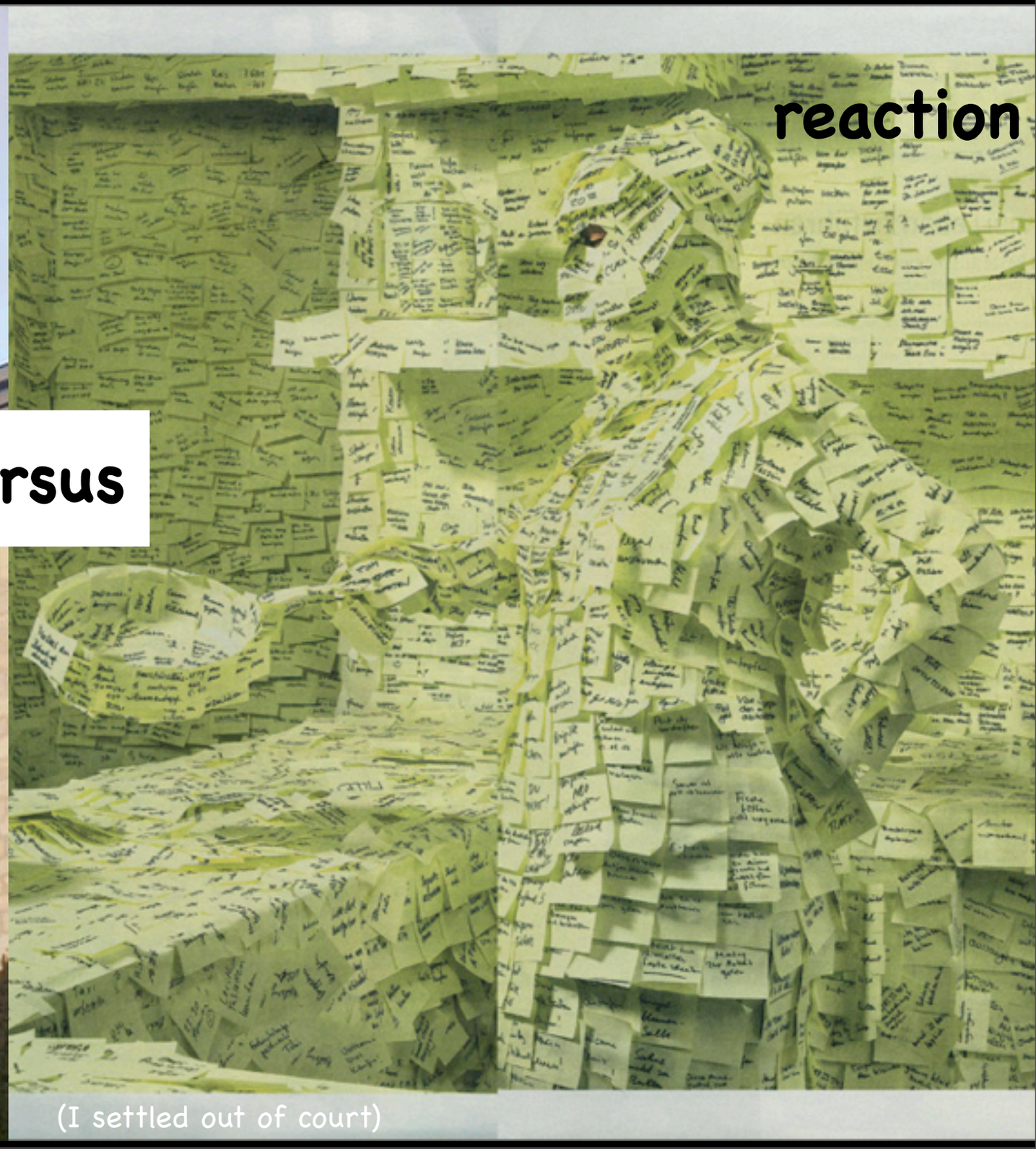
Topic 4:
Autonomous Control Architectures

