INTELLIGENT ROBOTS

CHAPTER 1: MOBILE ROBOT, CONTROL AND DECISION ARCHITECTURE

Outlines

- Introduction to Mobile Robot
- Control & Decision Paradigms
- Control & Decision Architecture

Introduction to Mobile Robot

- Definition of Mobile Robot
- Turtlebot2 Structure
- Turtlebot2 Base
- Turtlebot2 Operating System
- Turtlebot2 Simulator： Gazebo

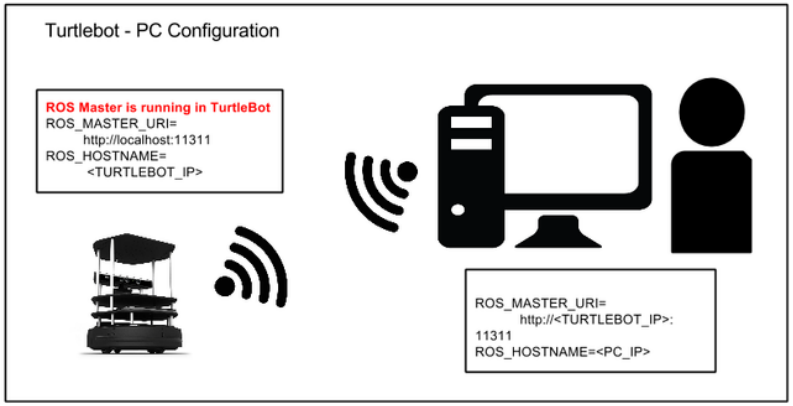
Definition of Mobile Robot

- **A mobile robot is an automatic machine that is capable of locomotion.**

