EGN 4060c: Introduction to Robotics

Lecture 5:

Robot Architectures: Reactive and Hybrid

Instructor: Dr. Gita Sukthankar
Email: gitars@eecs.ucf.edu

Lab Report

- First one due Sept 17th in webcourses
- Between 1-2 page writeup describing your robot code
- Include any other pictures or results specifically asked for (e.g., the tables of measurements)
- Useful styles
  - Pseudocode
  - Javadoc style comments about key methods
  - List of steps describing operation of robot
  - Free form text does **not** usually work well
- Accompanying java files for the main section of program and any other new classes and methods introduced (CreateMove.java)
- Graded on: clarity, detail, and correctness (1-3 pts)

Example Report

Common Problems

- Spelling and grammar issues
  - Fixing these minor problems is relatively easy and can help make a better impression on the reader.
  - Have a friend proof-read your document.
- Writeup was a synopsis of a single source document written for a general audience.
  - Better to synthesize material from several sources. This add more value for the reader rather than just being a summary of existing material.
- Include references after technical details.
  - When you include specific details (e.g., the robot weighs 10 kgs) it is usually good to include a citation after the sentence or the fact.

Common Problems

- Summary isn't coherent but more a collection of slightly related facts
  - Make an outline
  - Introduce the project, the purpose of the project, what makes it interesting/hard
  - Method---hardware, software, any details which might be useful. Break it up into appropriate subtopics
  - Results/evaluation
  - Conclusion: what has been achieved, future things the researchers are planning to work on
  - Move from general ideas to specific ones

References

- Most commonly, citations will either be numbers or [author, year] format.
- Generally a reference should include a title and author even if it is only a web page.
- When you use images from other sources it is important that a citation appear in a caption to the image. For work that is published, you need to obtain permission from the original source to use the image.

Types of Publications

- Group website: non-archival documents and videos summarizing group's research
- Workshop: preliminary results
- Conference: complete results of one aspect of the project
- Journal: longer, most complete and detailed version of research project
- Popular magazine article: high-level description of project (skimps on methods and results)
- Textbook: endeavors to present a balanced view covering multiple research projects in a given topic area

Guide to Reading Papers

- Skim the introduction
- Skip to the method section to determine precisely what the authors did
  - Simulation, partial-robot, full implementation, field-test?
- Look at both the references and the related work section to determine what else has been done in that area
- If still interested, give the results and the discussion section a detailed reading to discover and evaluate what the authors accomplished

Control Architecture Types

- Deliberative control
- Reactive control
- Hybrid control
- Behavior-based control

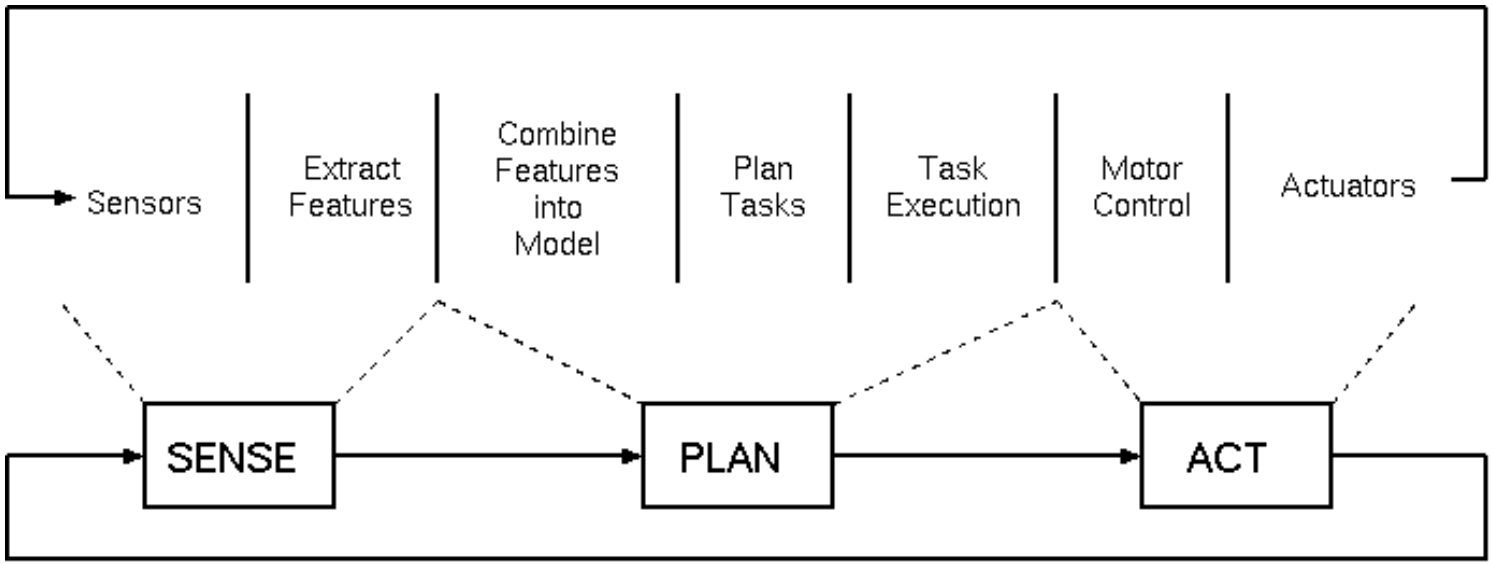
Reactive Architecture

- No maps, no state
- No look ahead
- No planner, no need for fancy search techniques
- Biologically inspired by S-R behaviors in animals

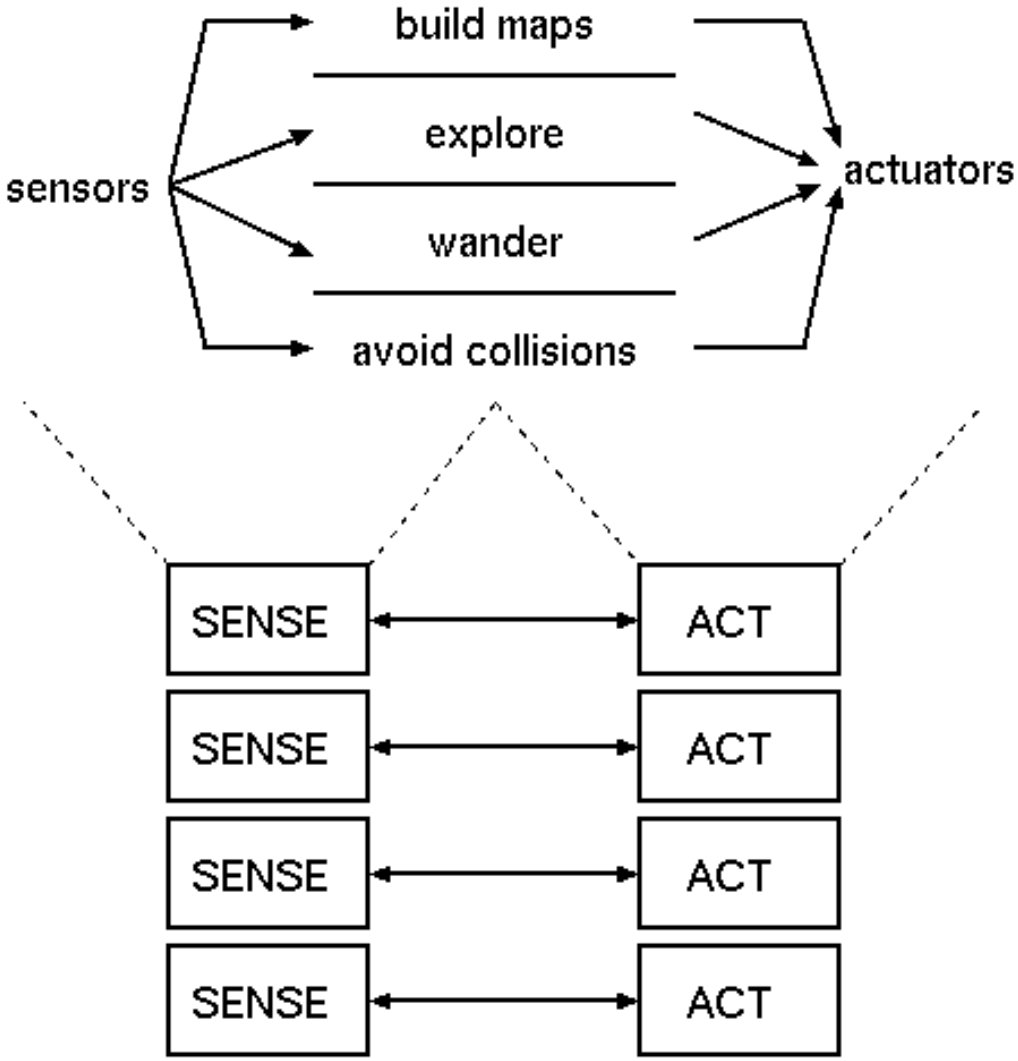
Animal Behavior

- Some robots model the form of animals (Sony Aibo).
- Many roboticists have been inspired to mimic and model animal behaviors.
  - fleeing
  - foraging
  - taxes (movement towards a particular orientation)
  - homing
- Reactive architectures have been used to model many types of animal behaviors.
- Genghis video http://www.youtube.com/watch?v=K2xUHYFcYKI

Deliberative Architectures are “Horizontal”

