behavior-Based Architectures

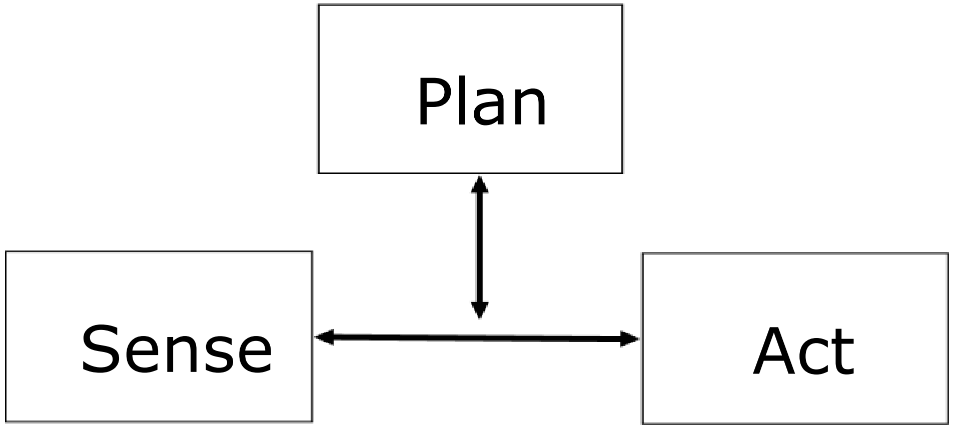
- Behaviors are the basic building block for robotics actions, with an overall emergent behavior obtained from their arbitration
- Behaviors support good software design principles due to modularity
- However, we have some assumptions:
  - We need to be able to decompose the task into appropriate “primitive” behaviors
  - Each “primitive” behavior requires only simple perceptual and physical talents

Behavior-based Architecture



Hybrid Paradigm

- World model, used for planning
- Closed-loop, reactive control
- Many architectures of today follow this paradigm



Choice of Robot Control Architecture

SECTION 3

Choice of Robot Control Architecture

- In many cases, it is impossible to tell, just by observing a robot's behavior, what control architecture it is using. Only for simple robots, it is often the case.
- However, when it comes to more complex robots, i.e., robots that have to deal with complex environments and complex tasks, the control architecture becomes very important.

Choice of Robot Control Architecture

- The different properties of an environment that will impact the robot's controller (and therefore the choice of control architecture):
  - noisy,
  - speed/response time of sensors and effectors
  - total/partial hidden state/ observable
  - discrete v. continuous state ; static v. dynamic ...
- Similarly, the properties of the robot's task impact the choice of the control architecture. The task requirements can constrain the architecture choice.

Hardware Types

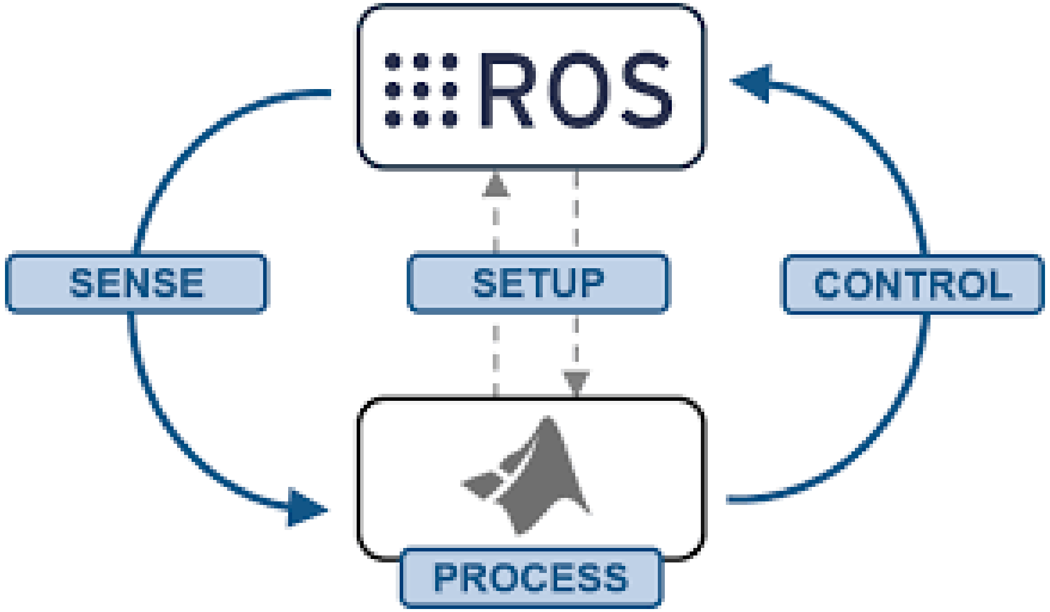
- Reactive Filters
- ASIC-based (FSM)
- DSP/Micro-processor-based
- Deep-learning Neural network