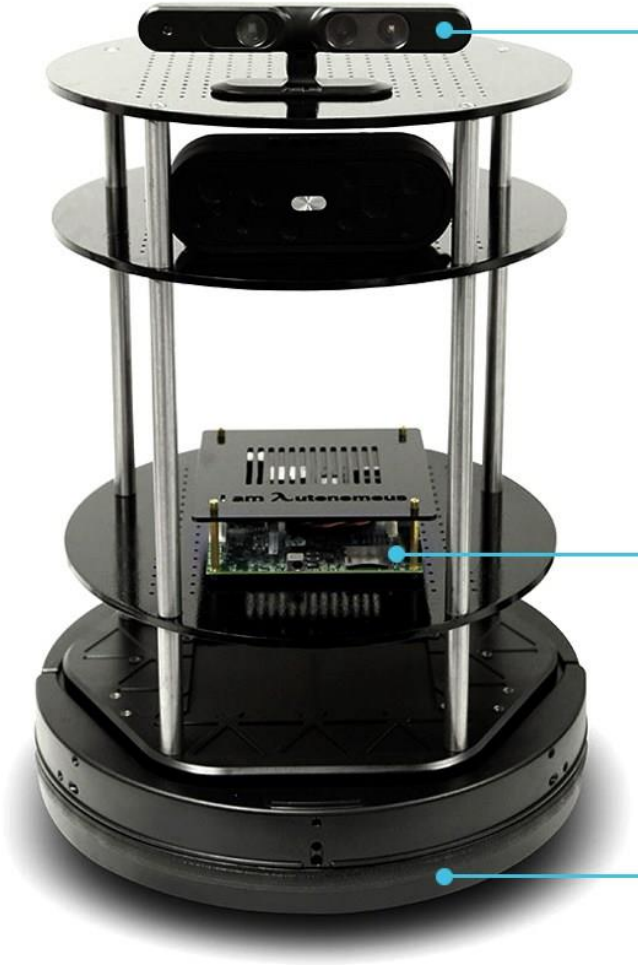
Turtlebot2

Links of learning materials:
https://www.ncnynl.com/archives/201609/786.html

Link of Turtlebot package on ROS website:
http://wiki.ros.org/Robots/TurtleBot



Turtlebot2 Structure

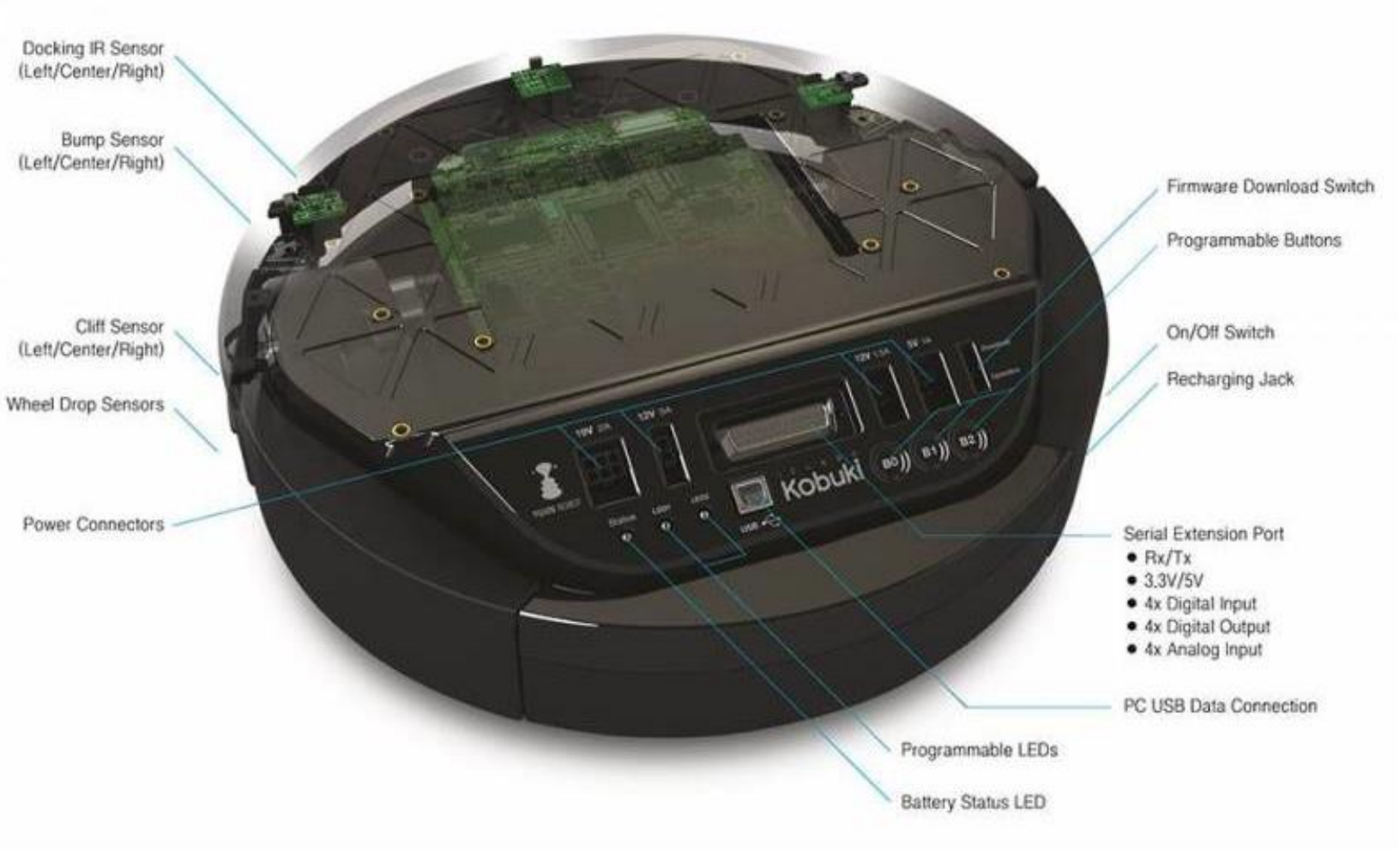


3D Depth Camera

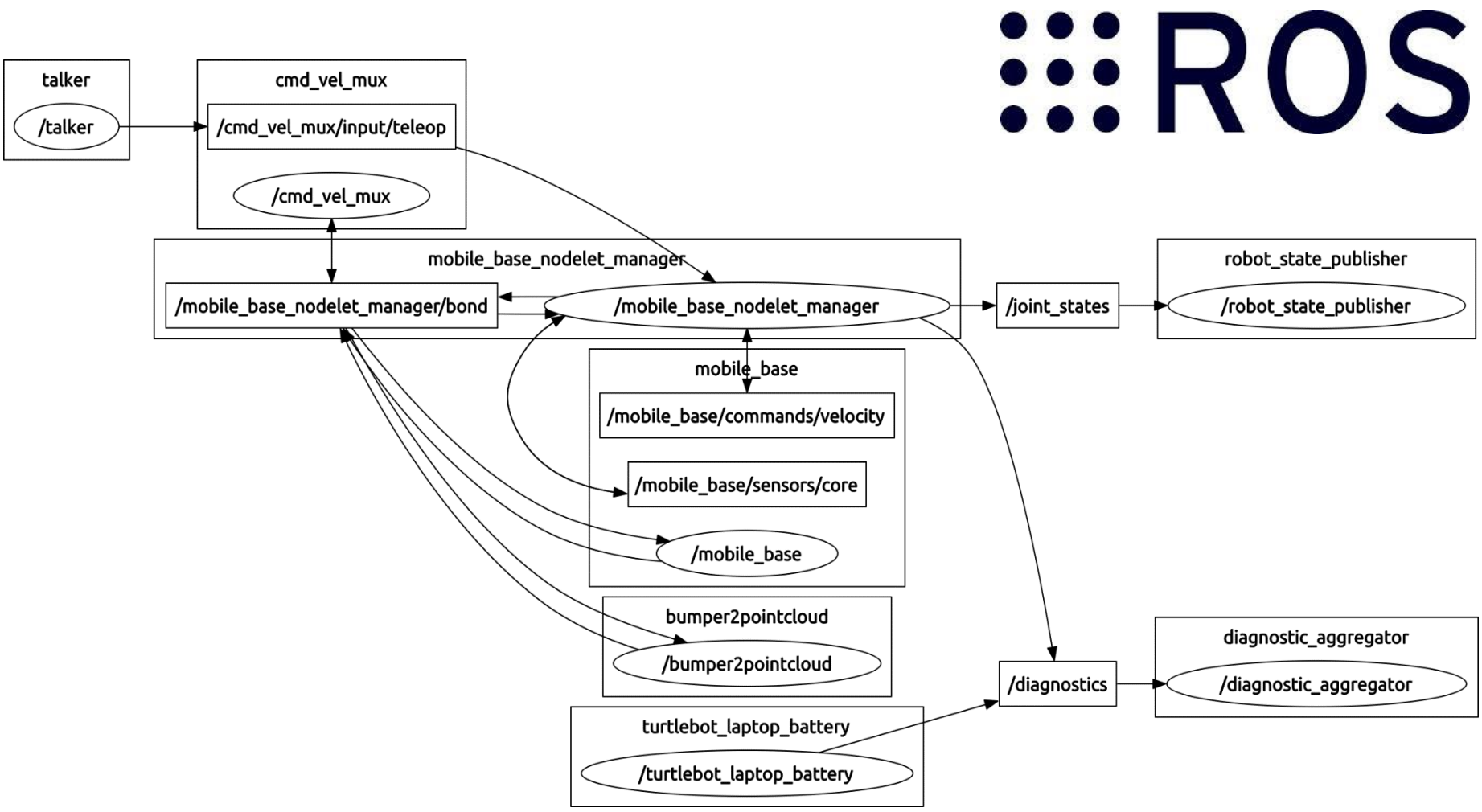
Google TensorFlow
Robot Operating System (ROS)
Torch
Theano
Caffe
CUDA + cuDNN
Tegra K1
Wifi + Bluetooth