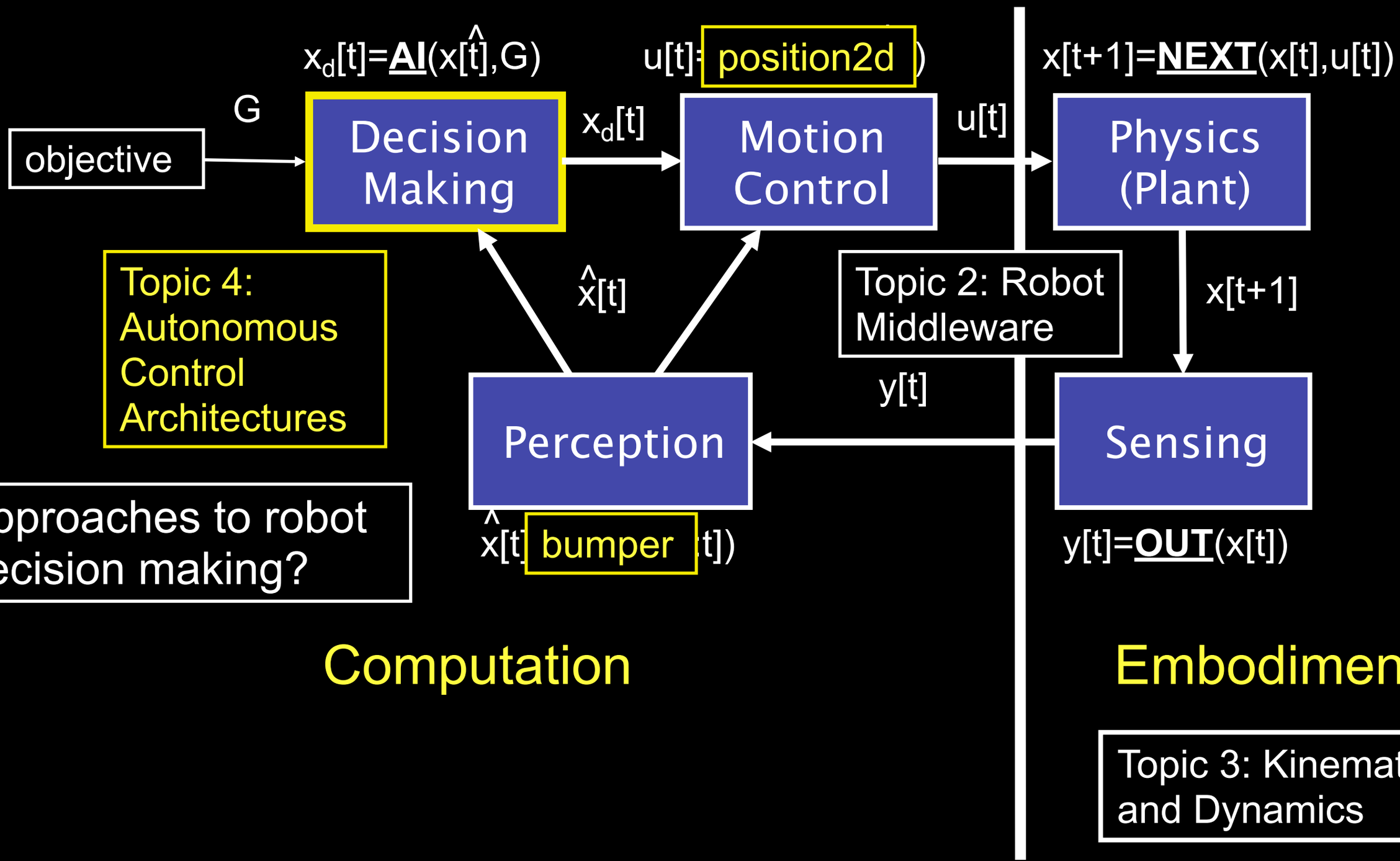
versus



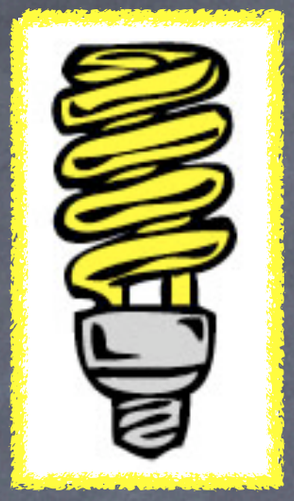
robot control loop

- someone please sketch on the board

The Robot Control Loop



Should your robot's decision making

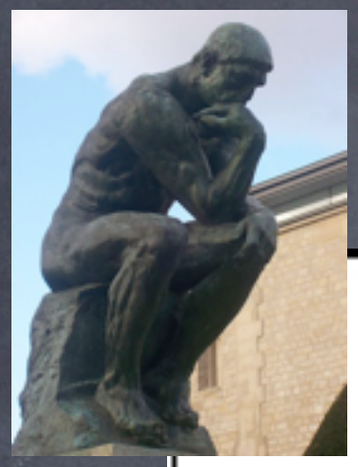


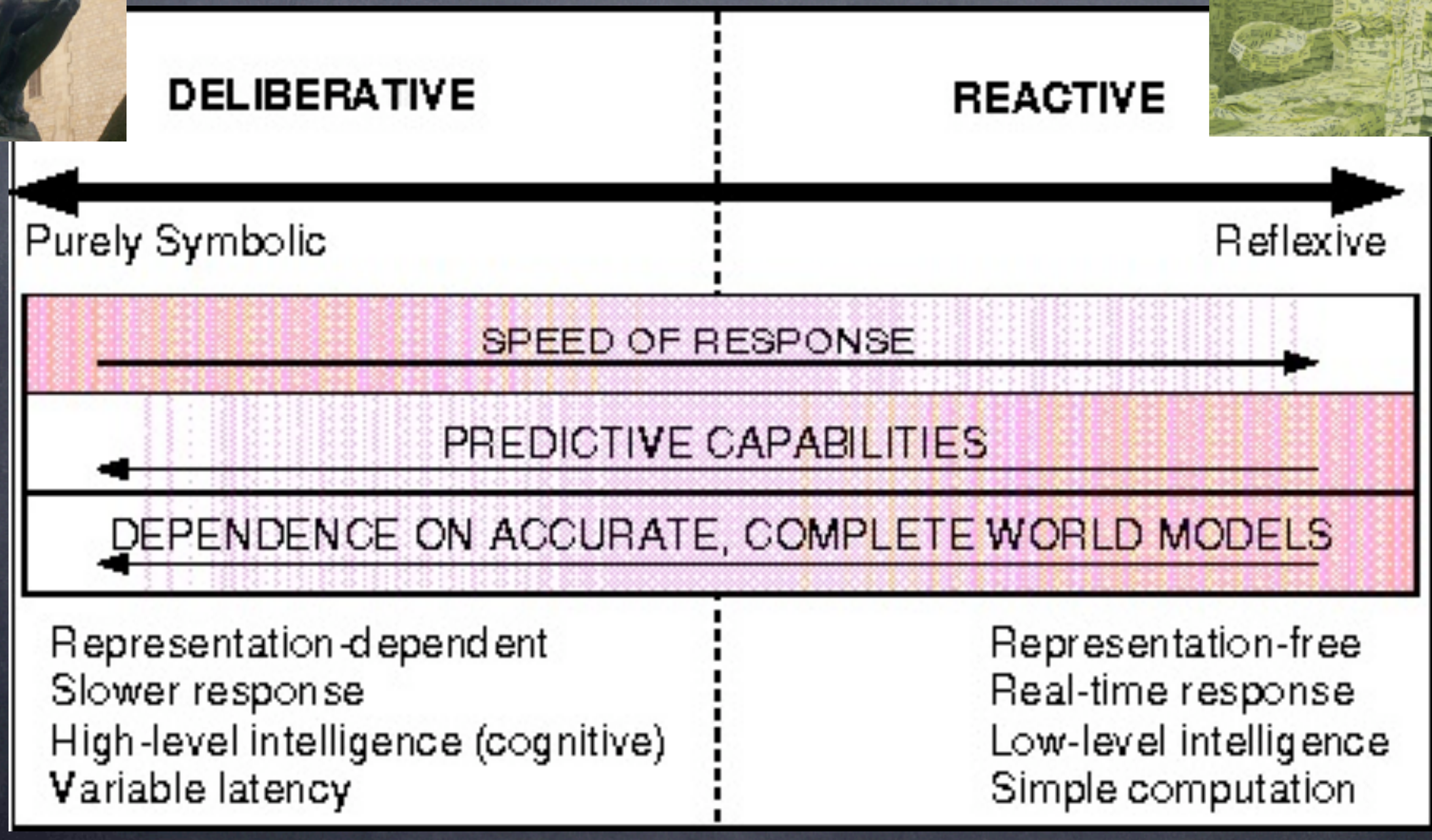
OR

fully think through solving a problem?

react quickly to changes in its world?

Deliberation-Reaction spectrum



