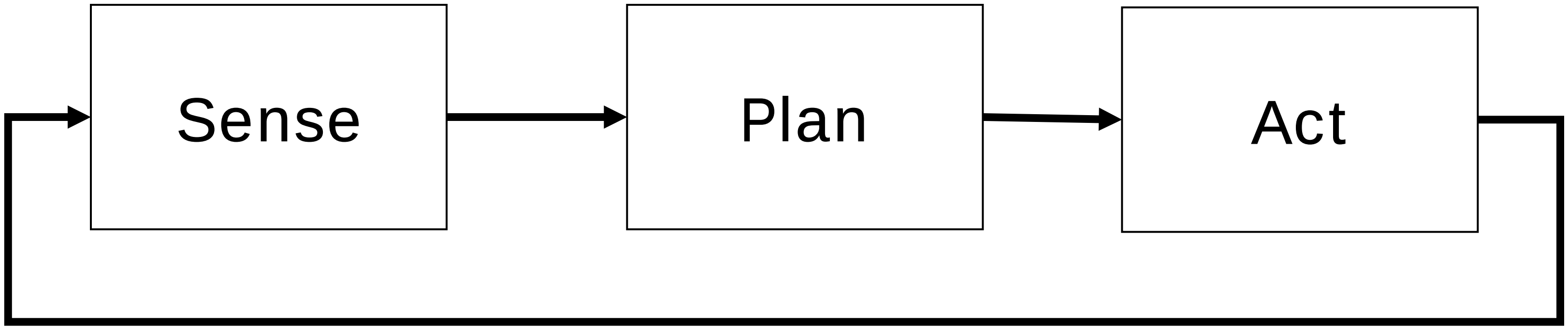
Introduction to Mobile Robotics

Robot Control Paradigms

Wolfram Burgard

Classical / Hierarchical Paradigm



- 70s
- 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 the designated position