Mobile Base

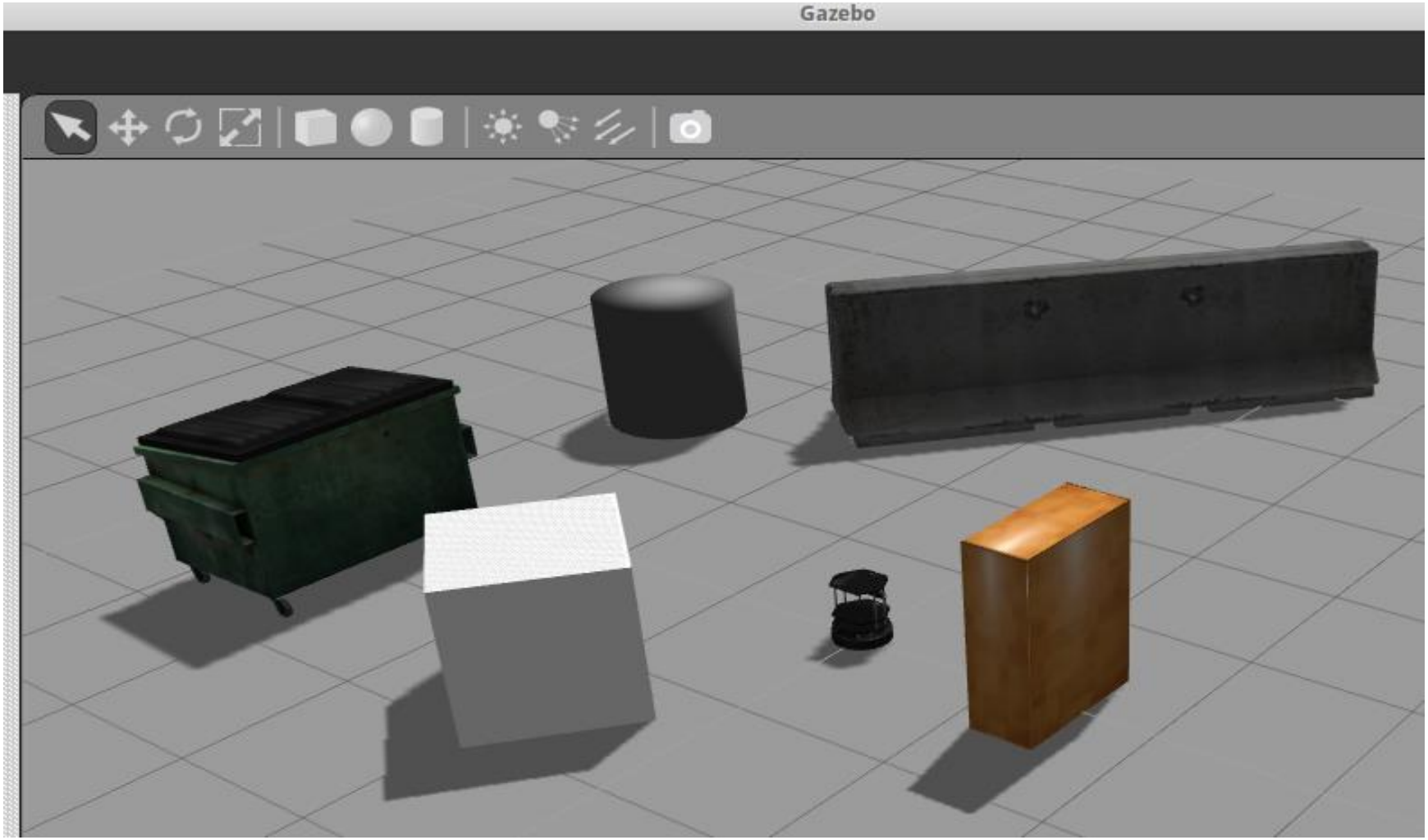
Turtlebot2 Base



Turtlebot2 Operating System



Turtlebot2 Simulator: Gazebo



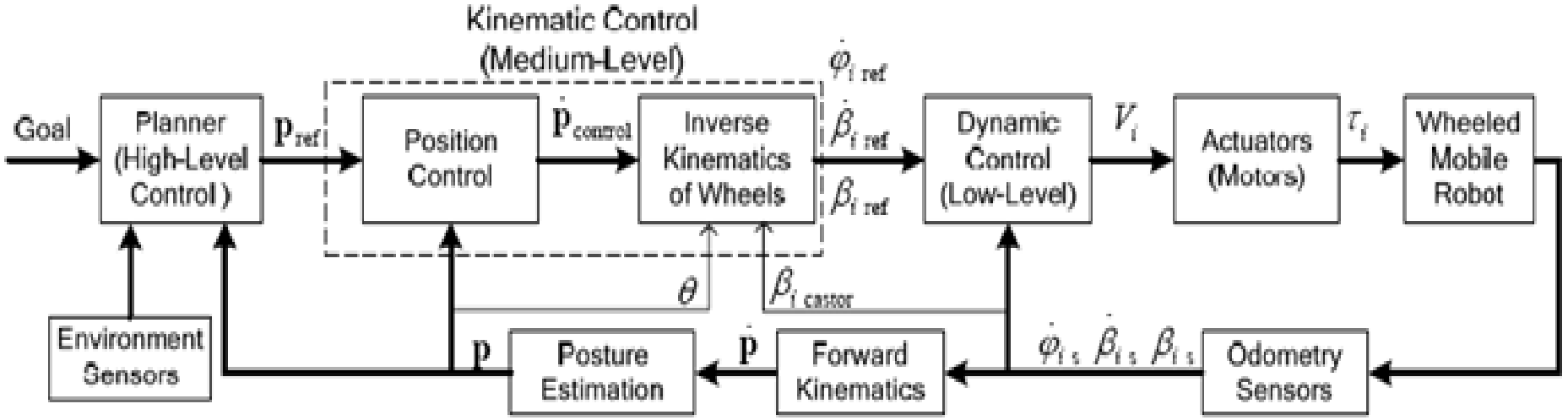
Outlines

- Introduction to Mobile Robot
- Control & Decision Paradigms
- Control & Decision Architectures

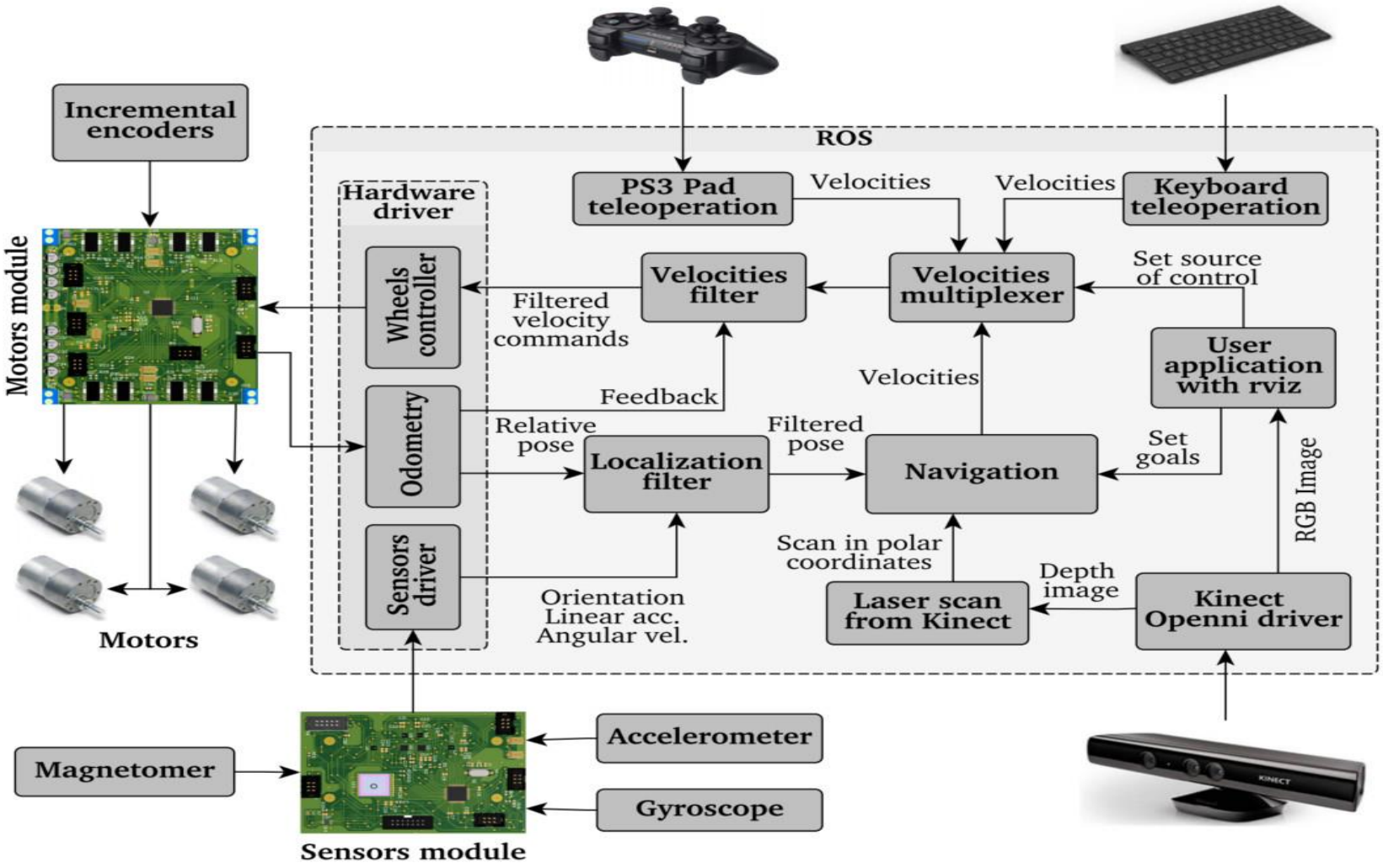
Control & Decision Paradigms

- Mathematical Model
- System Diagram
- Classical Paradigm
- Reactive Paradigm
- Hybrid Paradigm
- Potential Field Method