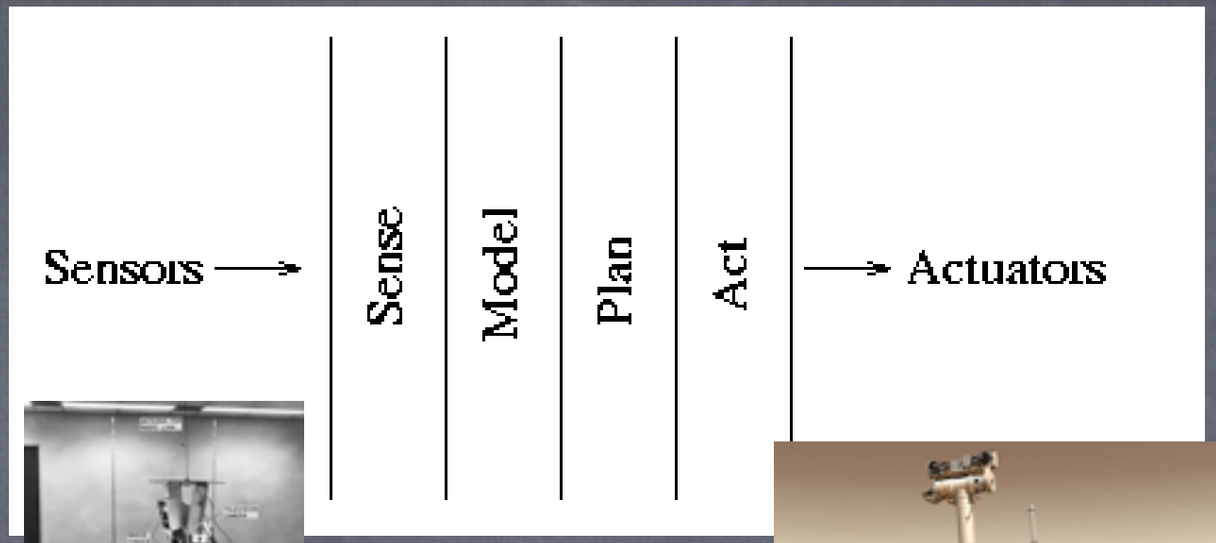


Considerations: time-scale/tractability, generality, representation/state

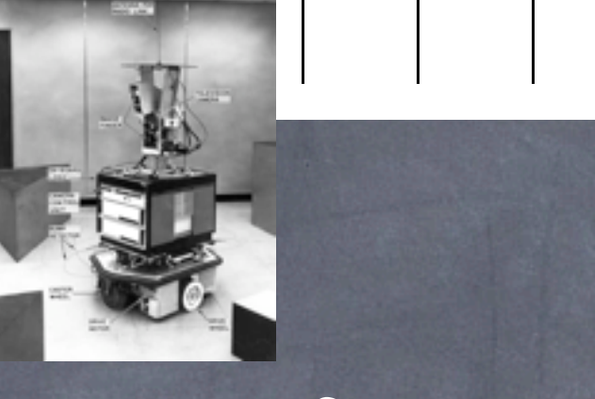
Types of robot policies

- Deliberative (Planner-based) Control
  - "Think hard, act later."
- Reactive Control
  - "Don't think, (re)act."
- Hybrid Control
  - "Think and act separately & concurrently."
- Behavior-Based Control
  - "Think the way you act."