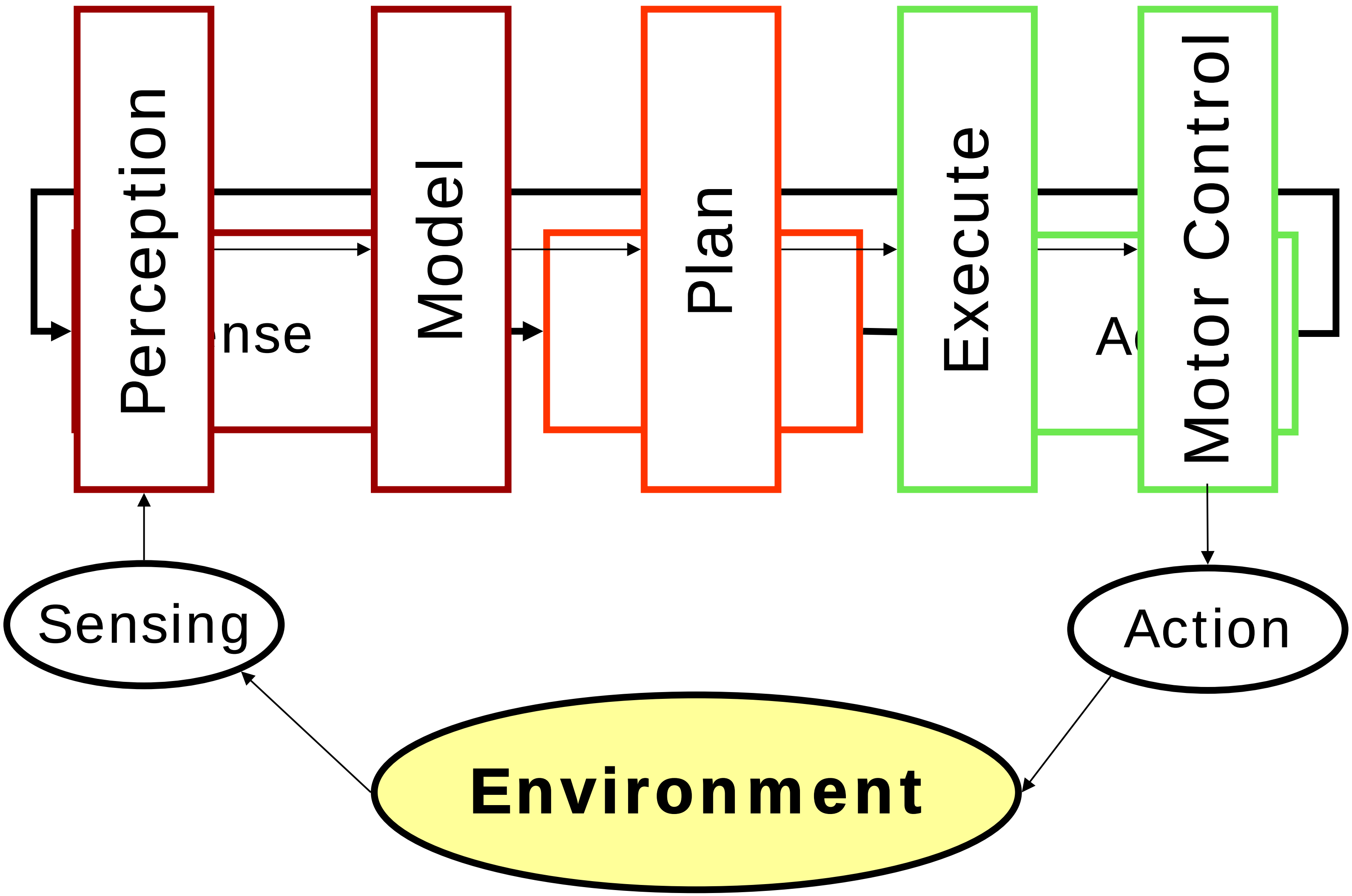
Stanford CART 1973

Stanford AI Laboratory / CMU (Moravec)

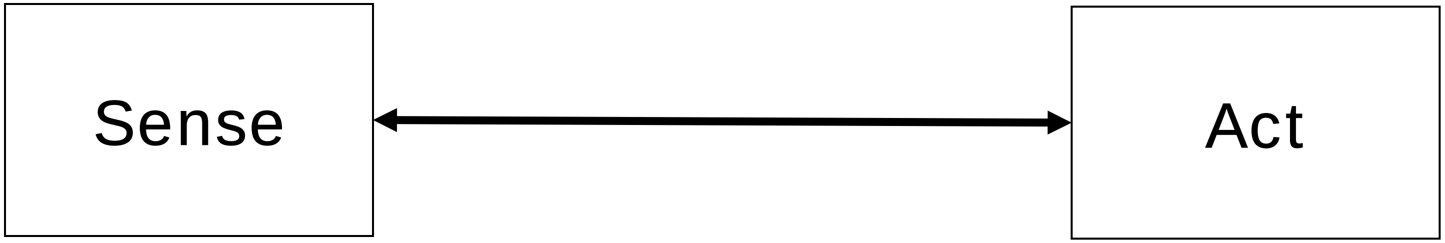
Classical Paradigm
Stanford Cart

1. Take nine images of the environment, identify interesting points in one image, and use other images to obtain depth estimates.
2. Integrate information into global world model.
3. Correlate images with previous image set to estimate robot motion.
4. On basis of desired motion, estimated motion, and current estimate of environment, determine direction in which to move.
5. Execute the motion.