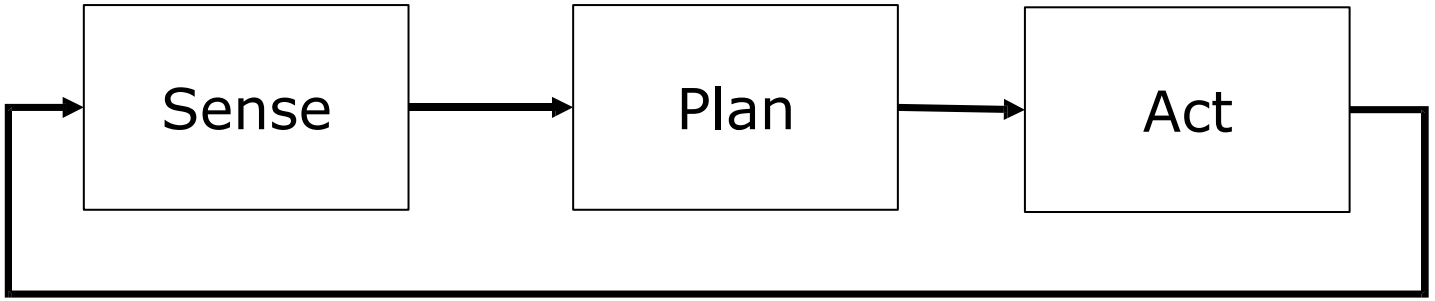
Mathematical Model



System Diagram

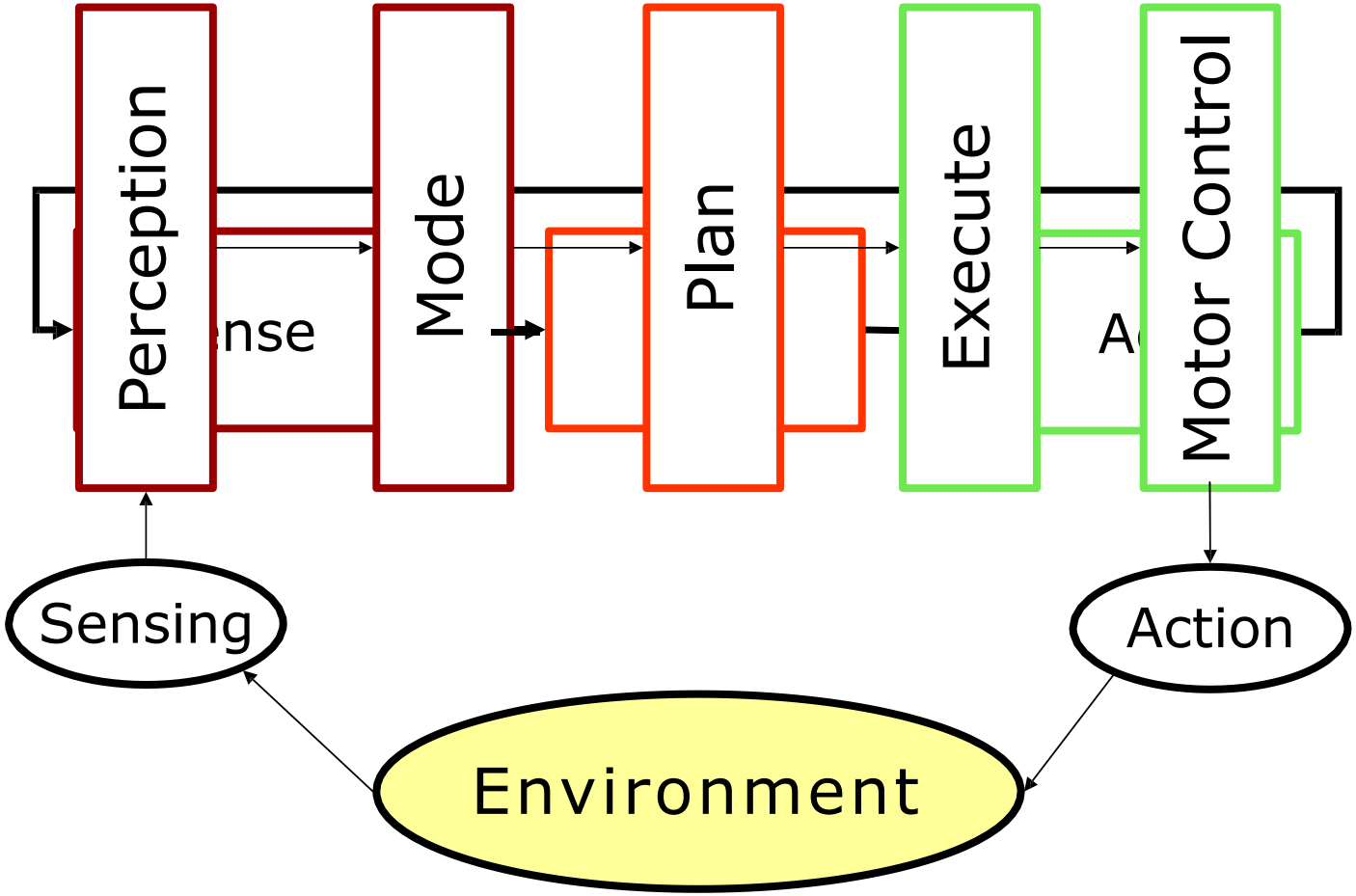


Classical / Hierarchical Paradigm

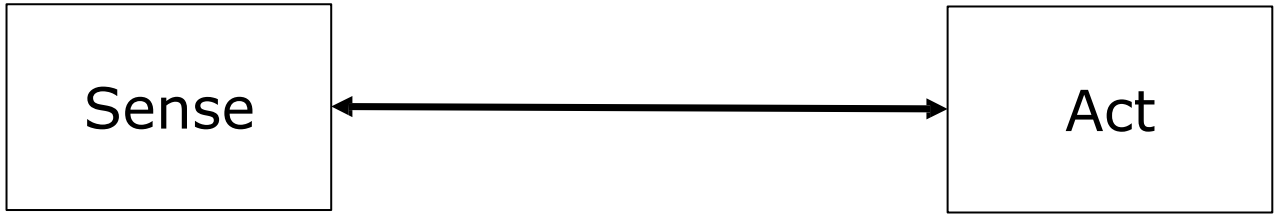


- 70's
- Focus on automated reasoning and knowledge representation
- STRIPS (Stanford Research Institute Problem Solver): Perfect world model, closed world assumption
- Find boxes and move them to designated position