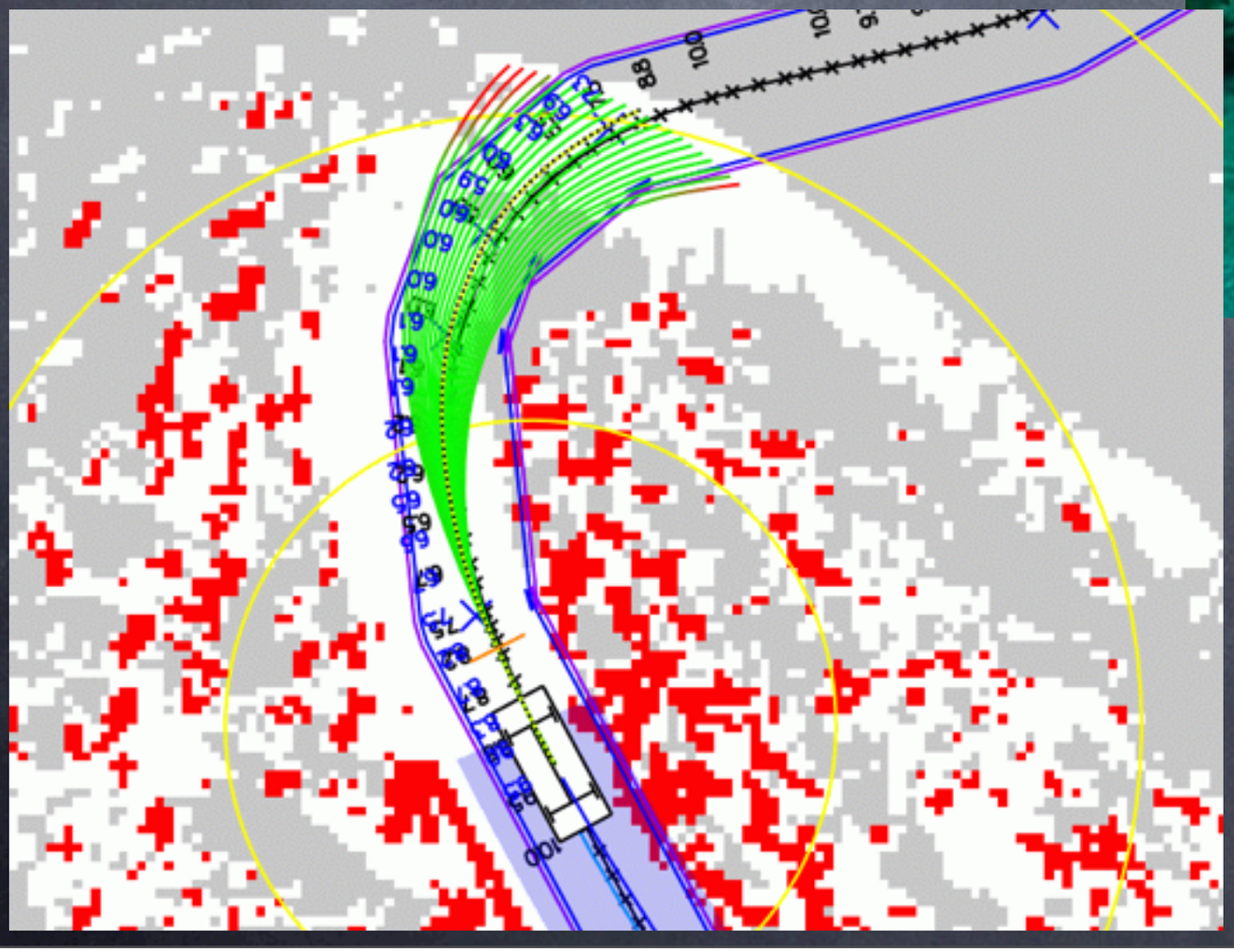
Deliberation



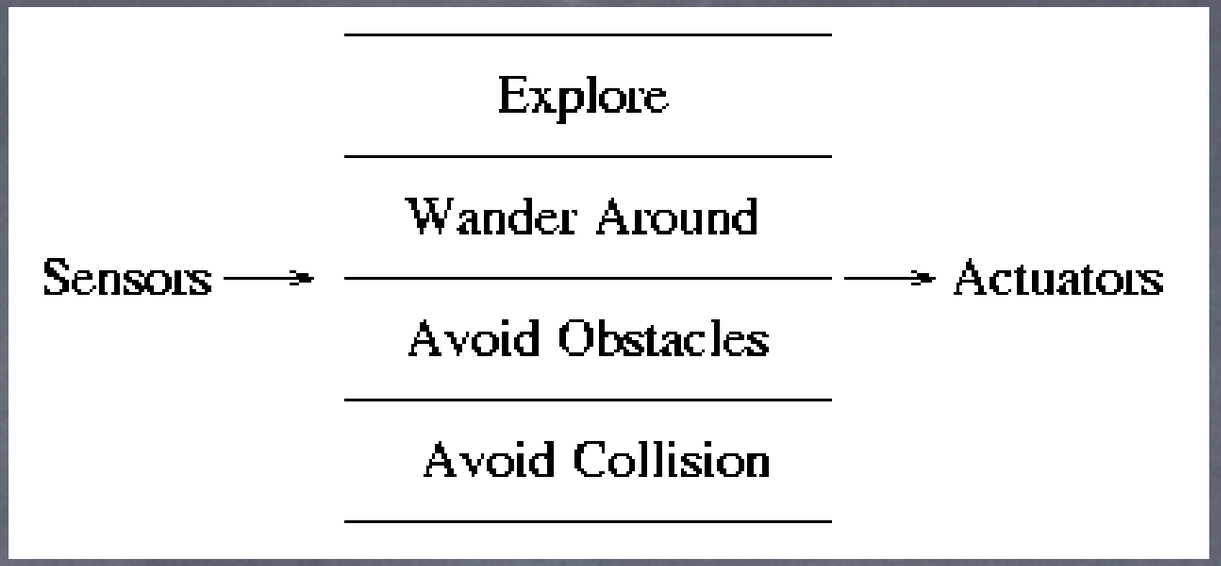
- "sense-plan-act" paradigm
- sense: build most complete model of world
- plan: search over all possible outcomes
- act: execute plan through motor forces
- BFS, DFS, Dijkstra, A*
- Shakey, Grand Challenge
  - GPS waypoints



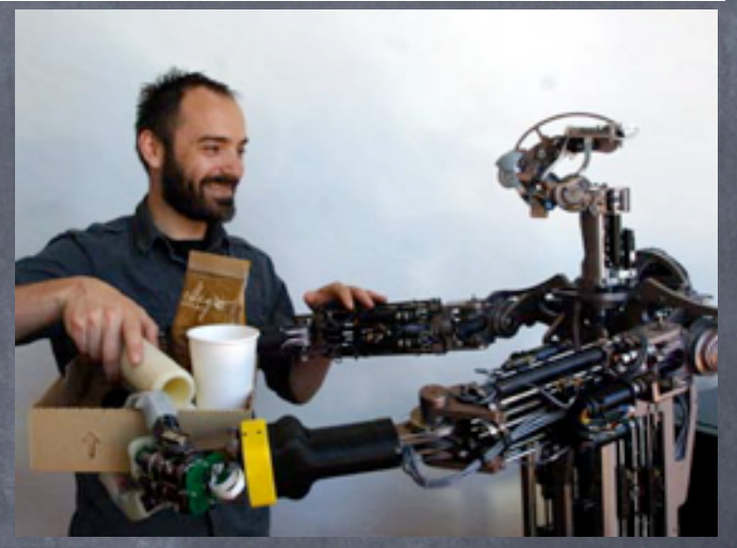
Stanley (Grand Challenge)