Classical Paradigm as Horizontal/Functional Decomposition

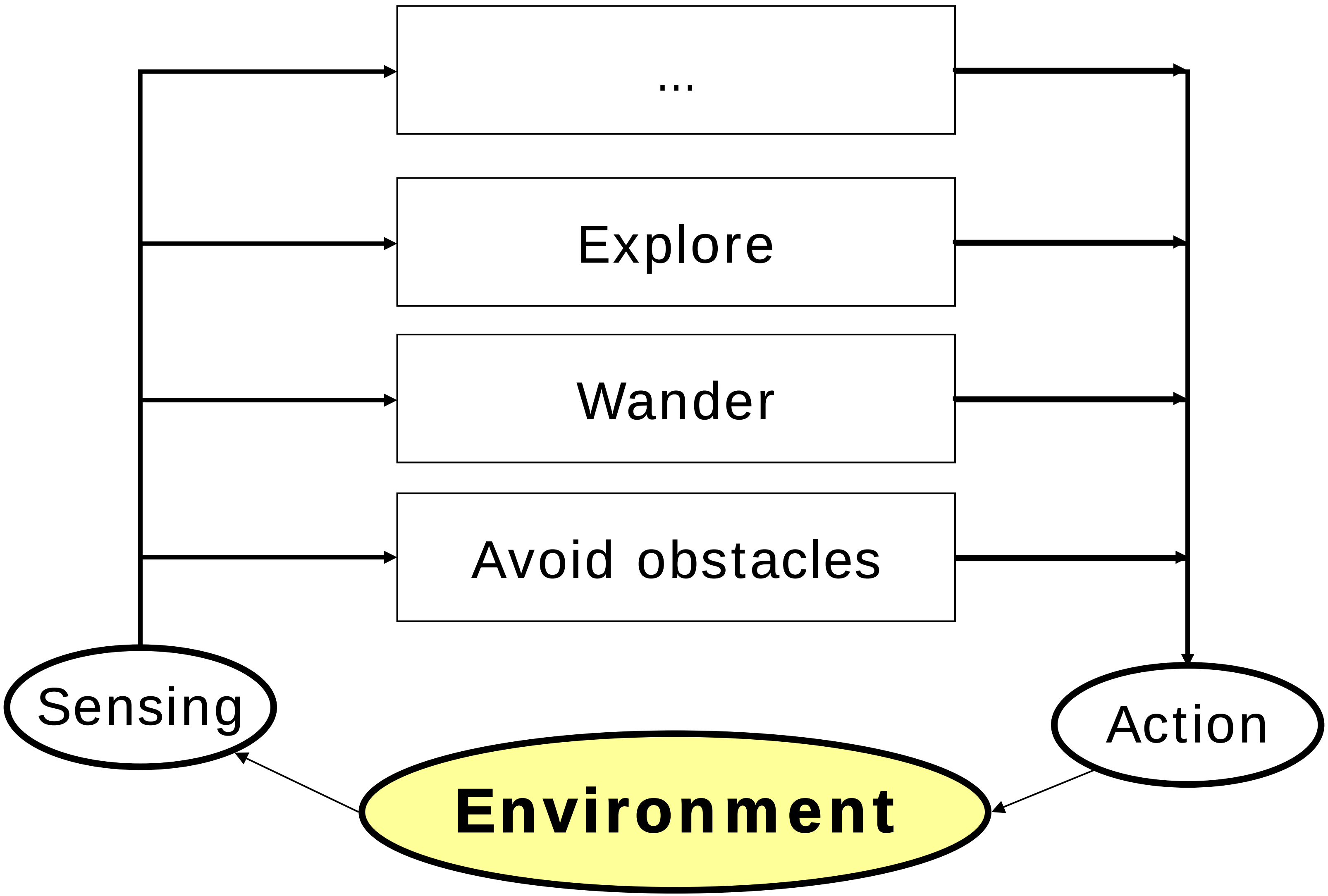


Reactive / Behavior-based Paradigm



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

Reactive Paradigm as Vertical Decomposition



Characteristics of Reactive Paradigm

- Situated agent, robot is integral part of its environment.
- 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

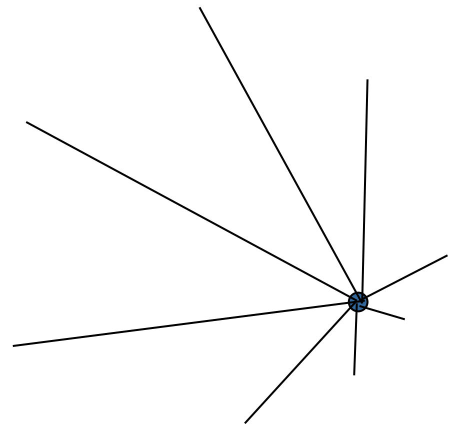
- ... are a direct mapping of sensory inputs to a pattern of motor actions that are then used to achieve a task.
- ... serve as the basic building blocks for robot actions, and the overall behavior of the robot is emergent.
- ... support good software design principles due to modularity.