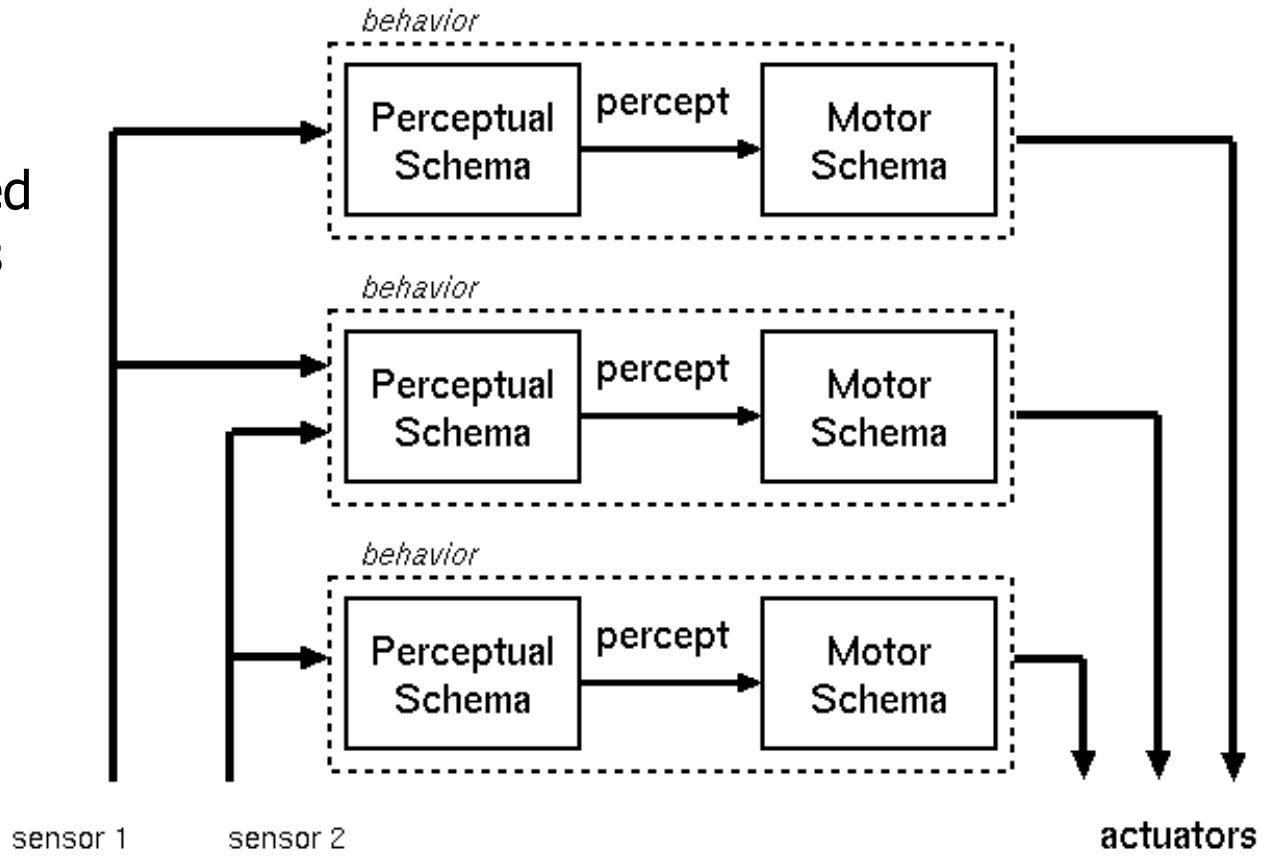


More Biological is “Vertical”



Sensing is Behavior-Specific or Local

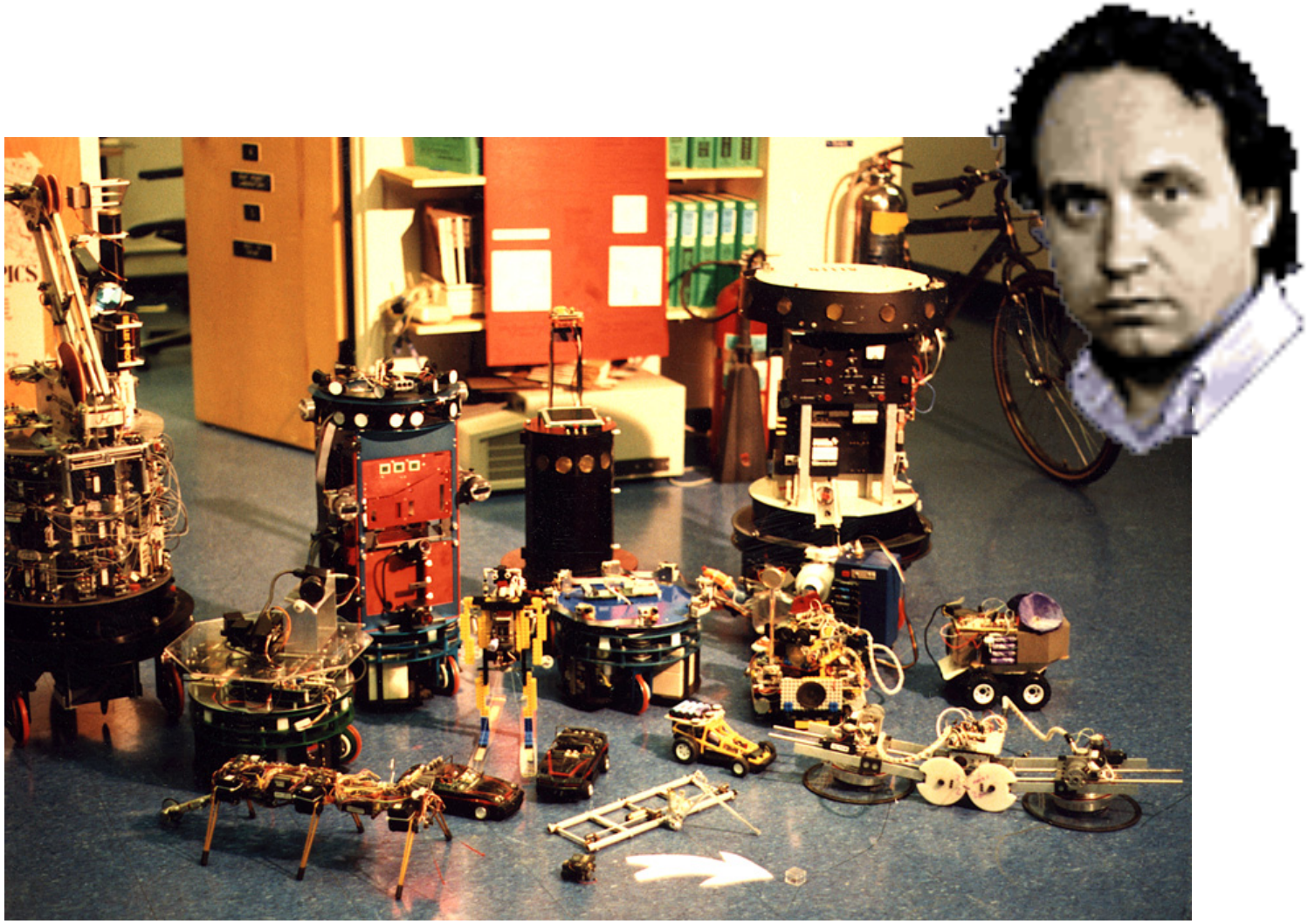
No central world model maintained across behaviors



Reactive

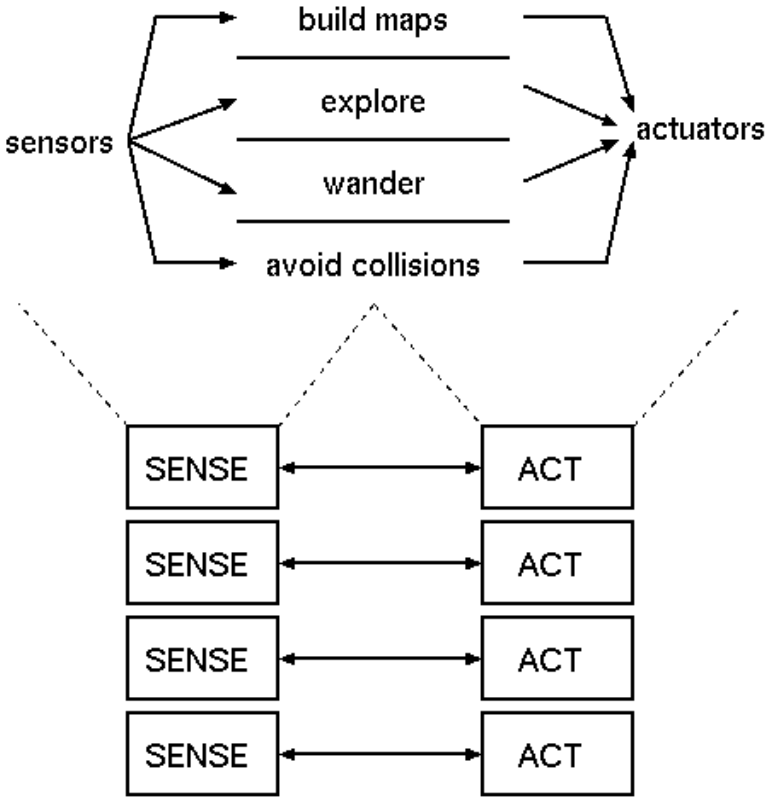
- Historically, there are two main styles of creating a reactive system
  - Subsumption architecture
    - Layers of behavioral competence
    - How to control relationships
  - Potential fields
    - Concurrent behaviors
    - How to navigate
- They are equivalent in power; the main difference is in how the behaviors are combined.