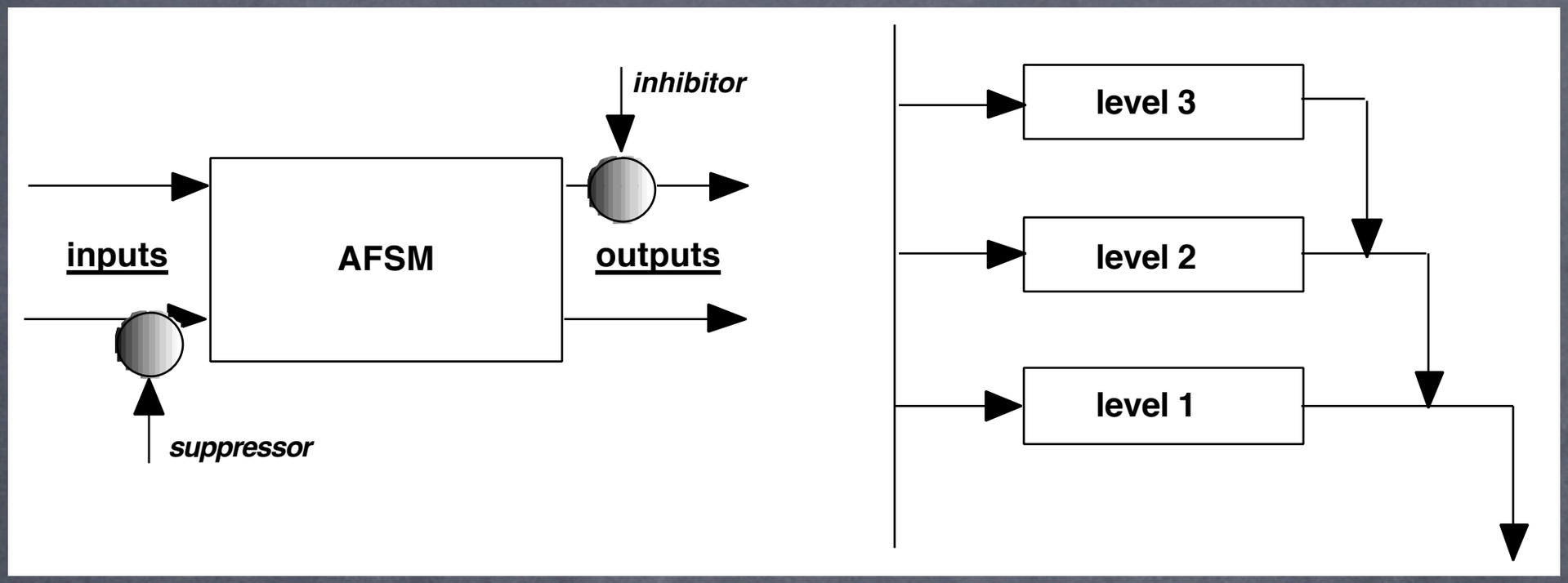
Reaction



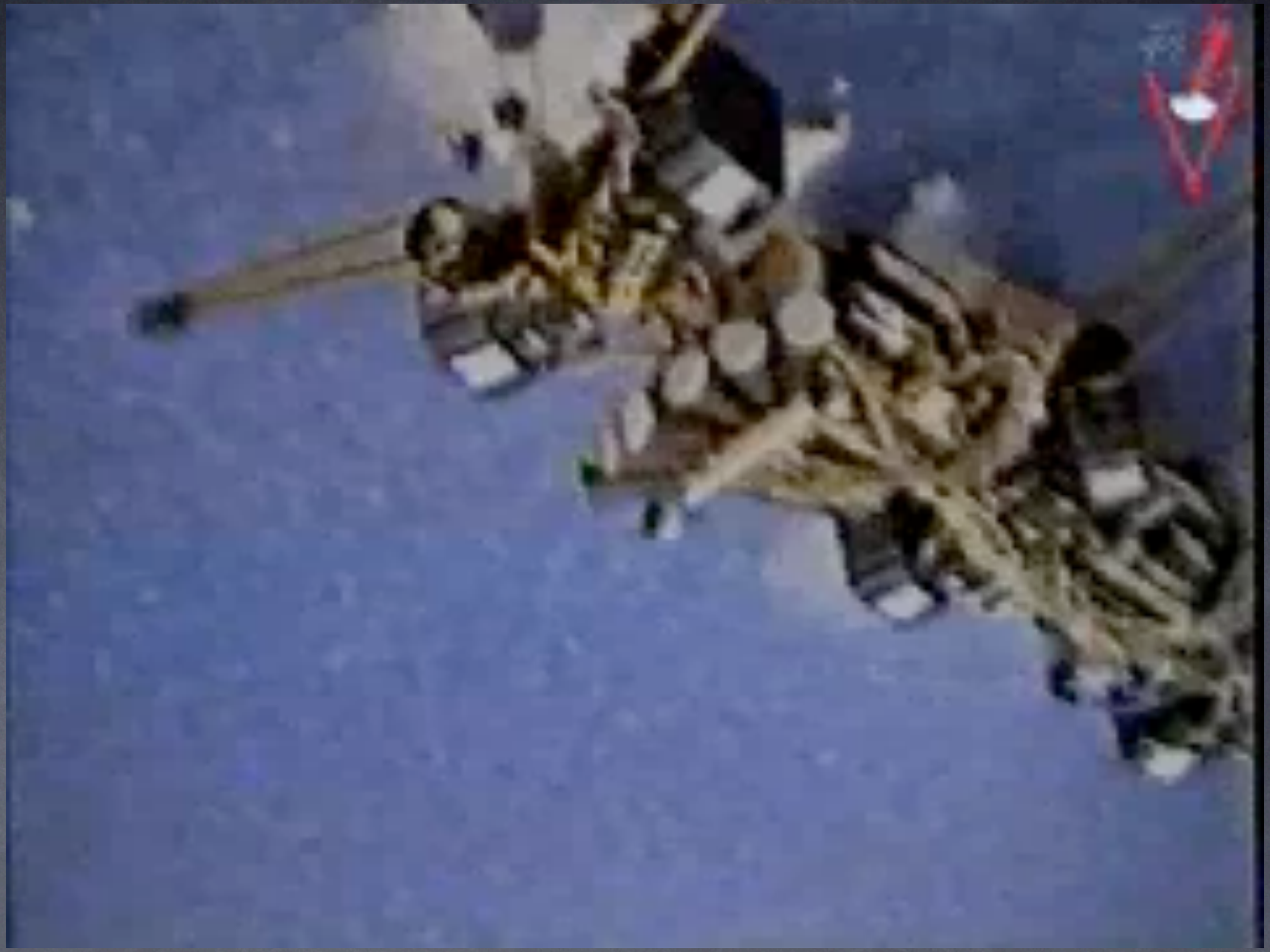
- No representation of state
- Typically, fast hardcoded rules
- Embodied intelligence
  - behavior <- control + embodiment
  - ant analogy, stigmergy
- Subsumption architecture
  - prioritized reactive policies
- Ghengis hexpod video