Classical Paradigm: Horizontal Decomposition

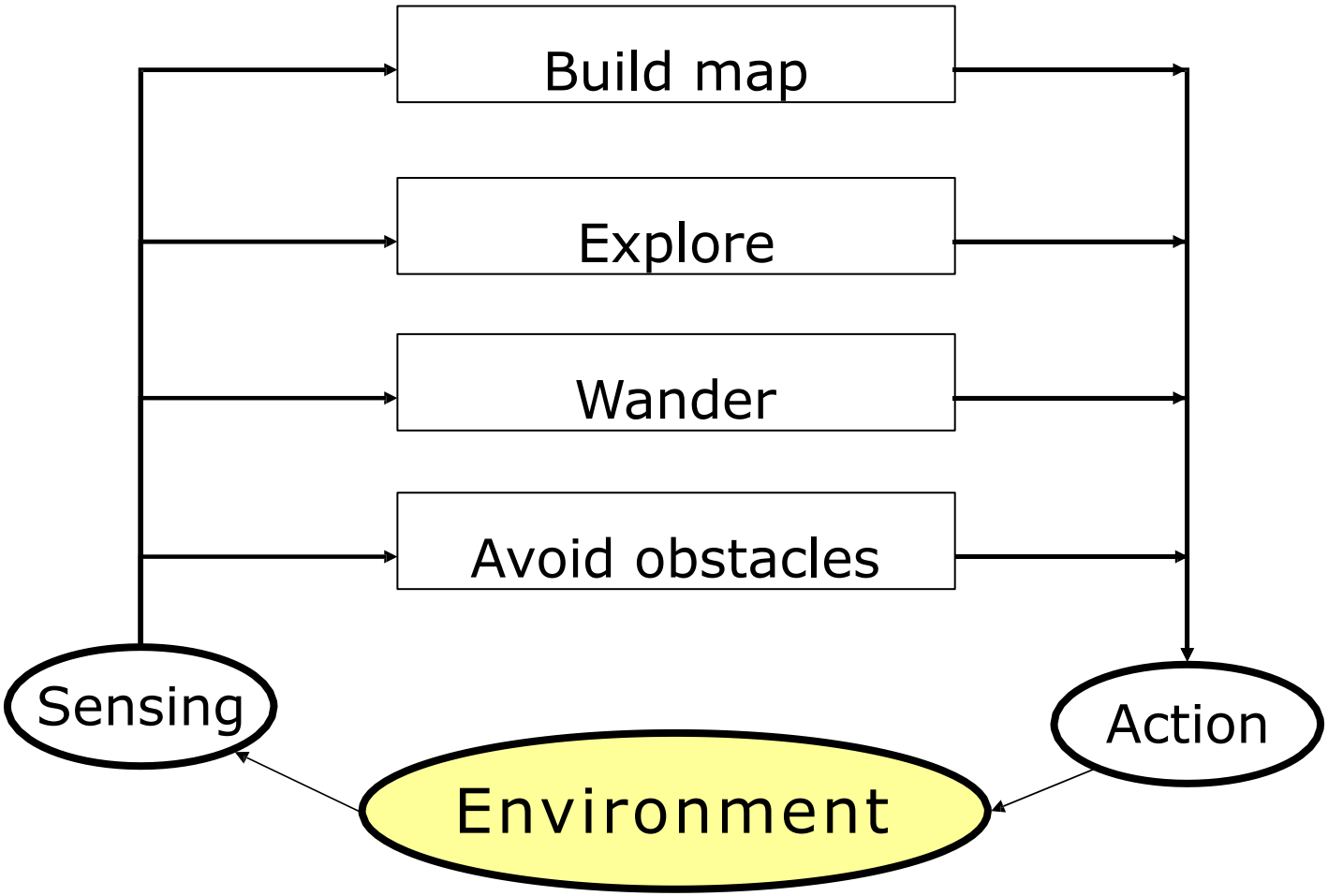


Reactive / Behavior-based Paradigm



- No models: The world is its own, best model
- Easy successes, but also limitations
- Investigate biological systems

Reactive Paradigm: Vertical Decomposition



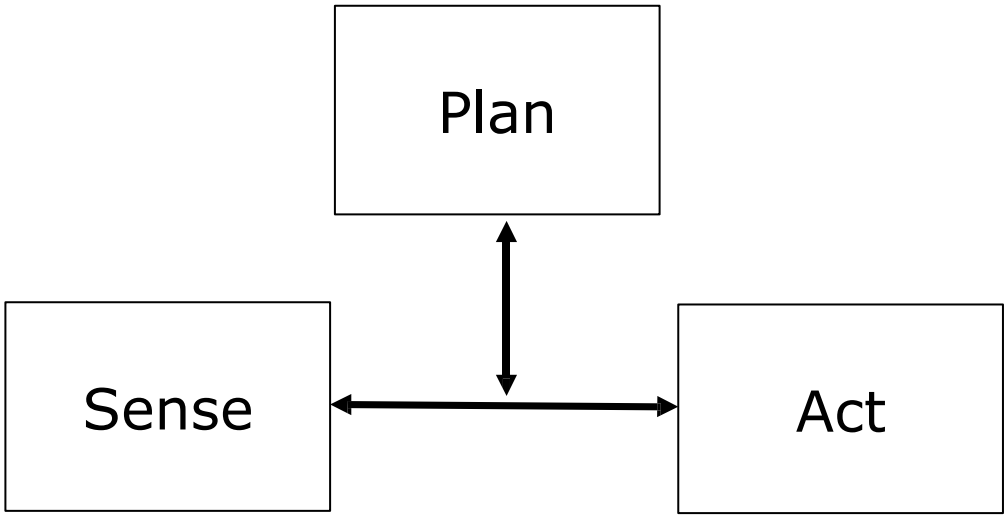
Characteristics of Reactive Paradigm

- Situated agent, robot is integral part of the world.
- No memory, controlled by what is happening in the world.
- Tight coupling between perception and action via behaviors.
- Only local, behavior-specific sensing is permitted (ego-centric representation).

Behaviors

- a direct mapping of sensory inputs to a pattern of motor actions that are then used to achieve a task.
- serve as the basic building block for robotics actions, and the overall behavior of the robot is emergent.
- support good software design principles due to modularity.

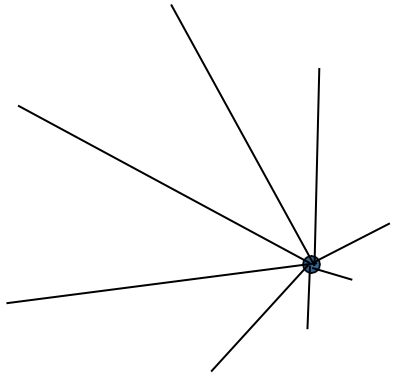
Hybrid Deliberative/Reactive Paradigm

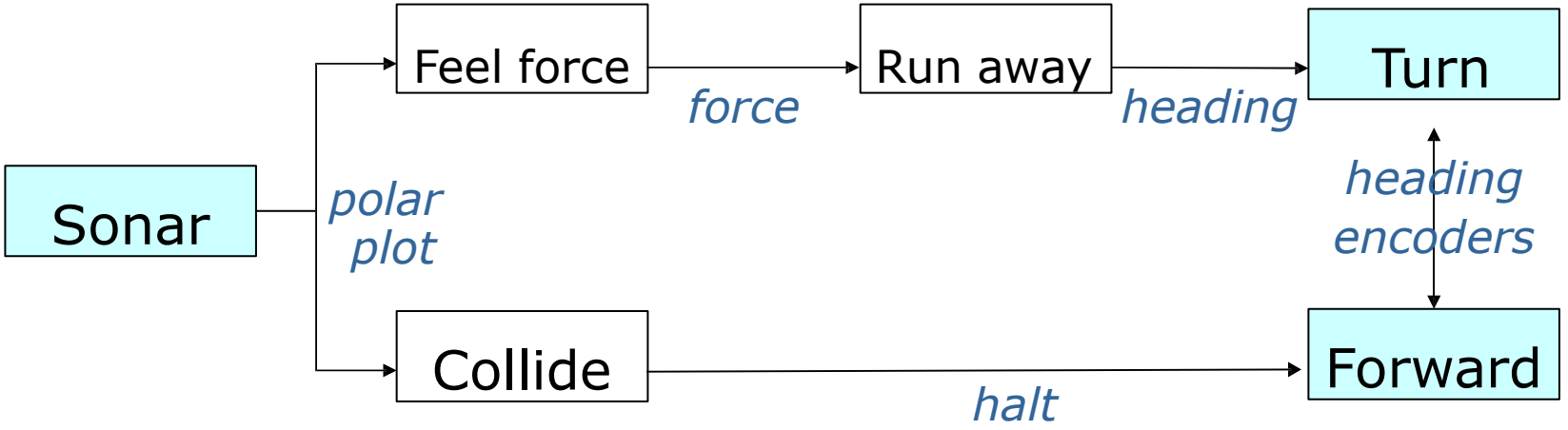


- Combines advantages of previous paradigms
  - World model used for planning
  - Closed loop, reactive control