Subsumption (Brooks, MIT)



Subsumption Philosophy

- Modules should be grouped into *layers of competence*
- Modules in a higher lever can override or *subsume* behaviors in the next lower level
  - Suppression: substitute input going to a module
  - Inhibit: turn off output from a module
- ***No internal state*** in the sense of a local, persistent representation similar to a world model.
- Architecture should be *taskable*: accomplished by a higher level turning on/ off lower layers

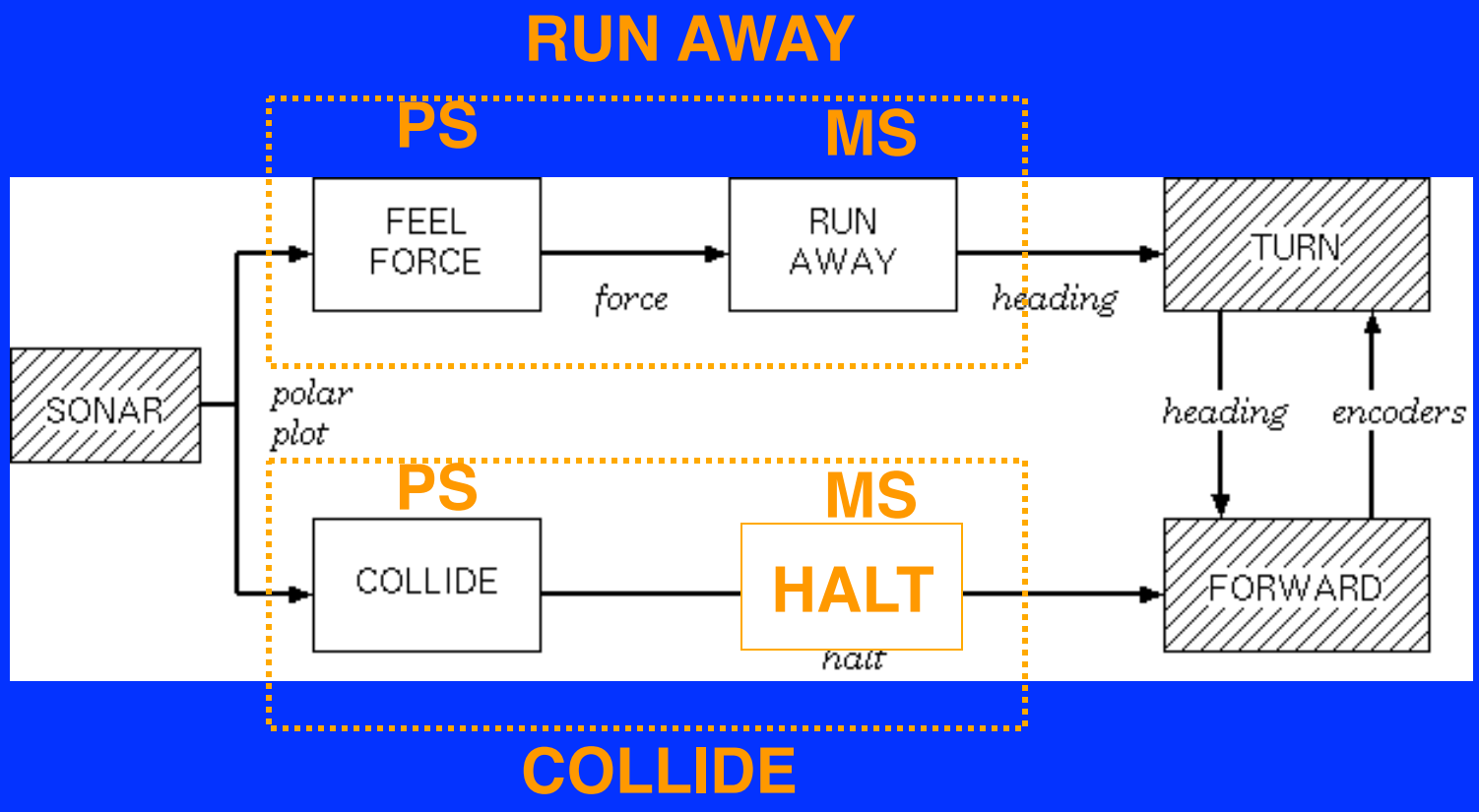


Level 0: Runaway

follow-corridor 2

wander 1

runaway 0

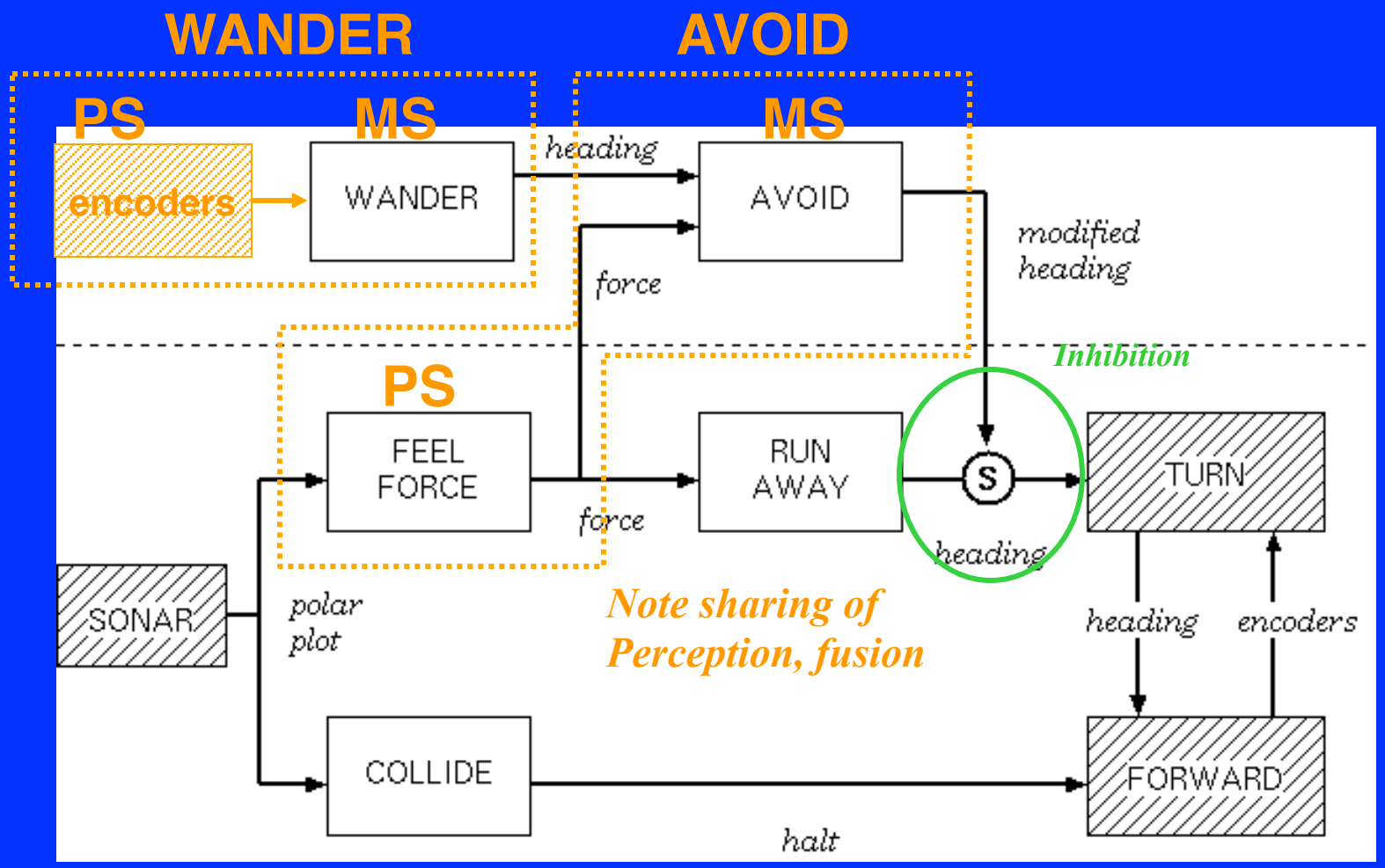


Level 1: Wander

follow-corridor 2

wander 1

runaway 0

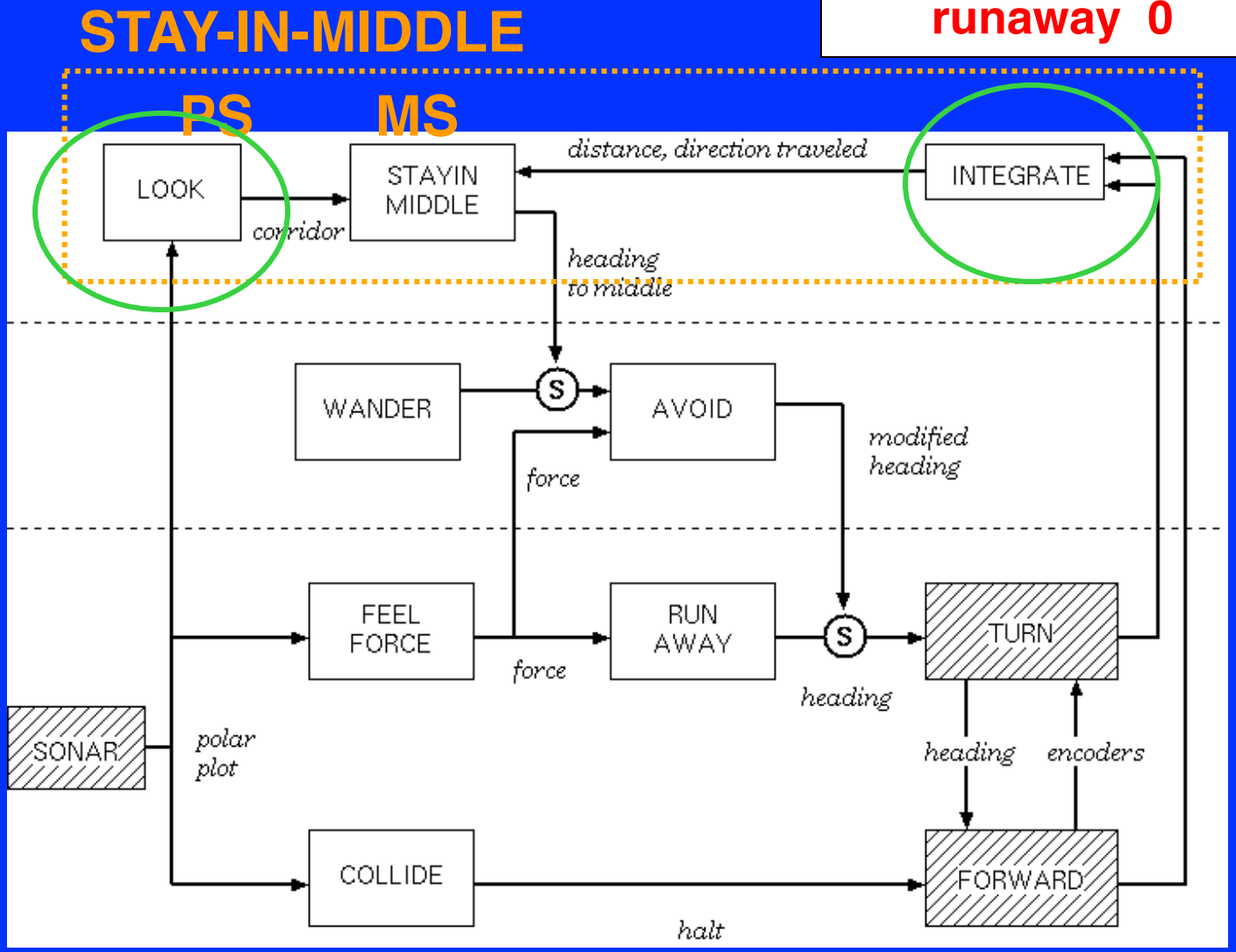


Level 2: Follow-Corridors

follow-corridor 2

wander 1

runaway 0



Subsumption Summary

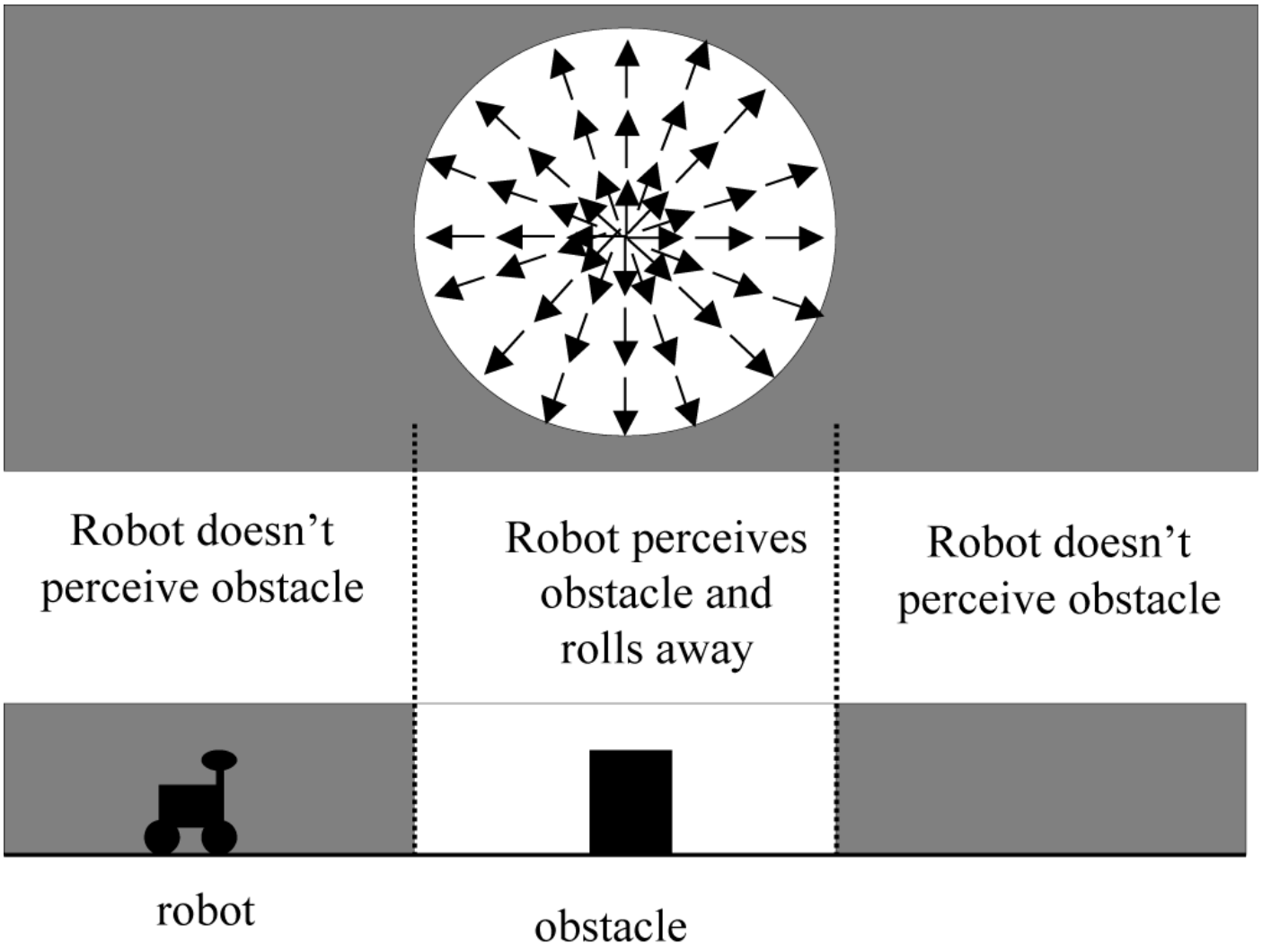
- Many modules operating concurrently at different layers of competence.