Ghengis videos

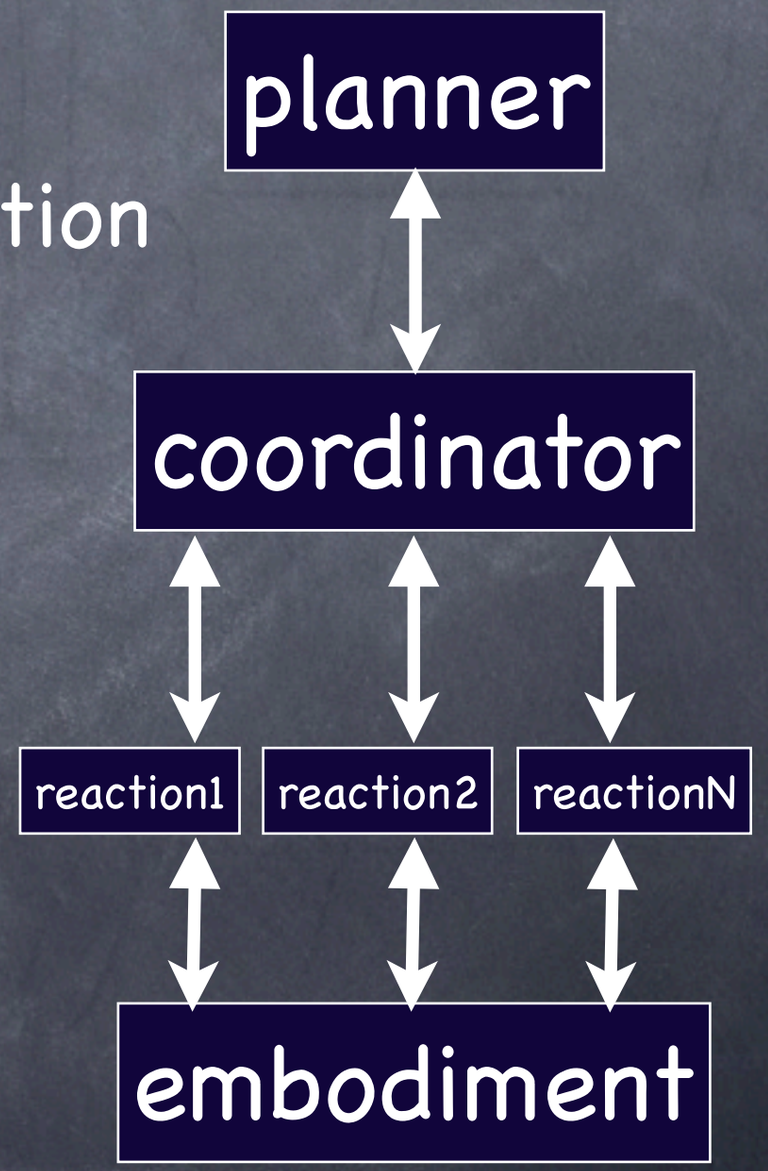


- http://www.youtube.com/watch?v=BUxFfv9JimU
- http://www.youtube.com/watch?v=RKeBl0-msGQ