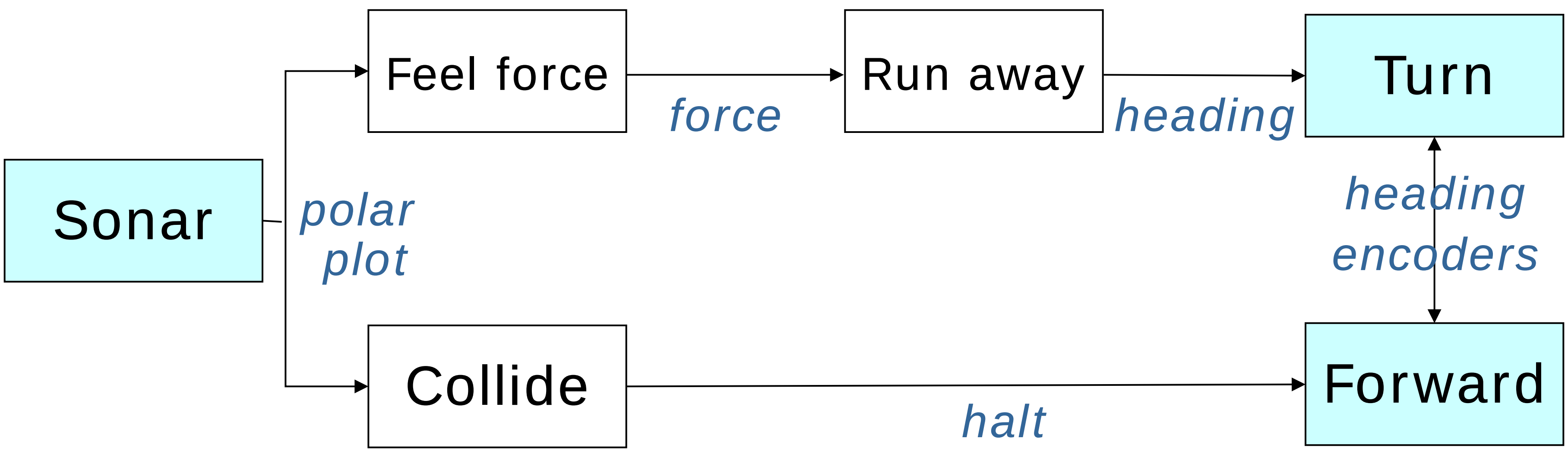
Subsumption Architecture

- Introduced by Rodney Brooks ‘86.
- Behaviors are networks of sensing and acting modules (augmented finite state machines AFSM).
- Modules are grouped into layers of competence.
- Layers can subsume lower layers.
- No internal state!