- Modules from higher layers of competence can inhibit or suppress other lower level modules.
- Higher level modules can be added to the system without removing or modifying lower-level modules.
- No single world model is maintained; each module can draw from the outputs of different sensors and modules.

Potential Fields: R. Arkin (G. Tech)

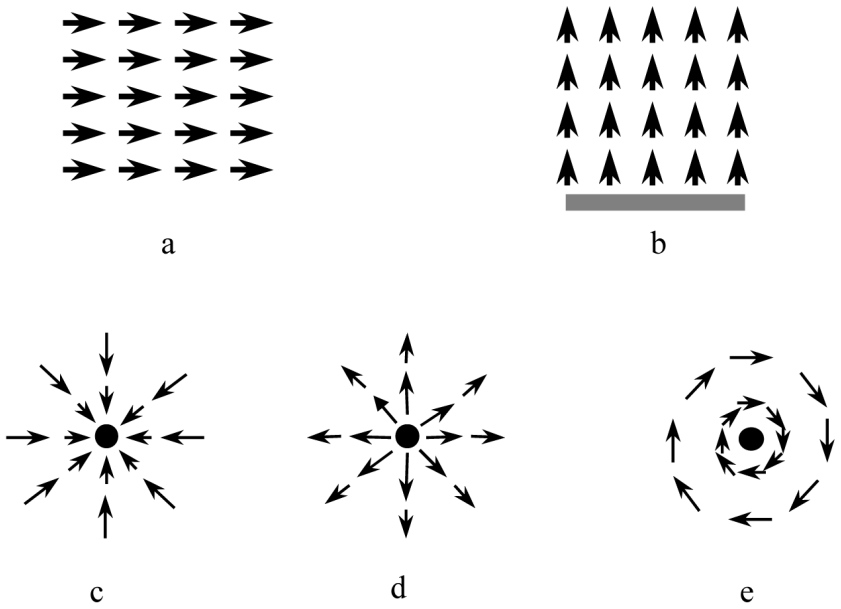
Potential Fields Philosophy

- The motor schema component of a behavior can be expressed with a potential fields methodology
  - A potential field can be a "primitive" or constructed from primitives which are summed together
  - The output of behaviors are combined using vector summation
- From each behavior, the robot "feels" a vector or force
  - Magnitude = force, strength of stimulus, or *velocity*
  - Direction
- But we visualize the "force" as a field, where every point in space represents the vector that it would feel if it were at that point