Hybrid systems

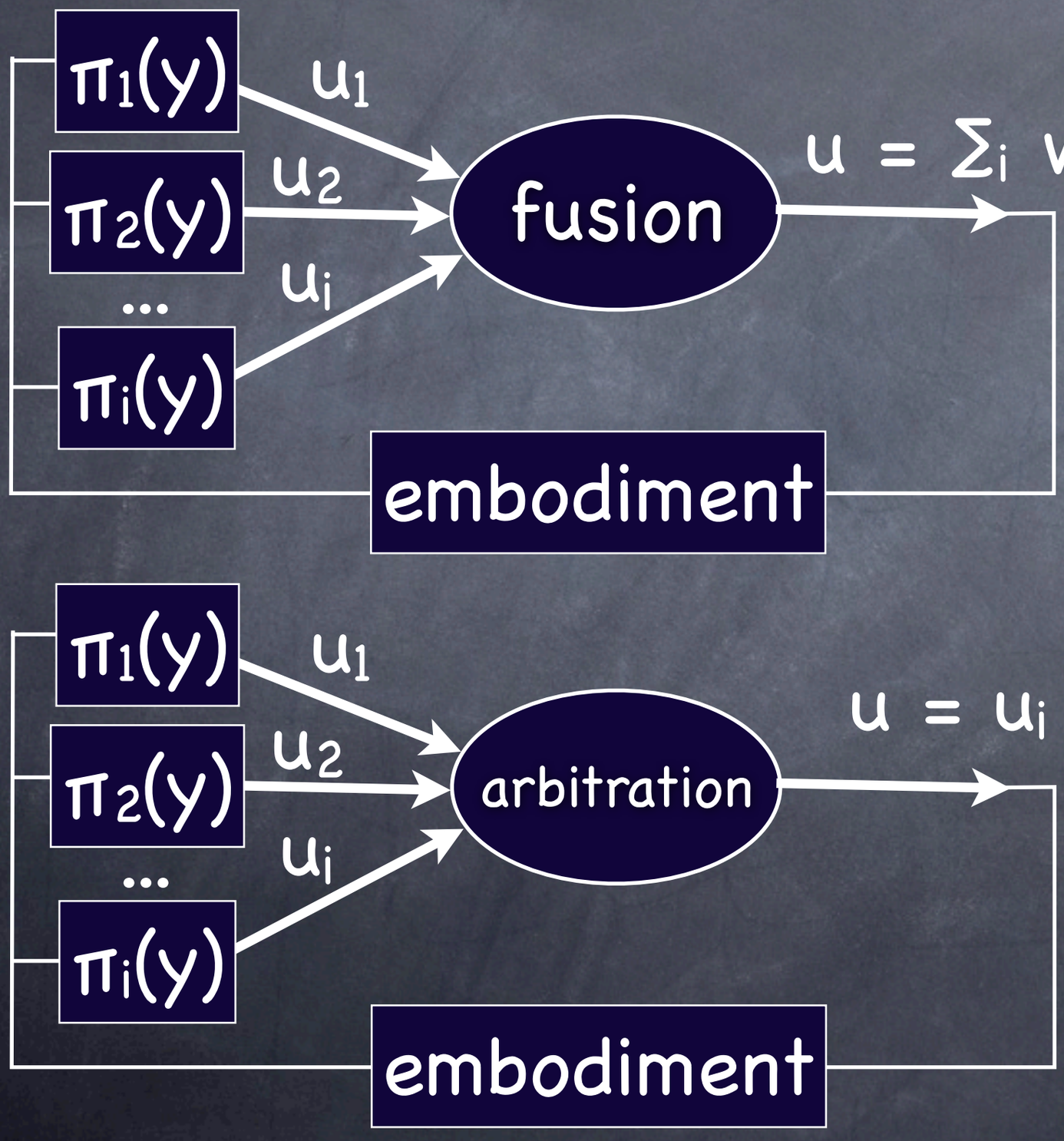
- Top-down planner for high-level goals
- Reaction for low-level immediate execution
- Interface layer to coordinate
  - how to balance long and short term
- modern cost maps?
- Not discrete-continuous hybrid control



Behavior-based systems

- All modules have equal priority, access to sensing, and output motor commands
- Modules can be deliberative, reactive, or whatever
- Commands merged through arbitration or fusion
- potential fields?