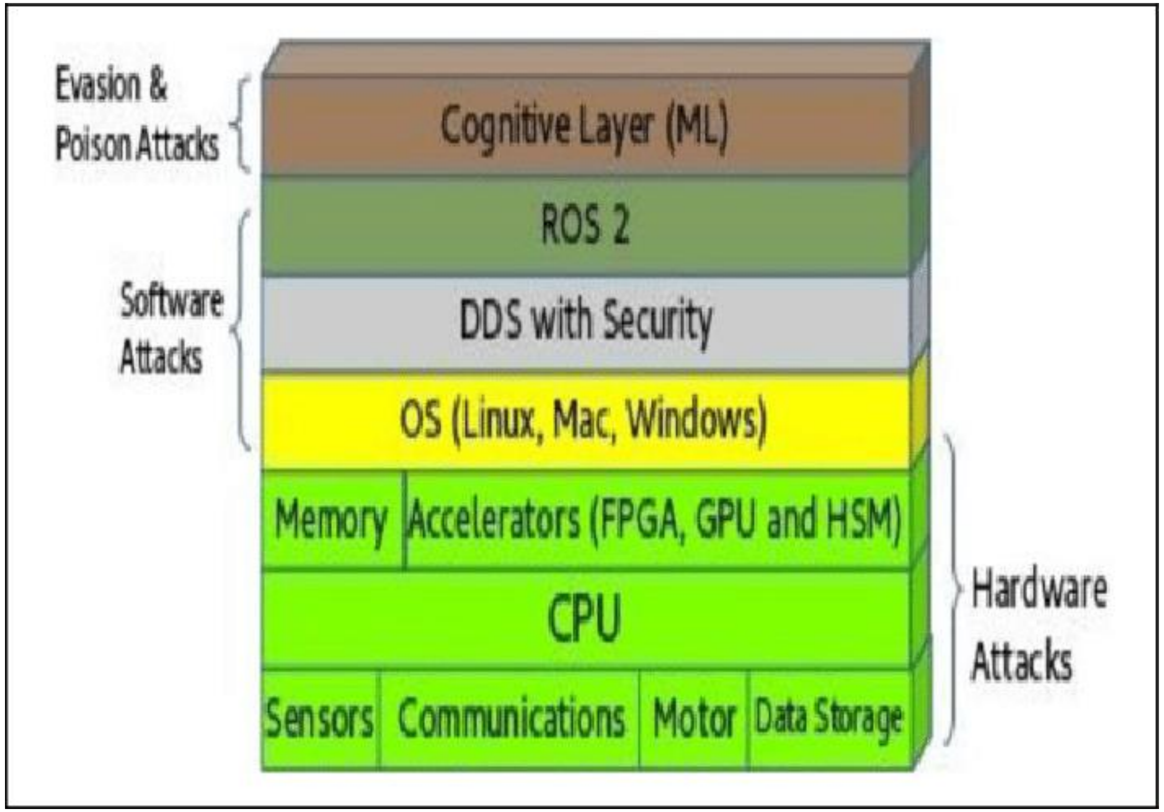
System Types

- Parallel Paradigm vs. Serial Paradigm
- Asynchronous (Event-driven) vs. Synchronous

Operation System

- ROS (Robot Operation System)





Some Criteria for Selecting a Control Architecture

- **support for parallelism**: the ability of the architecture to execute parallel processes/behaviors at the same time.
- **hardware targetability**:
  - how well the architecture can be mapped onto real-robot sensors and effectors.
  - how well the computation can be mapped onto real processing elements (microprocessors).
- **run-time flexibility:** does the architecture allow run-time adjustment and reconfiguration? It is important for adaptation/learning.
- **modularity:** how does the architecture address encapsulation of control, how does it treat abstraction?
  - Does it allow many levels, going from feedback loops to primitives to agents?
  - Does it allow re-use of software?