Run Away via Repulsion

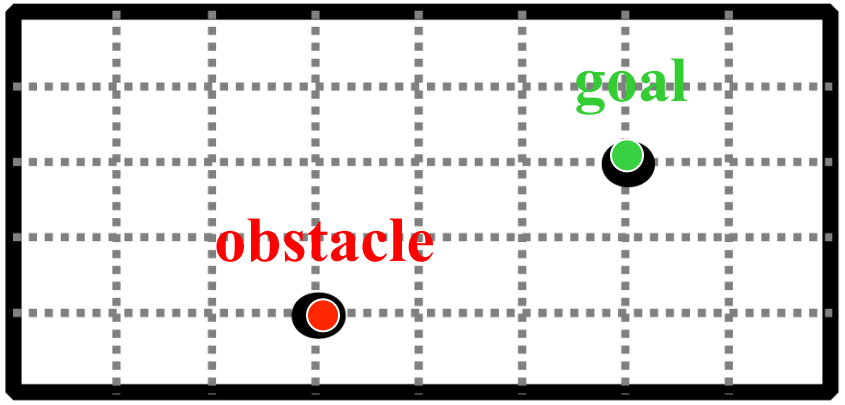


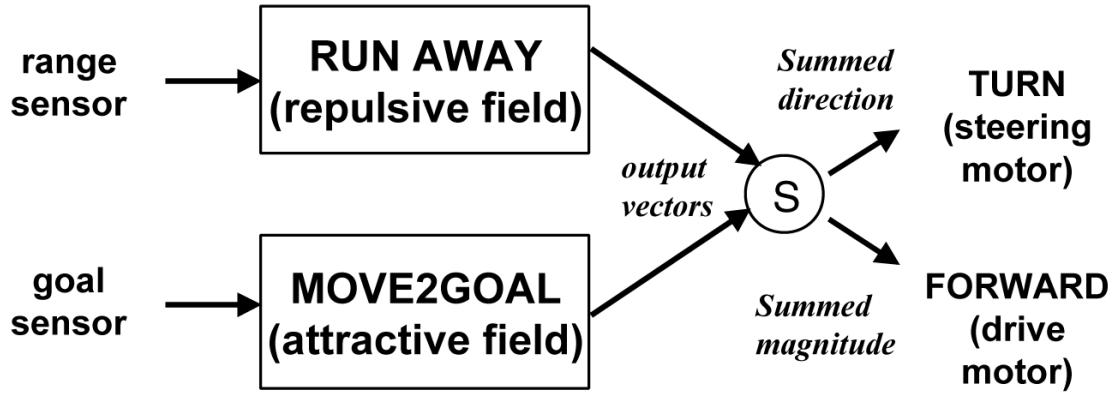
5 Primitive Potential Fields



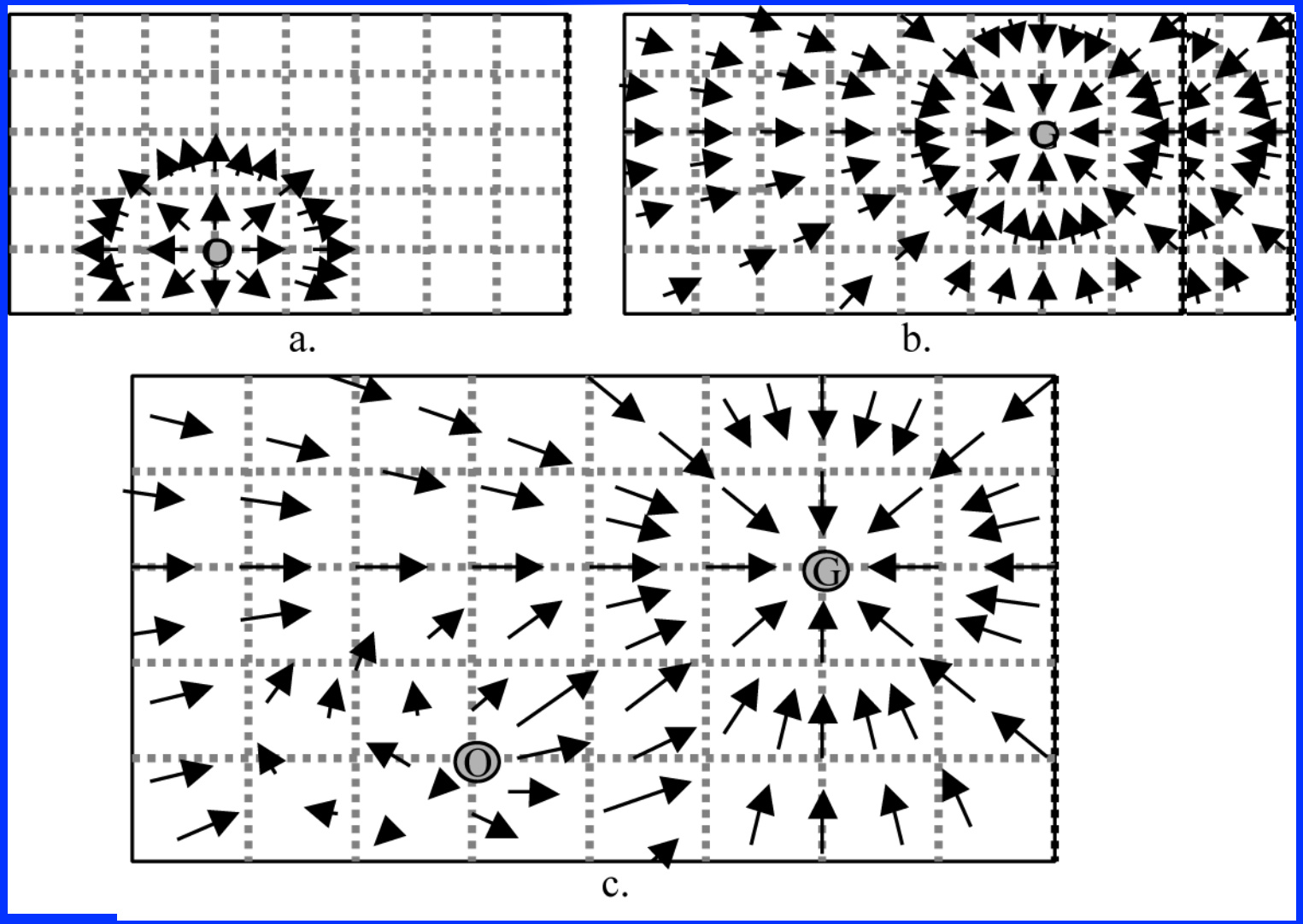
- Uniform
  - Move in a particular direction, corridor following
- Repulsion
  - Runaway (obstacle avoidance)
- Attraction
  - Move to goal
- Perpendicular
  - Corridor following
- Tangential
  - Move through door, docking