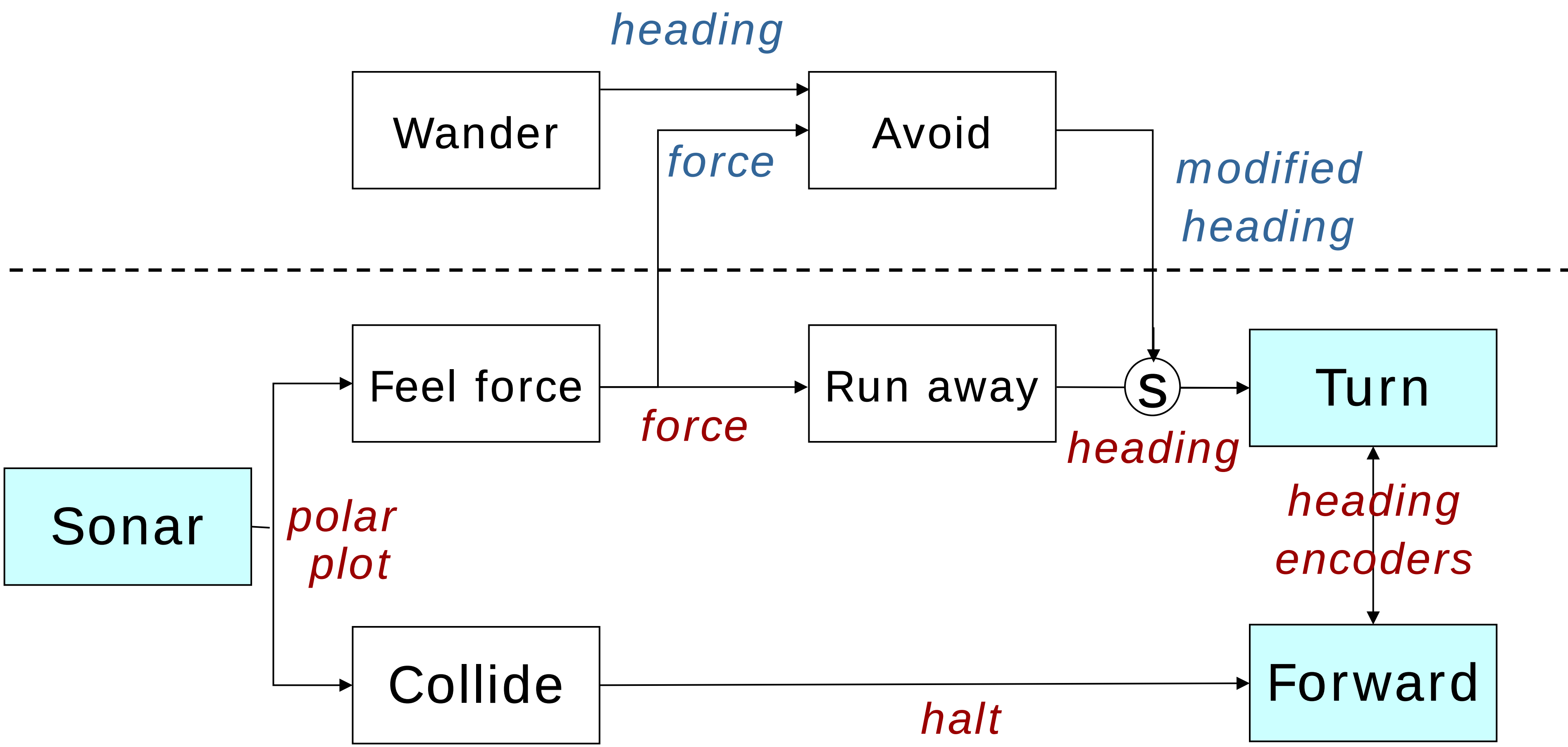
Level 0: Avoid

Polar plot of sonars

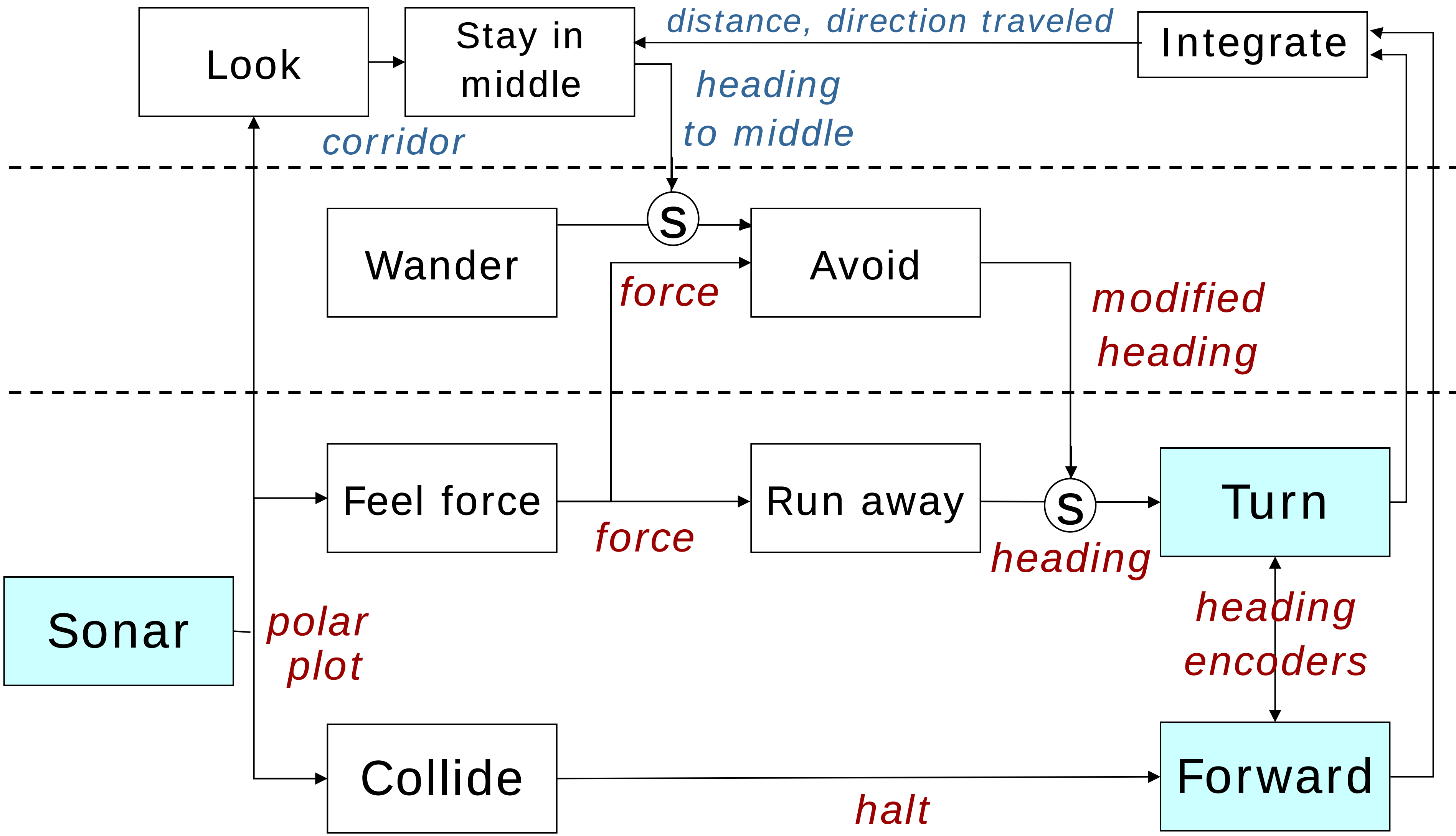




Level 1: Wander



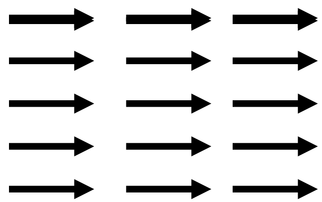
Level 2: Follow Corridor



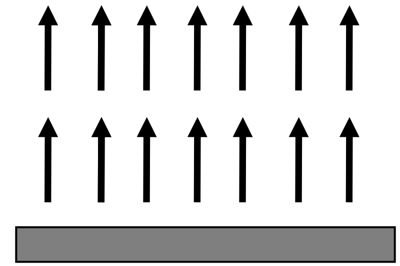
Potential Field Methods

- 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

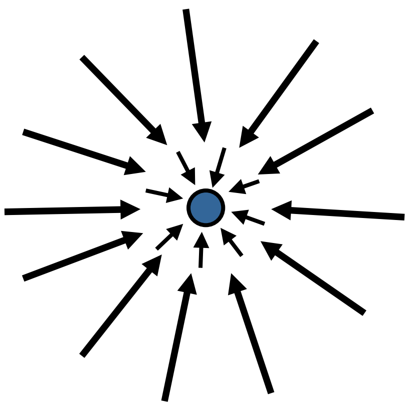