Some Criteria for Selecting a Control Architecture

- **niche targetability:** how well the architecture allows the robot to deal with its environment
- **robustness:** how well does the architecture perform if individual components fail? How well does it enable and facilitate writing controllers capable of *fault tolerance*?
- **ease of use:** how easy to use and accessible is the architecture? Are there programming tools and expertise?
- **performance:** how well does the robot perform using the architecture? Does it act in real-time? Does it get the job done? Is it failure-prone?

Criteria

- Response Time
- Environmental Model
- Internal States

Response Time Scale

Response Time-scale is an important way of distinguishing control architectures.

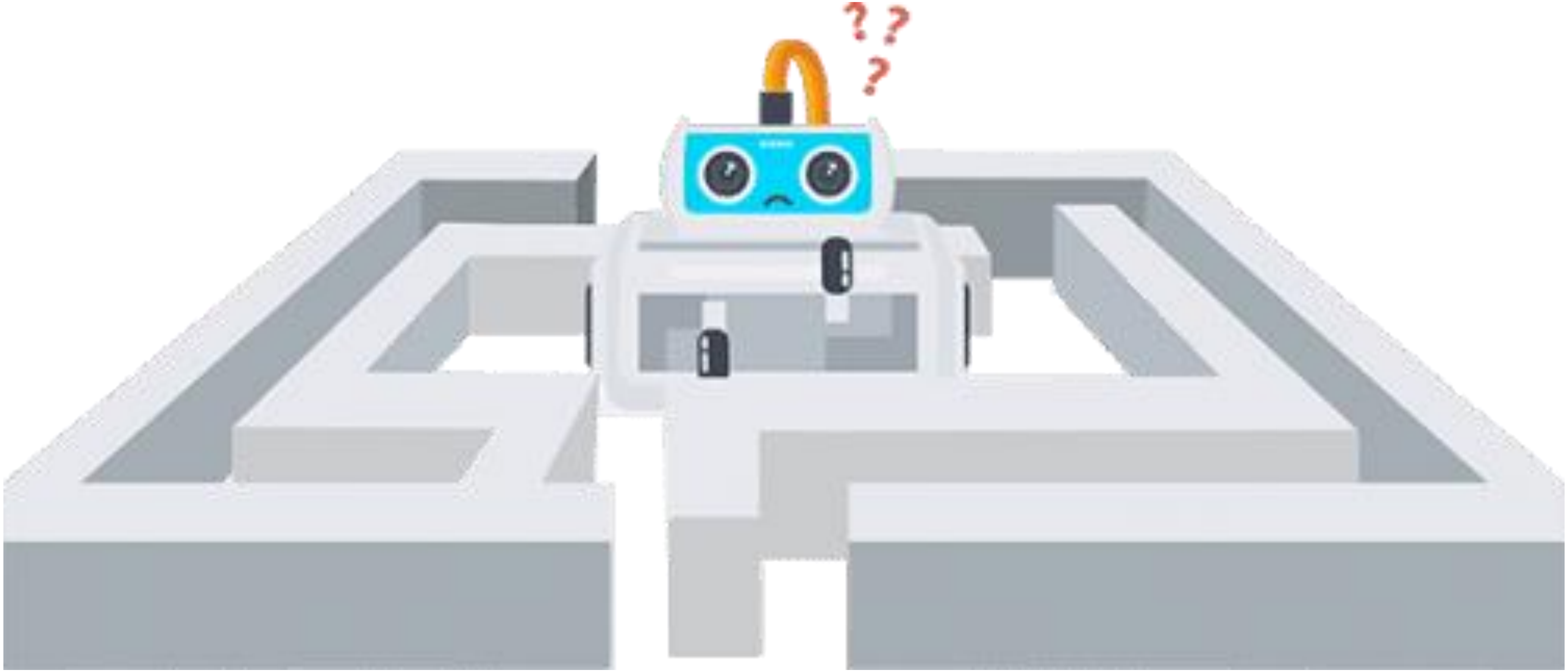
- **Reactive systems** respond to the real-time requirements of the environment,
- while **deliberative system** look ahead (plan) and thus work on a longer response time-scale.
- **Hybrid systems** must combine the two time-scales in an effective way, usually requiring a middle layer; consequently they are often called ***three-layer architectures***.
- Finally, **behavior-based systems** attempt to bring the different response time-scales closer together by distributing slower computation over concurrent behavior modules.

Environmental Models

- Another key distinguishing feature between architectures is **representation** of the world/environment, also called **environmental modeling.**
- Some tasks and architectures involve storing information about the environment **internally**, in the form of an **internal representation** of the environment.