Combining Fields for Emergent Behavior



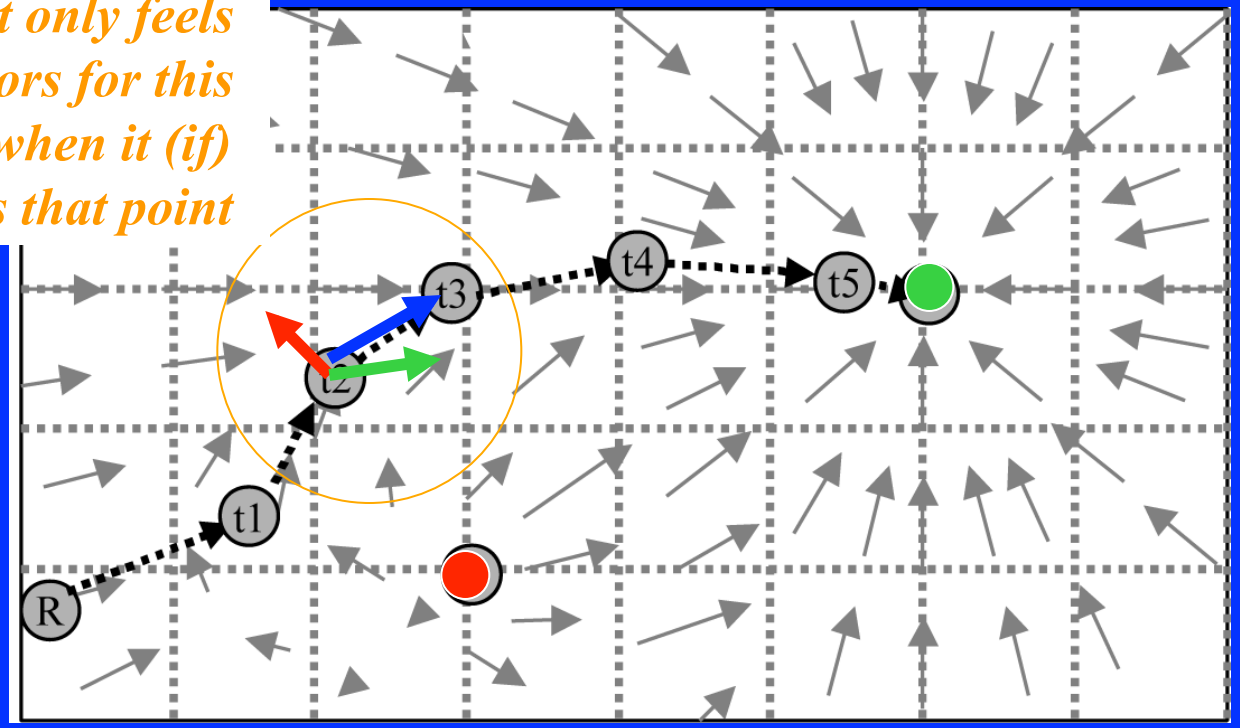


Fields and Their Combination



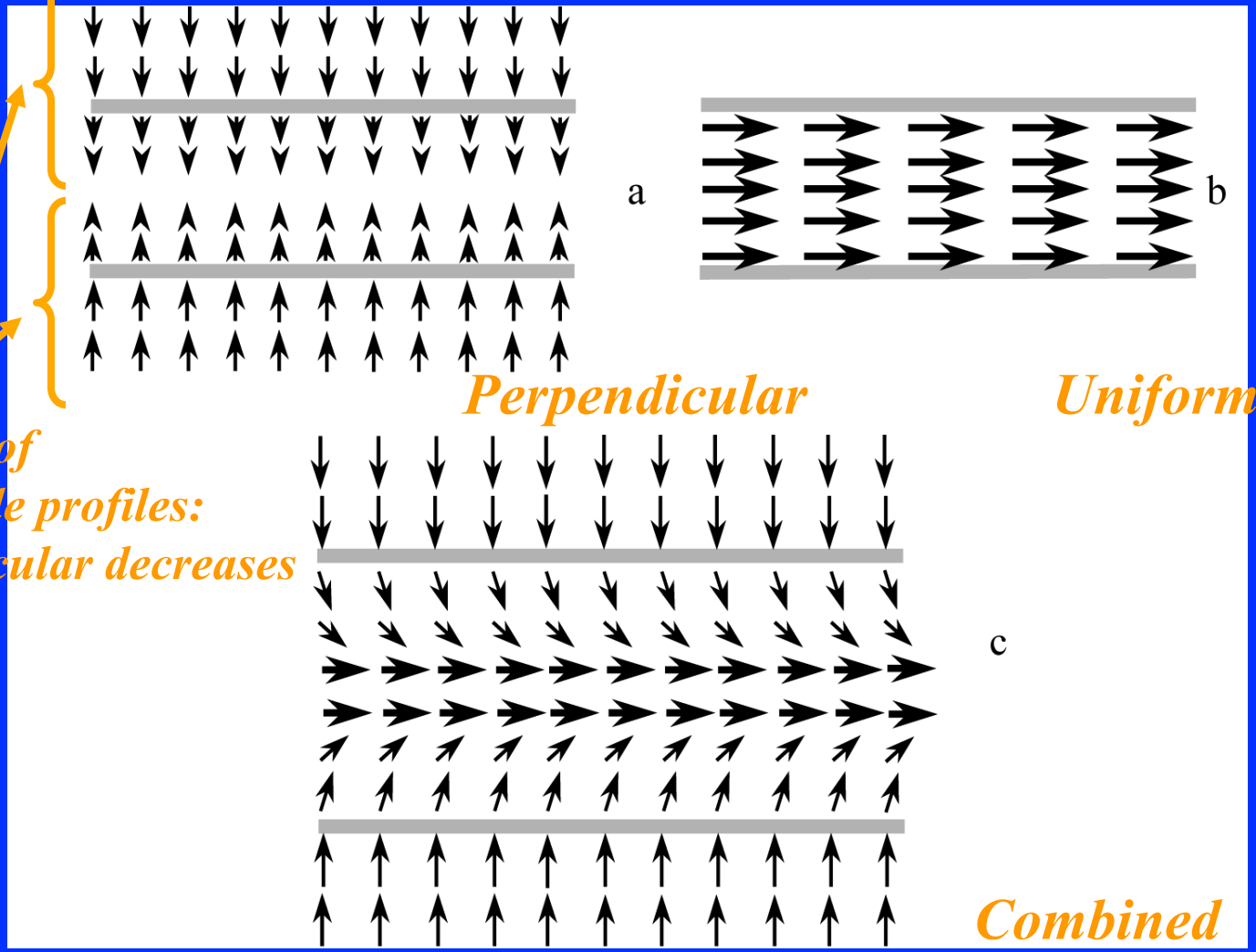
Path Taken

Robot only feels vectors for this point when it (if) reaches that point

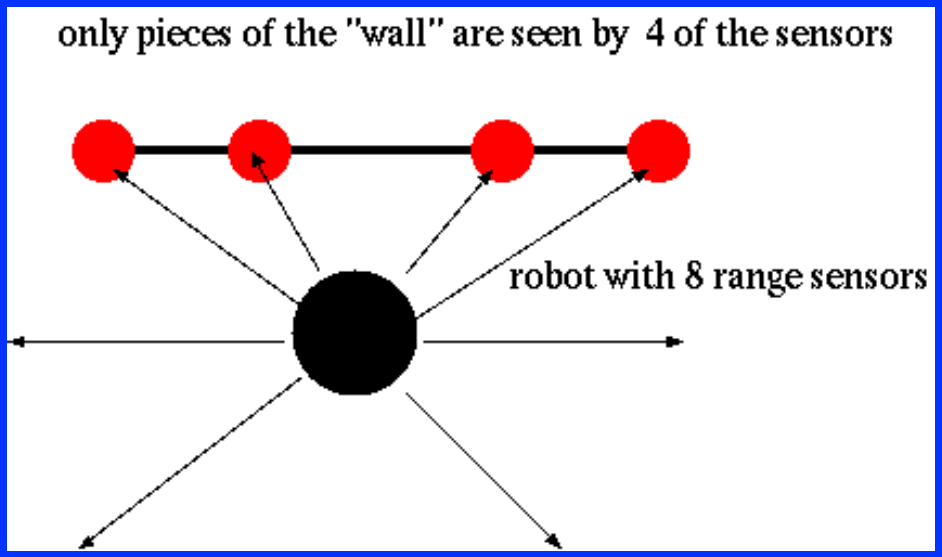


- If robot started at this location, it would take the following path
- It would only "feel" the vector for the location, then move accordingly, "feel" the next vector, move, etc.
- Pfield visualization allows us to see the vectors at all points, but robot never computes the "field of vectors" just the local vector

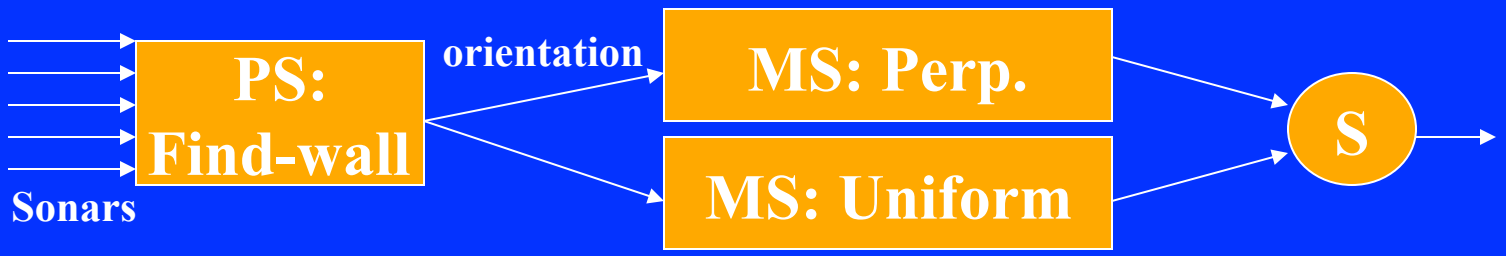
Example: follow-corridor or follow-sidewalk



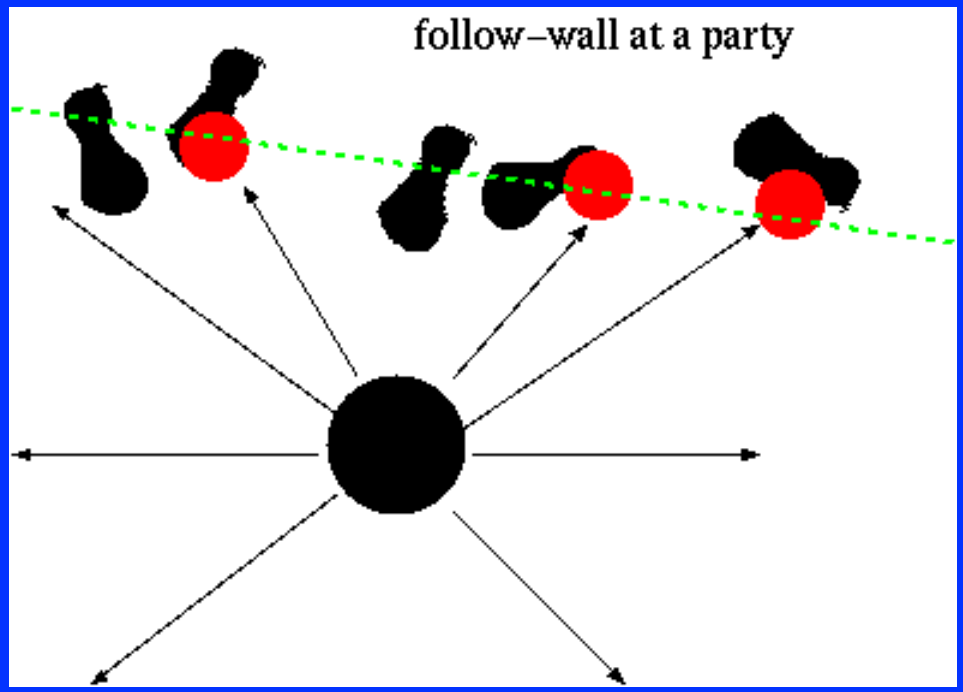
But how does the robot see a wall without reasoning or intermediate representations?