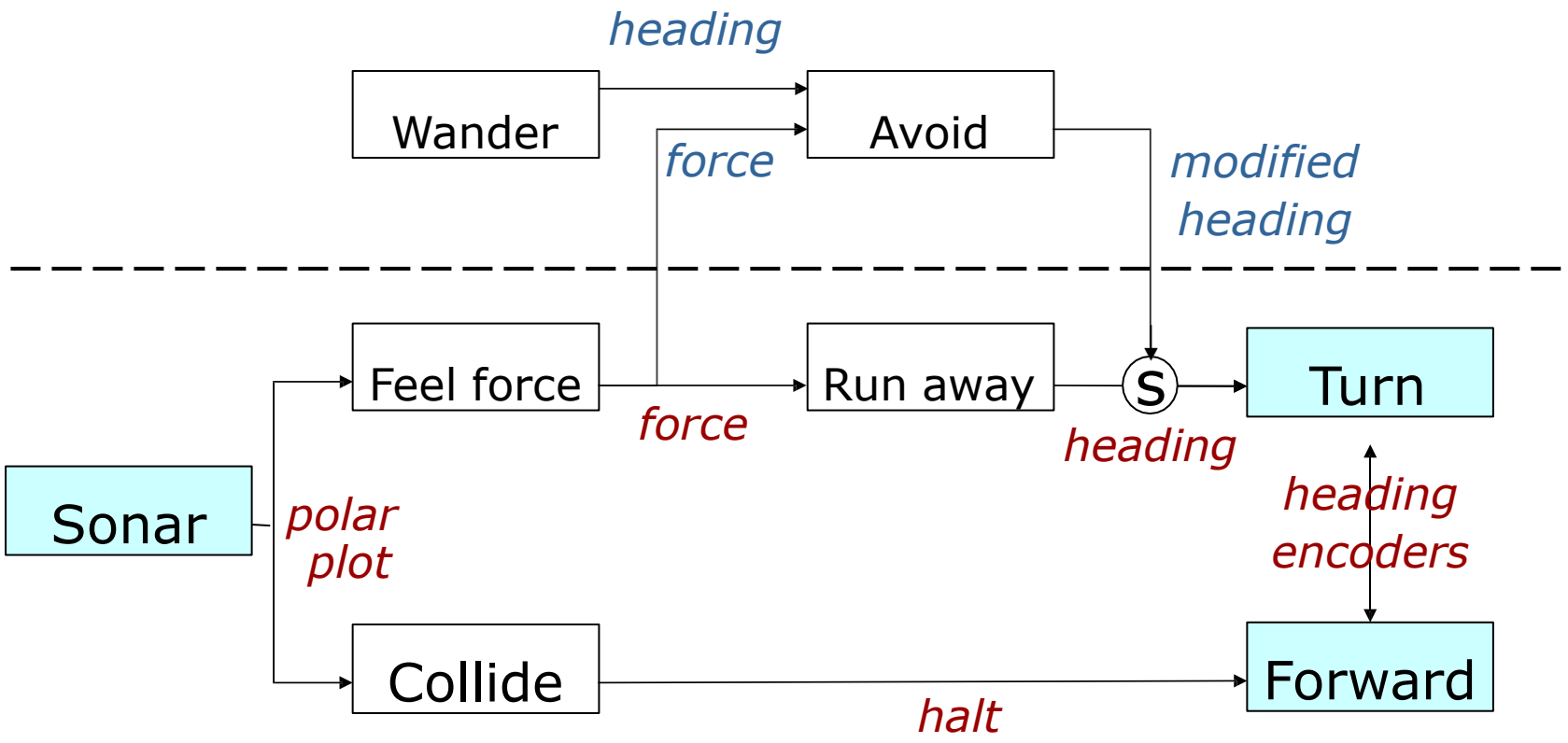
Level 0: Avoid

Polar plot of sonars

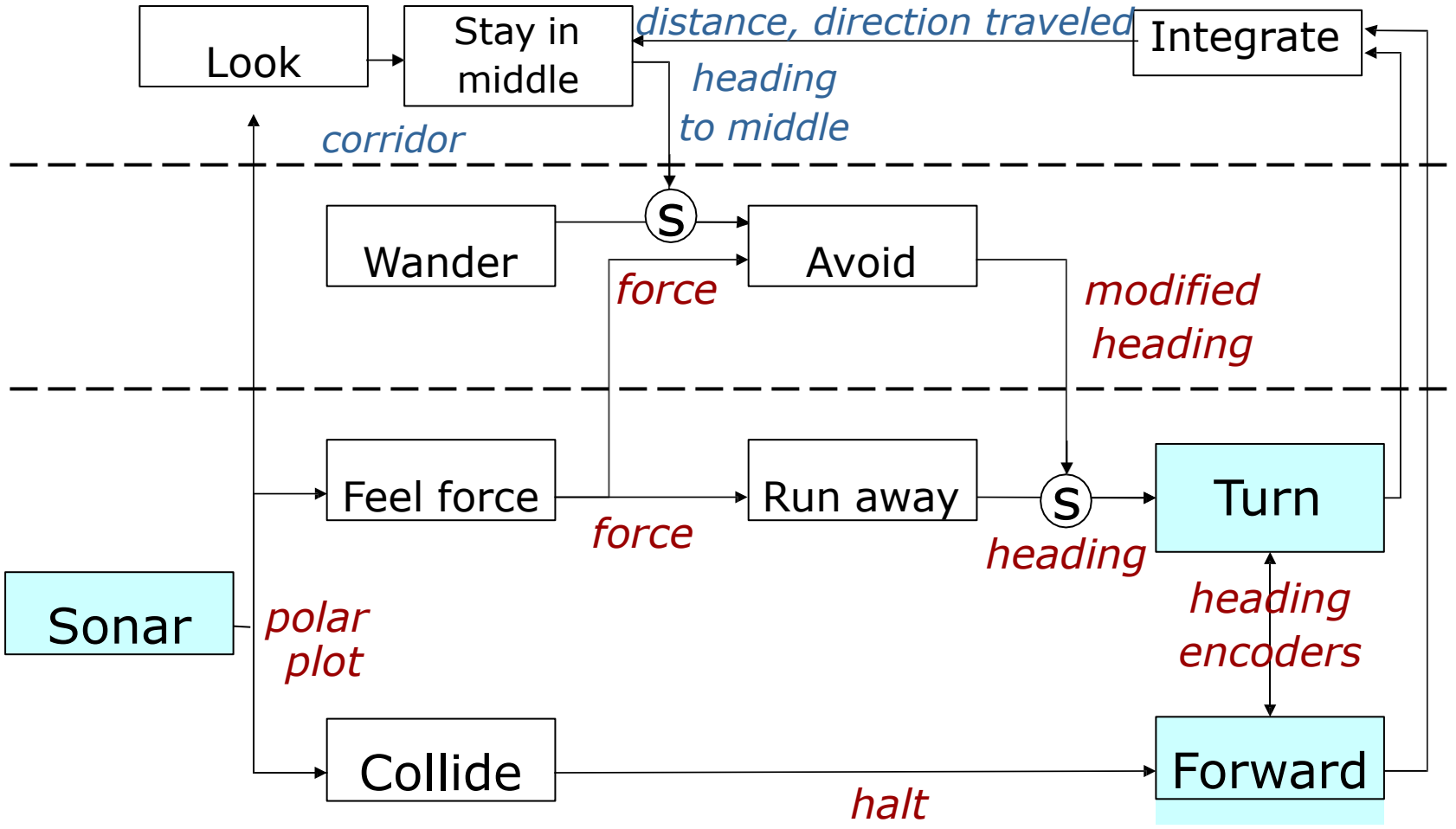




Level 1: Wander



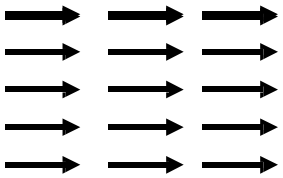
Level 2: Follow Corridor



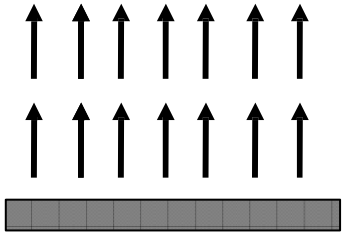
Potential Field Methodologies

- Treat robot as particle acting under the influence of a potential field
- Robot travels along the derivative of the potential
- Field depends on obstacles, desired travel directions and targets
- Resulting field (vector) is given by the summation of primitive fields
- Strength of field may change with distance to obstacle/target