Primitive Potential Fields



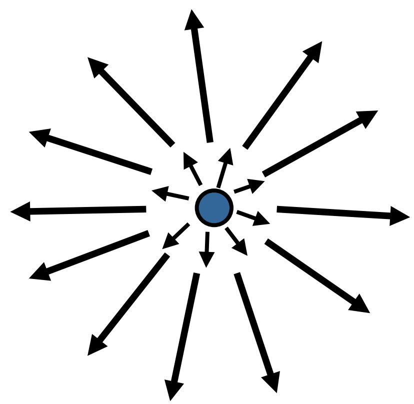
Uniform



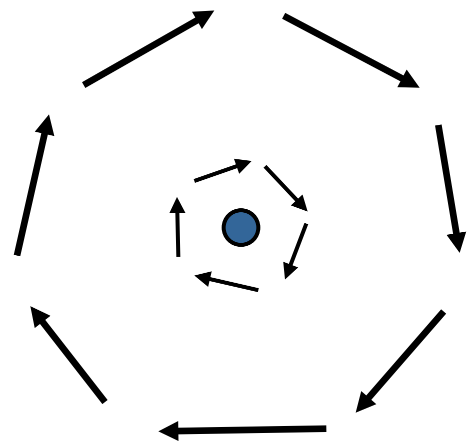
Perpendicular



Attractive



Repulsive



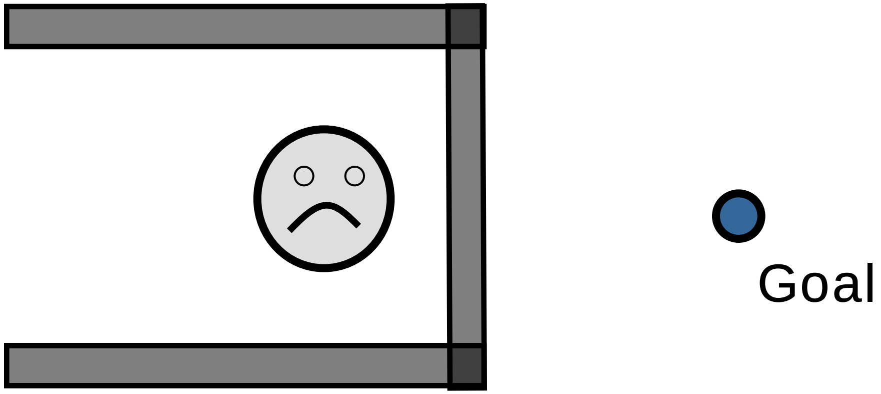
Tangential

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

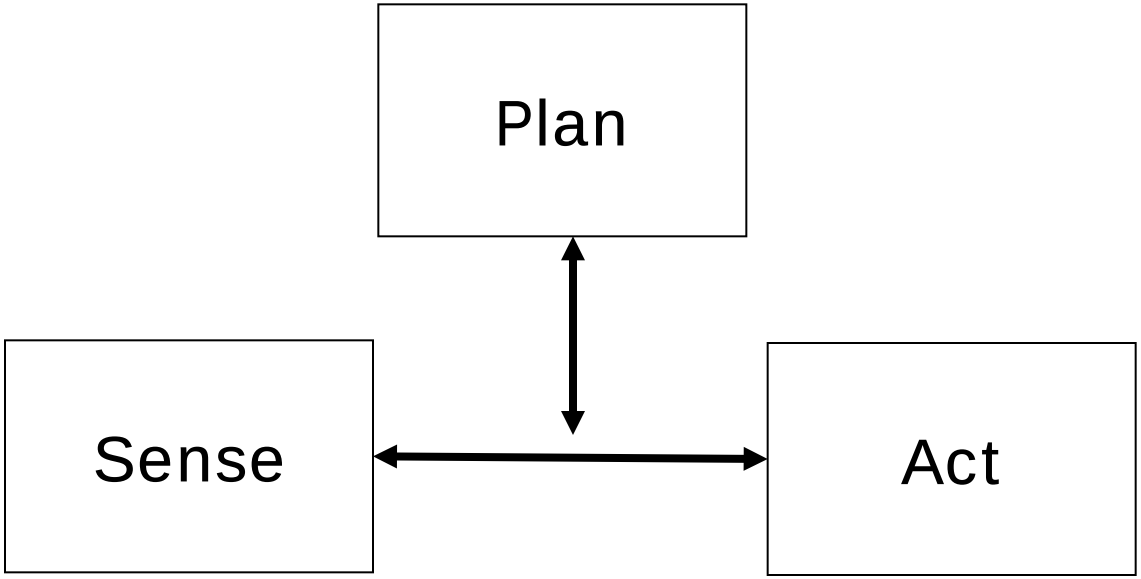
Characteristics of Potential Fields

- 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?

Hybrid Deliberative/Reactive Paradigm



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

Discussion

- Imagine you want your robot to perform navigation tasks, which approach would you choose?
- What are the benefits and drawbacks of the behavior based paradigm?
- What are drawbacks of the subsumption architecture?