Modeling

- For example, while exploring a maze, a robot may want to remember a sequence of moves it has made (e.g., "left, left, right, straight, right, left"), so it can back-track and find its way.
    - Thus, the robot is constructing a representation of its path through the maze.
    - The robot can also build a *map* of the maze, by drawing it using exact lengths of corridors and distances between walls, etc. .
    - This is also a representation of its environment, a model of the world.
- If two robots are working together, and one is much slower than the other, if the fast robot remembers/learns that the other is always slower, that is also a type of a model of the world, in this case, a model of the other robot.



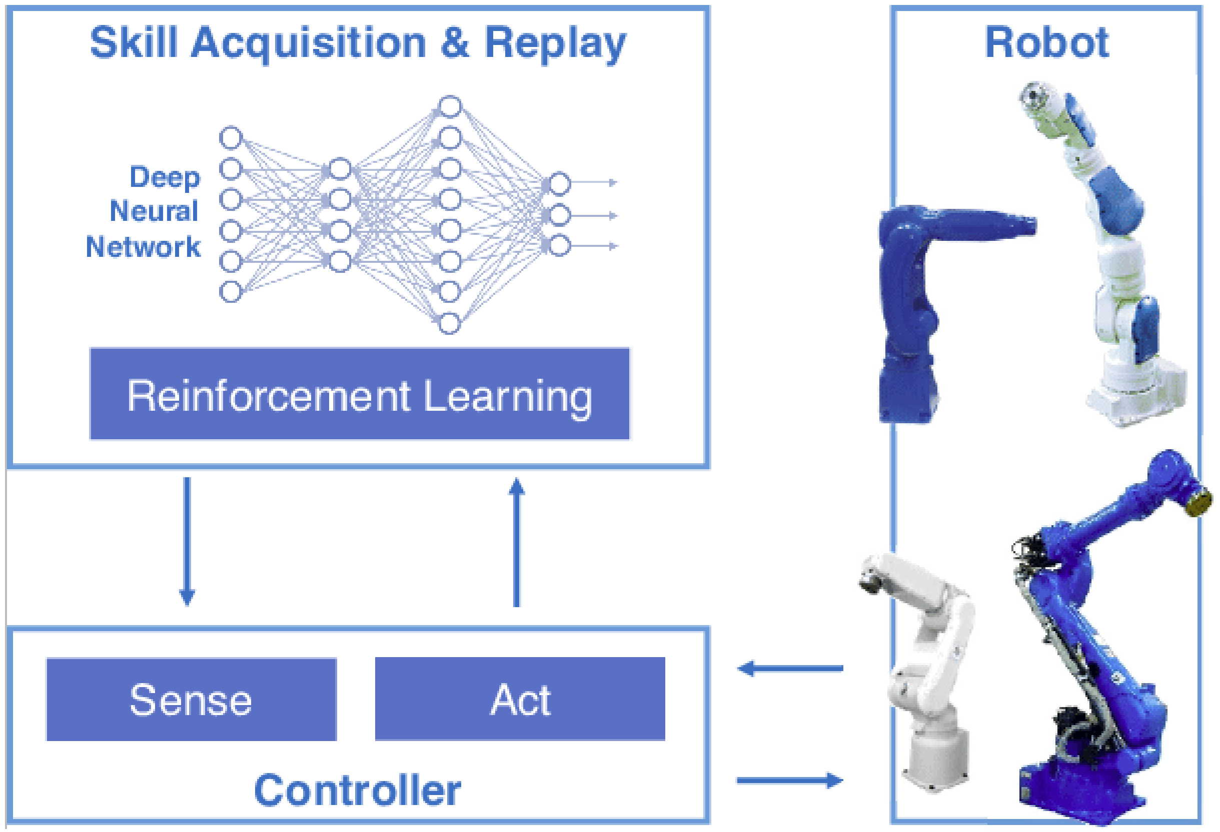
Different Environment Models.

There are numerous aspects of the world that a robot can represent/model, and numerous ways in which it can do it, including:

• **Spatial**	metric or topological: maps, navigable spaces, structures
• **Objects**	instances of detectable things in the world
• **Actions**	outcomes of specific actions on the self and environment
• **self/ego**	stored proprioception: sensing internal state, self- limitations, etc.
• **Intentional**	goals, intended actions, plans
• **Symbolic**	abstract encoding of state/information

Robots

SECTION 5


Skill Acquisition & Replay
Deep
Neural
Network
Reinforcement Learning
Sense
Act
Controller
Robot