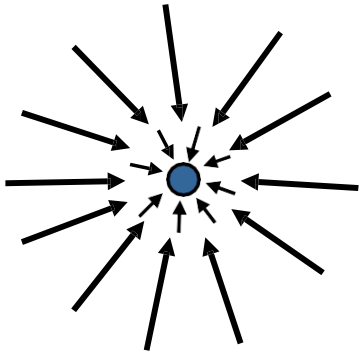
Primitive Potential Fields



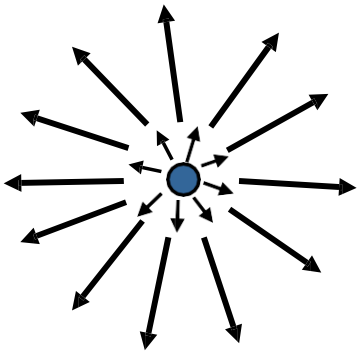
Uniform



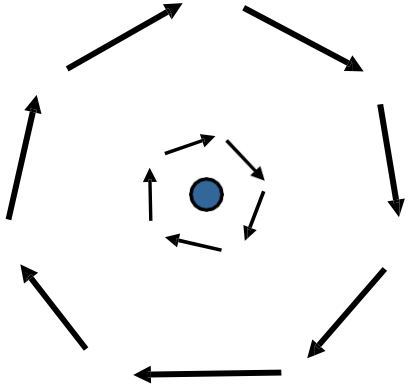
Perpendicular



Attractive



Repulsive



Tangential

Corridor Following With Potential Fields

- Level 0 (collision avoidance)
  is done by the repulsive fields of detected obstacles.

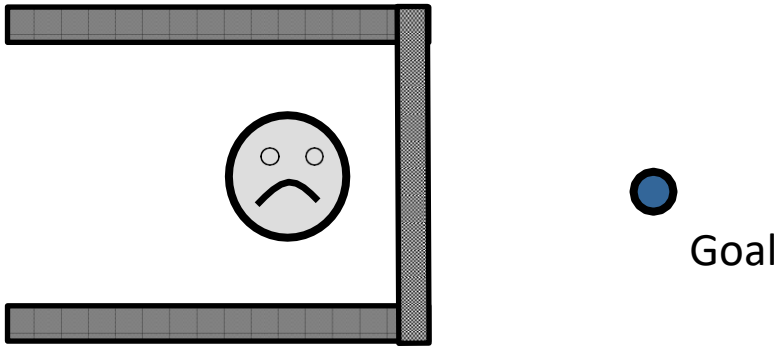
- Level 1 (wander)

  adds a uniform field.

- Level 2 (corridor following)
  replaces the wander field by three fields (two perpendicular, one uniform).

Characteristics of Potential Fields

- Suffer from local minima



  - Backtracking
  - Random motion to escape local minimum
  - Procedural planner s.a. wall following
  - Increase potential of visited regions
  - Avoid local minima by harmonic functions

Characteristics of Potential Fields

- No preference among layers
- Easy to visualize
- Easy to combine different fields
- High update rates necessary
- Parameter tuning important

Reactive Paradigm

- Representations?
- Good software engineering principles?
- Easy to program?
- Robustness?
- Scalability?

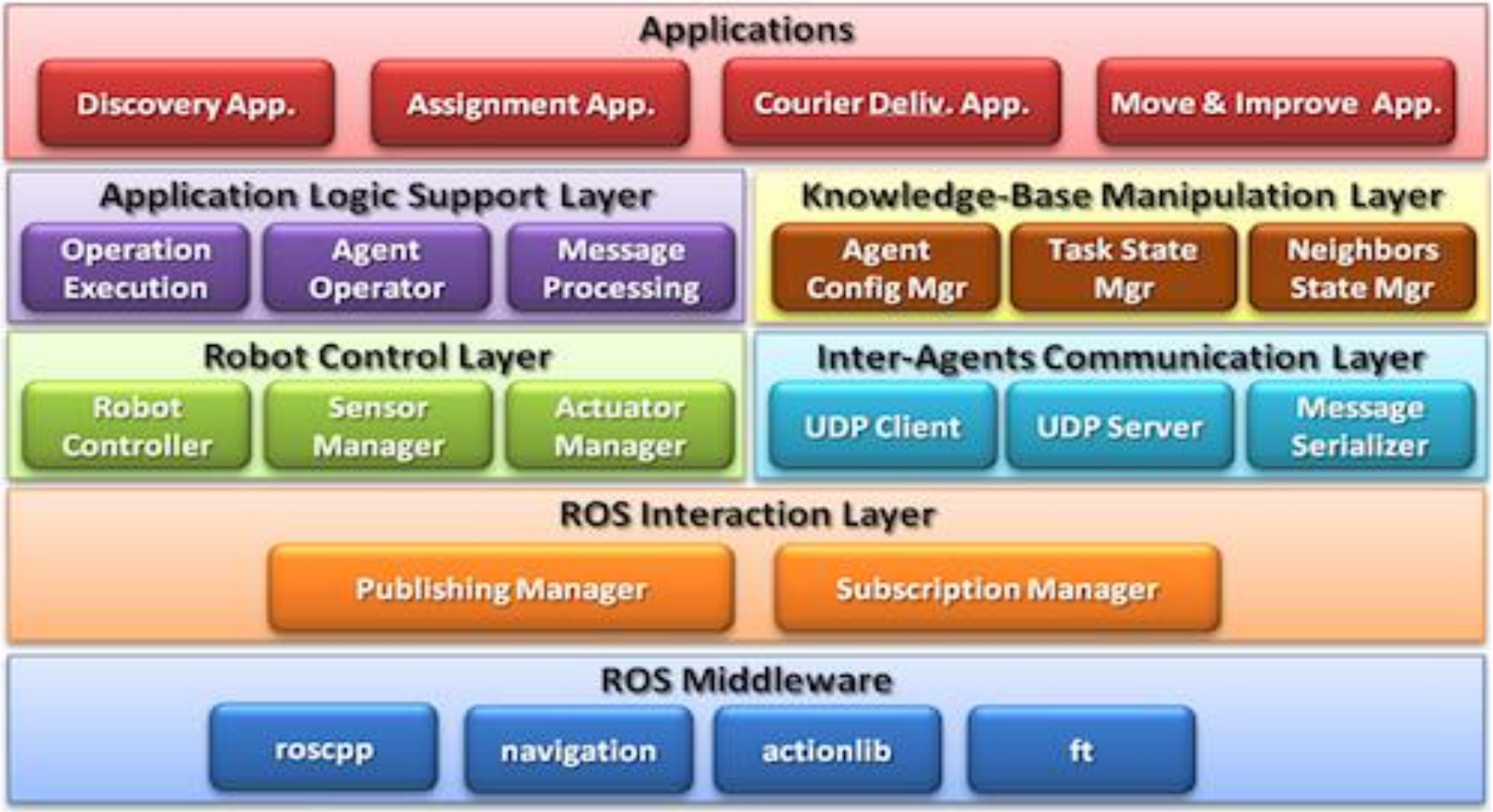
Outlines

- Introduction to Mobile Robot
- Control & Decision Paradigms
- Control & Decision Architectures