- Perceptual schema "connects the dots", returns relative orientation

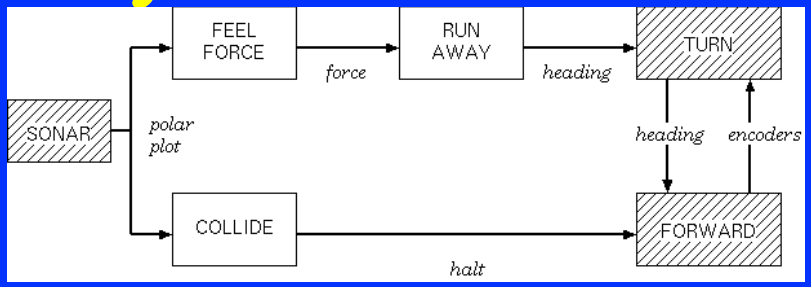


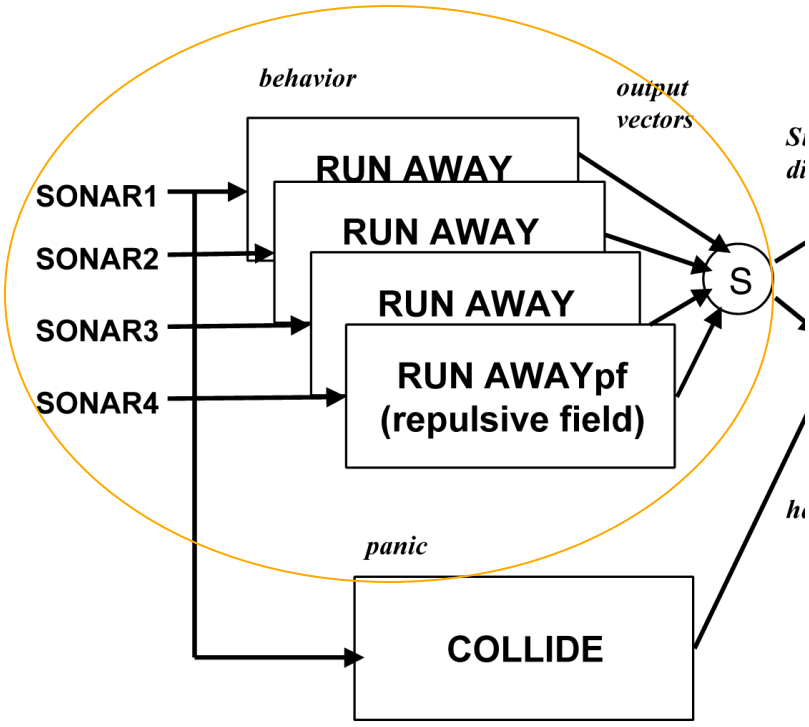
OK, But why isn't that a representation of a wall?



- It's not really *reasoning* that it's a wall, rather it is reacting to the stimulus which happens to be smoothed (common in neighboring neurons)

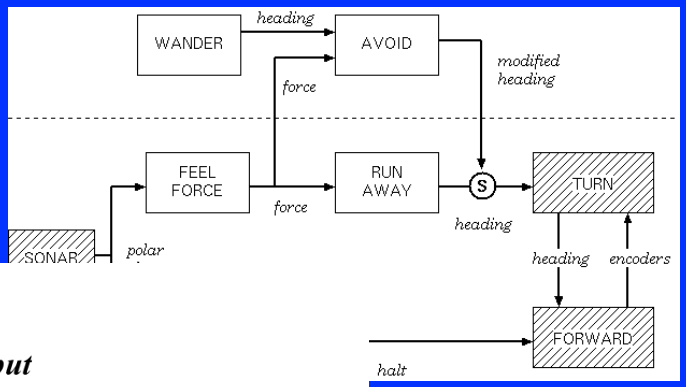
Level 0: Runaway





Note: multiple instances of a behavior vs. 1:
Could just have 1 Instance of RUN AWAY, Which picks nearest reading; Doesn't matter, but this Allows addition of another Sonar without changing the RUN AWAY behavior

Level 1: Wander



Wander is Uniform, but Changes direction aperiodically

WANDERpf (uniform field)

output vectors

RUN AWAYpf

RUN AWAYpf

RUN AWAYpf

RUN AWAYpf (repulsive field)

SONAR1

SONAR2

SONAR3

SONAR4

S

Summed direction

TURN (steering motor)

Summed magnitude

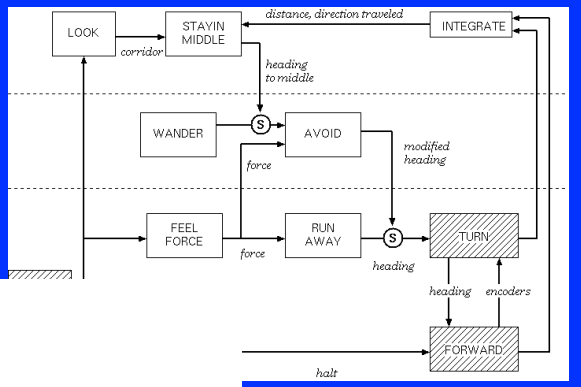
FORWARD (drive motor)

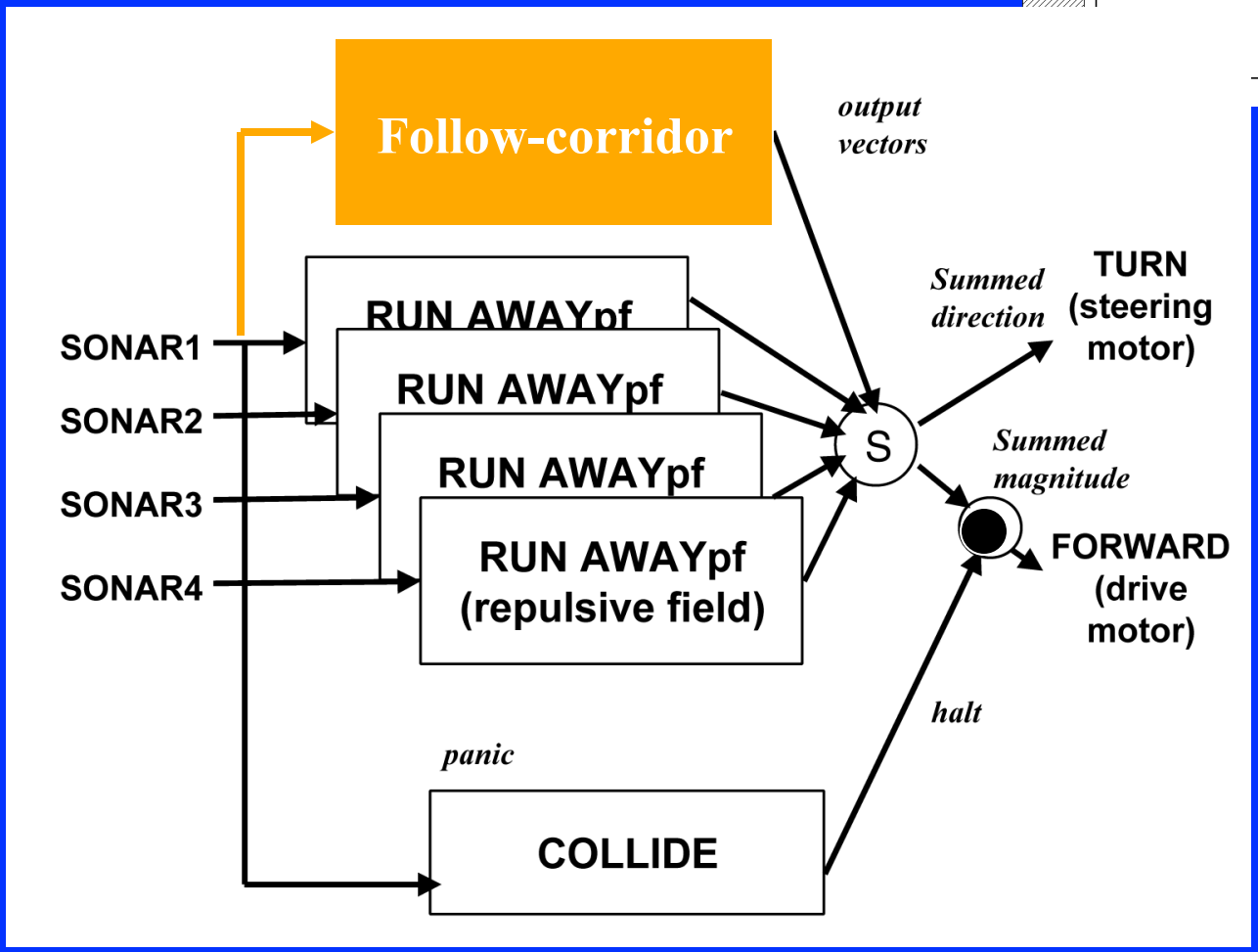
halt

panic

COLLIDE

Level 2: Follow Corridor





Potential Fields Summary

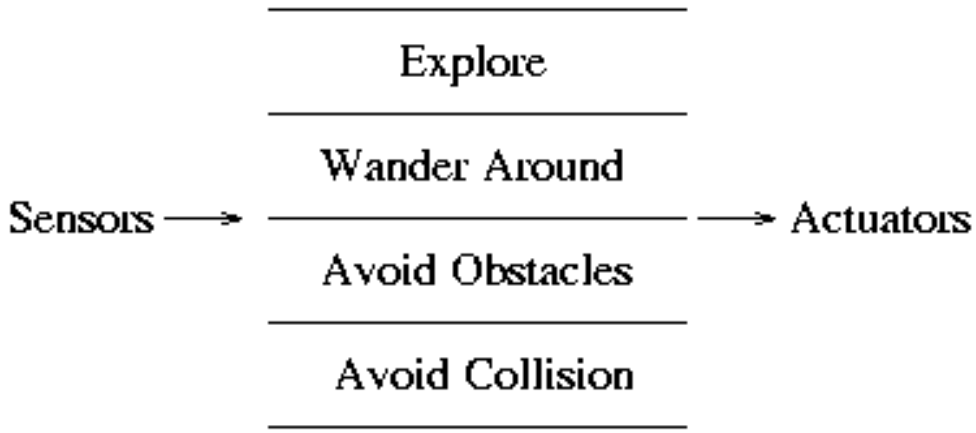
- Advantages
  - Easy to visualize
  - Easy to build up software libraries
  - Fields can be parameterized
  - Combination mechanism is fixed, tweaked with gains
- Disadvantages
  - Local minima problem (sum to magnitude=0)
    - Box canyon problem
  - Jerky motion
- Example video: https://www.youtube.com/watch?v=ka7Yb_XELAU