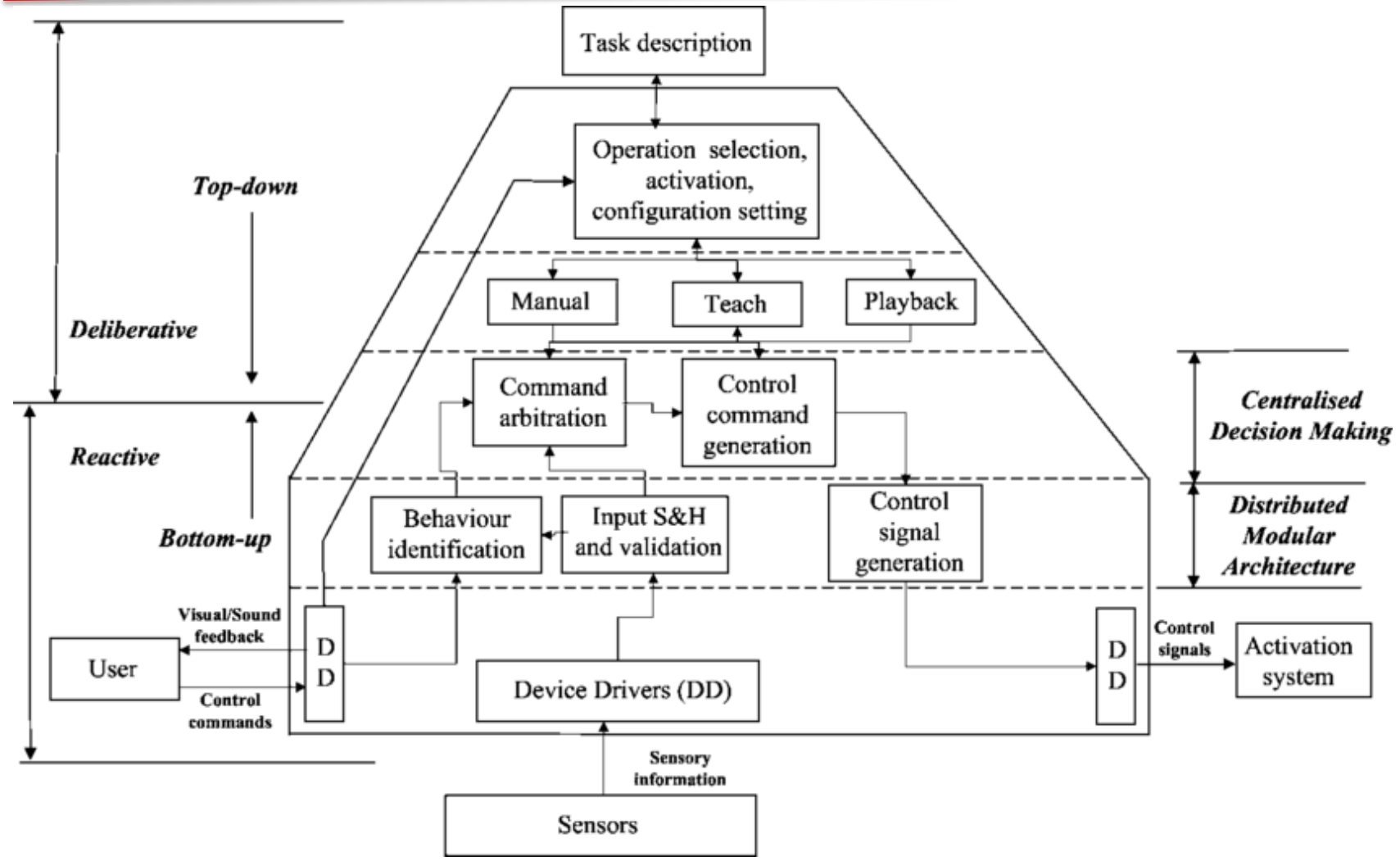
Control & Decision Architectures

- COROS Architecture and Development Framework
- Control and Decision Architecture
- Software Architecture

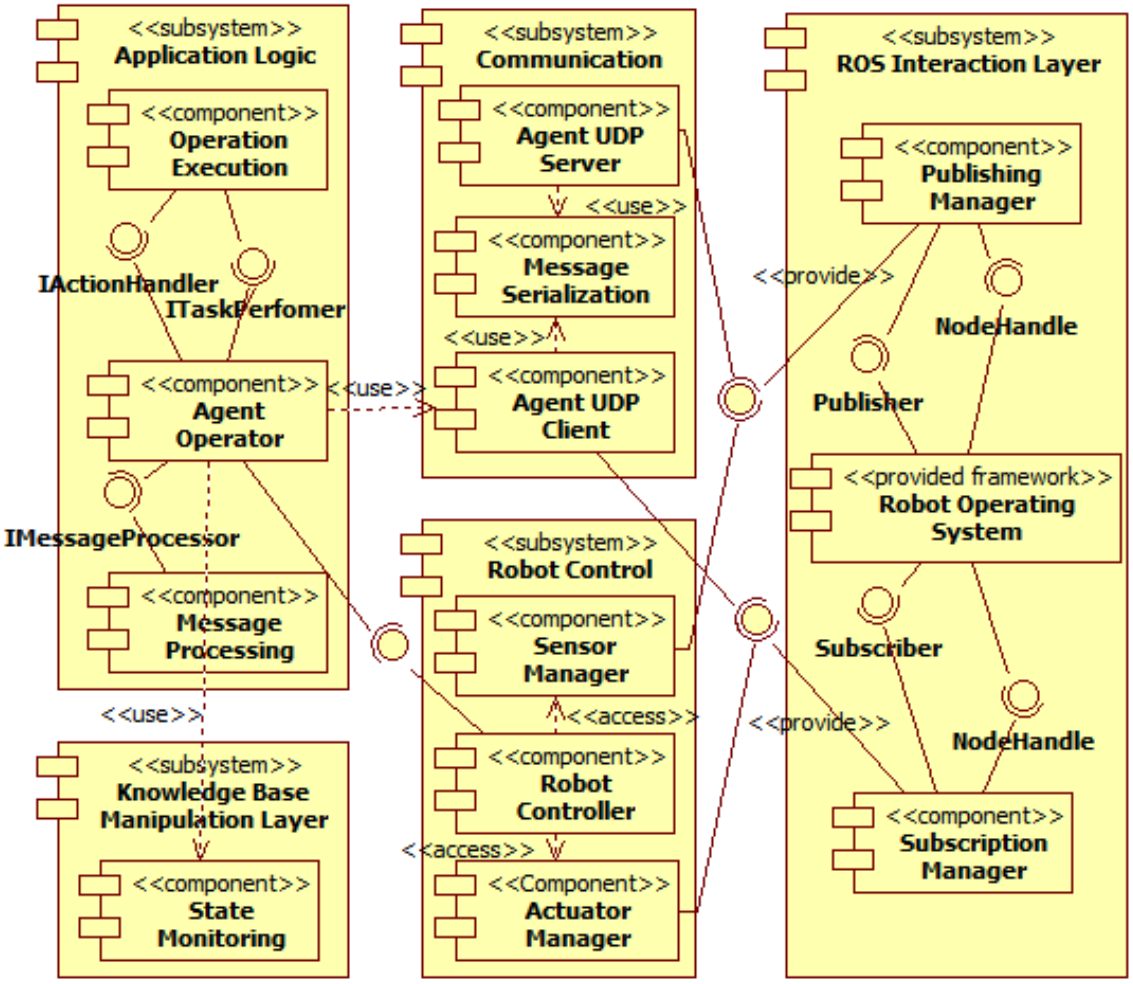
COROS Architecture (Multi-agent System)



Control & Decision Architecture



Software Architecture



Discussion

- Imagine you want your robot to perform navigation tasks, which approach would you choose?
- What are the benefits of the behavior based paradigm?
- Which approaches will win in the long run?