Reactive Summary

- Reactive Paradigm: SA, sensing is local
  - Eliminates the *frame problem* by not using any global or persistent representation
  - Perception is direct, ego-centric, and distributed
- Two architectural styles are: *subsumption* and *pfields*
- Behaviors in pfield methodologies are a tight coupling of sensing to acting; modules are mapped to schemas conceptually
- Potential fields and subsumption are logically equivalent but different implementations
- Pfield problems include
  - local minima (ways around this)
  - jerky motion
  - bit of an art

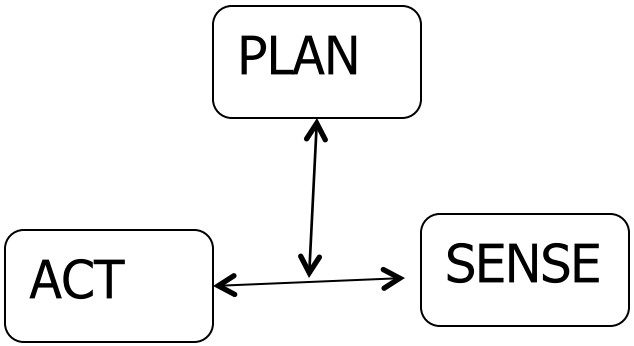
Behavior-based Architecture

- Reactive + state information
- State information allows robot to retain memory of previous actions
- Easily implemented



Hybrid architectures

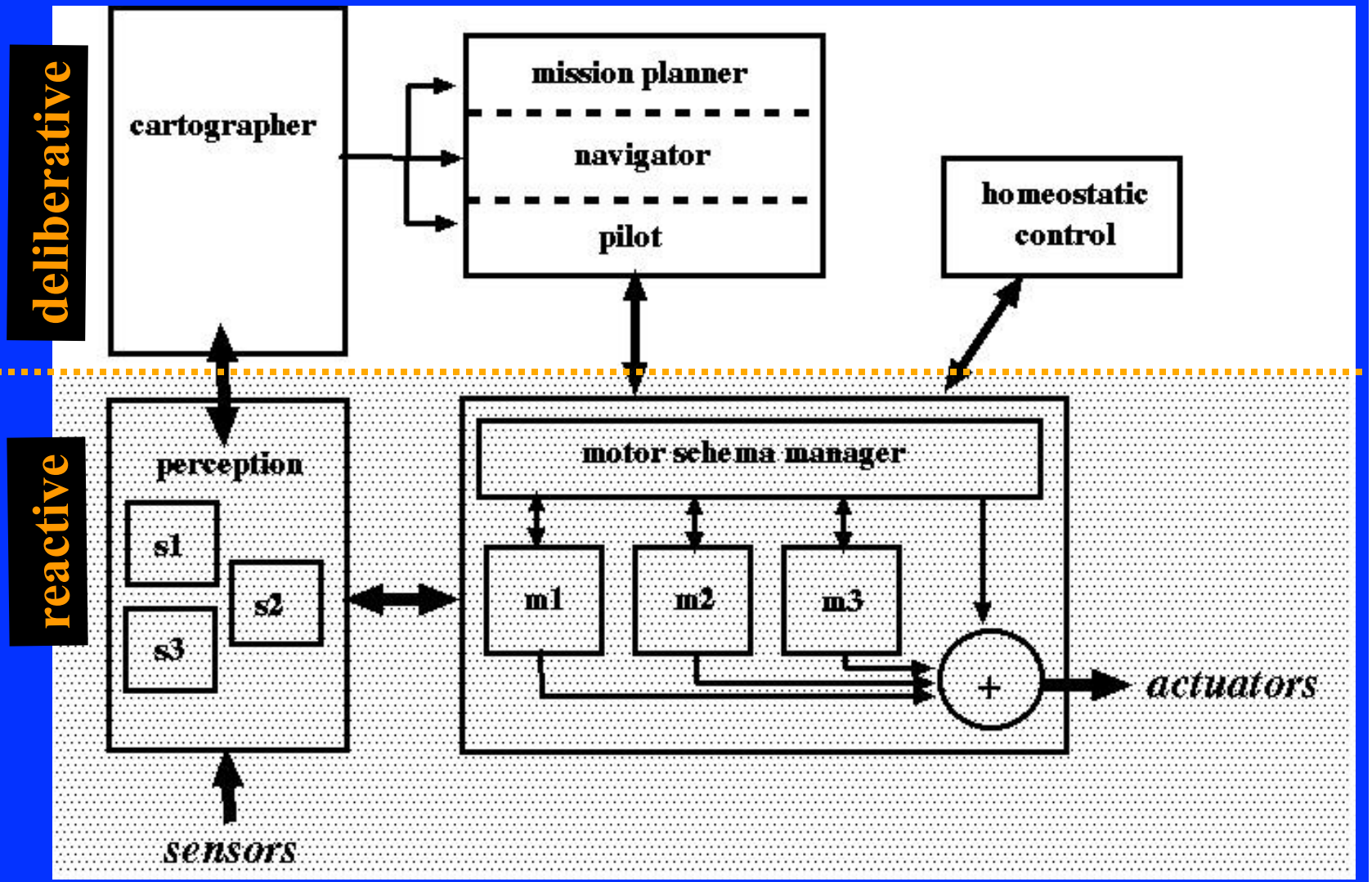
- Includes both deliberative and reactive control components,
- Multi-thread implementation
- Often referred to as three-tiered architecture
- Combines the strengths of both architectures but more complicated to implement
- Used by most real-world robotic systems



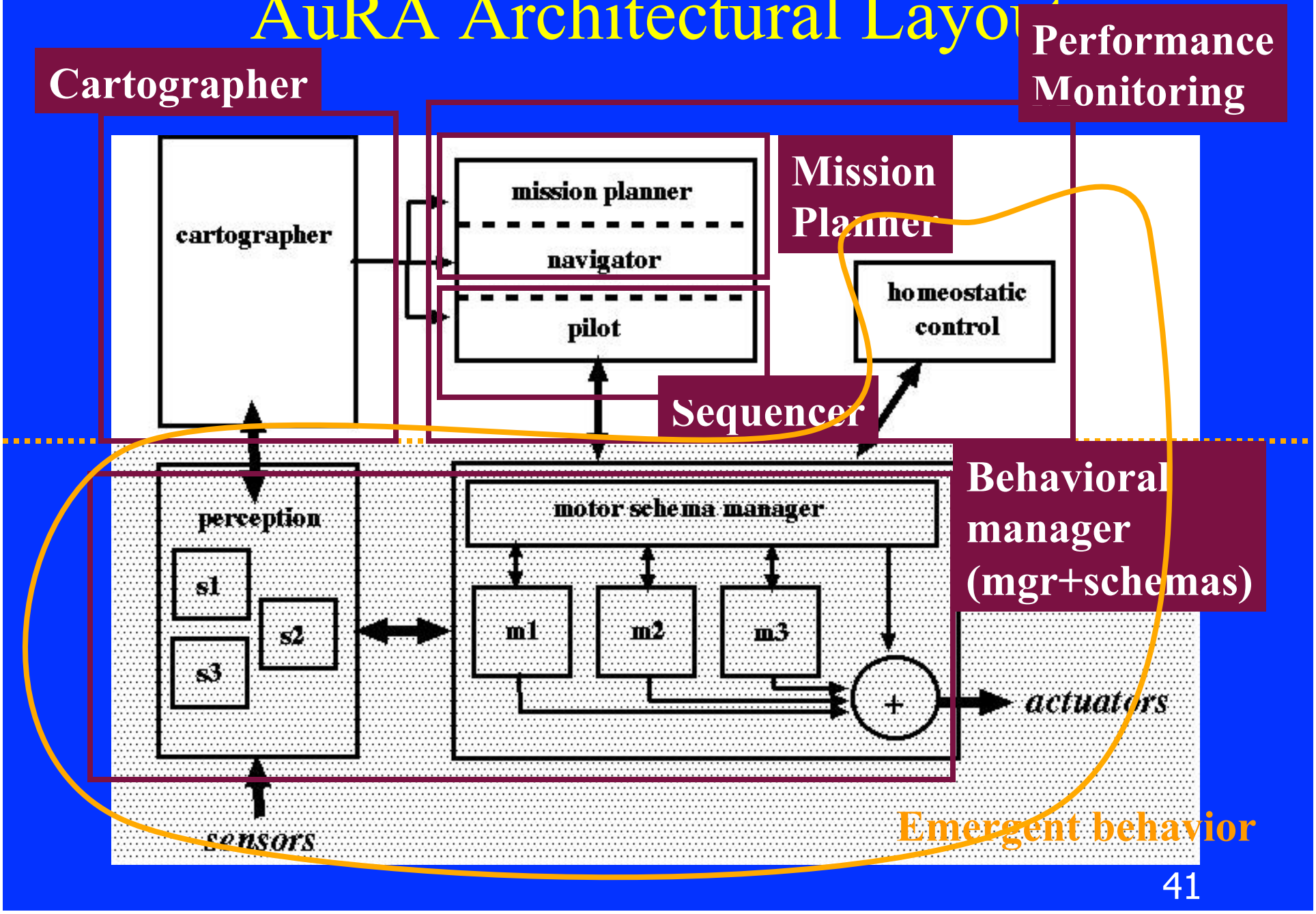
AuRA (Autonomous Robot Arch.)

Ron Arkin, Georgia Institute of Technology
Uses potential fields reactive layer

AuRA Architectural Layout



AuRA Architectural Layout



Summary

- Hybrid architecture generates a complicated interleaved workflow of perception/cognition/ action events.
- Can respond to rapid unforeseen events while continuing to complete a mission plan.
- Used by most real-world robotic systems
- Slides courtesy of Robin Murphy, Intro to AI Robotics