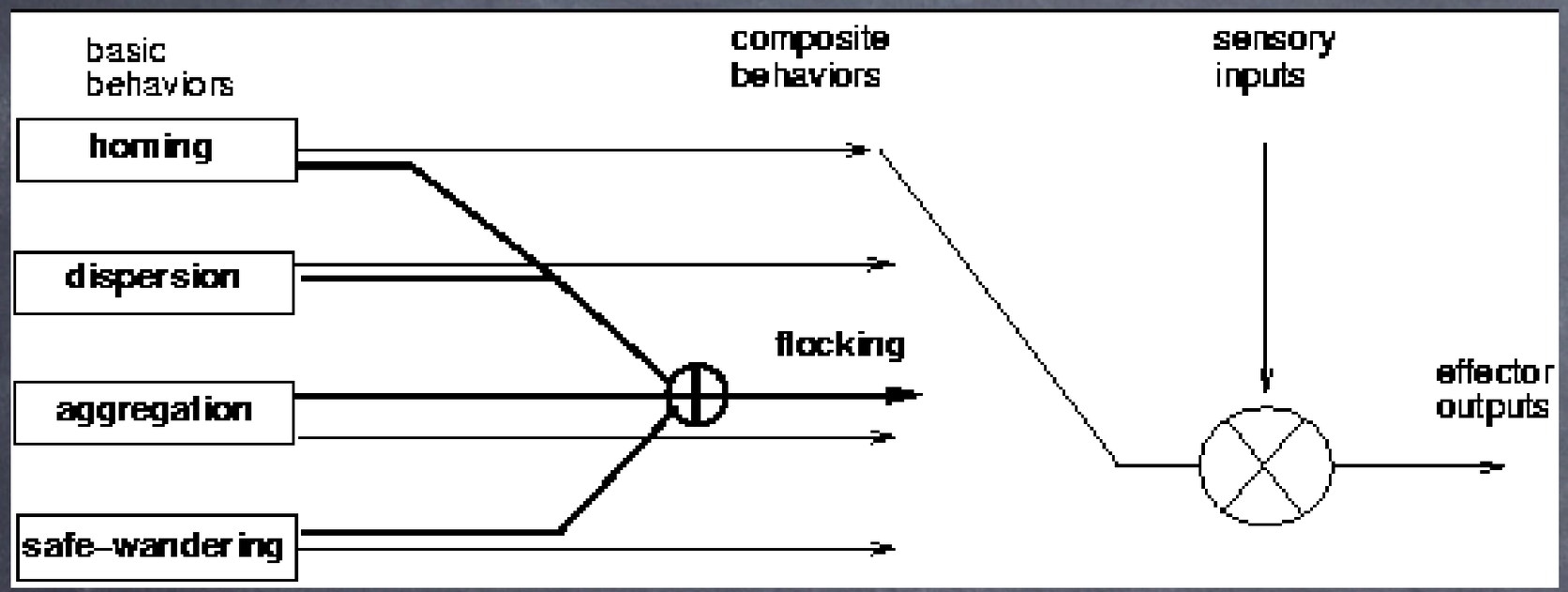
Arbitration vs. Fusion

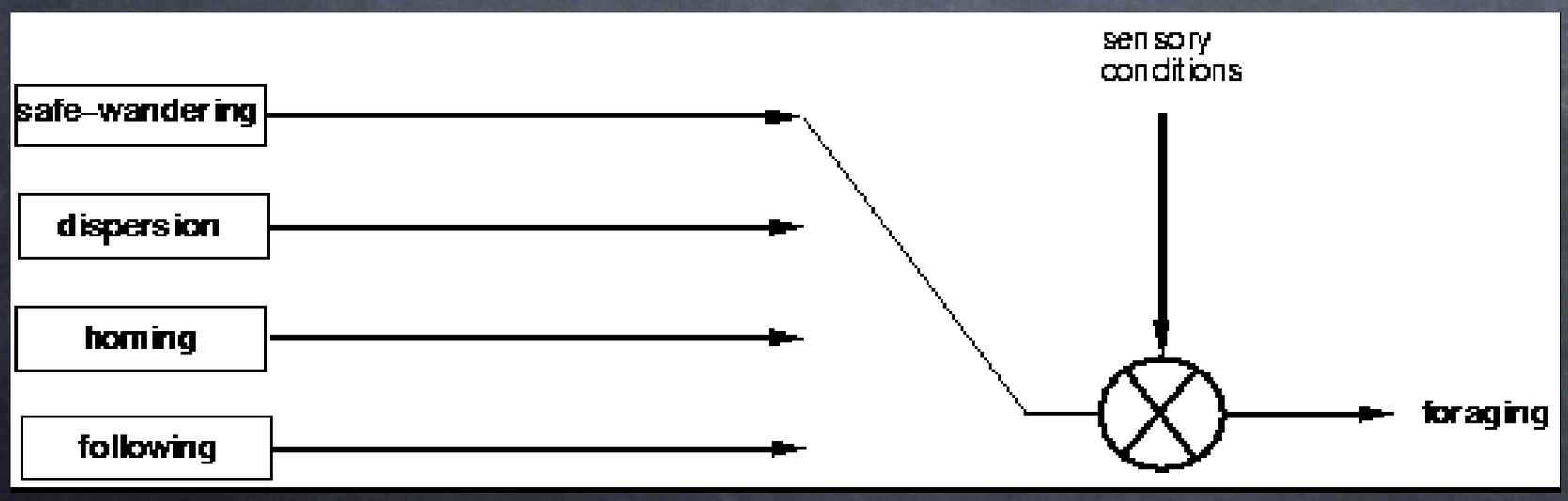


Fusion:
linearly combine
$u = \Sigma_i\ w_i\ u_i$

Arbitration:
winner-take-all
$u = u_i \mid \text{argmax}_i\ w_i$

Arbitration vs. Fusion





Fusion:
linearly combine
$u = \sum_i w_i u_i$

Arbitration:
winner-take-all
$u = u_i \mid \text{argmax}_i\ w_i$

Navigation

- get from point A to point B
- what is the simplest policy to perform nav?

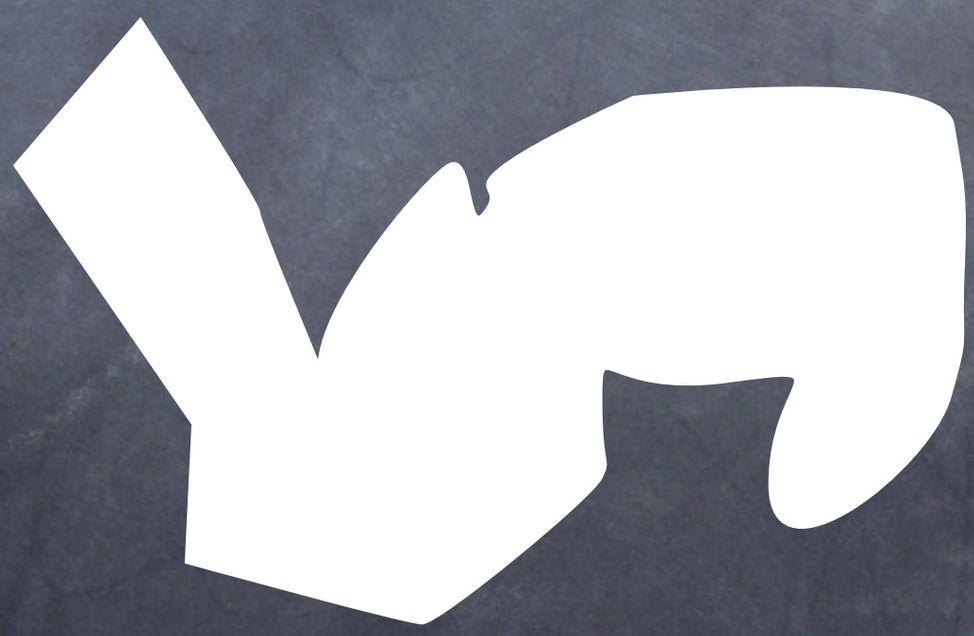
Navigation

- get from point A to point B
- what is the simplest policy to perform nav?
  - random walk (Enclosure Escape project)
  - reactive: embodied intelligence
- what is a "simple" deliberative policy?

Bug Algorithm

- Assume bumper/touch and localization or goal recognition
- show video

Goal



Start

Bug Algorithm

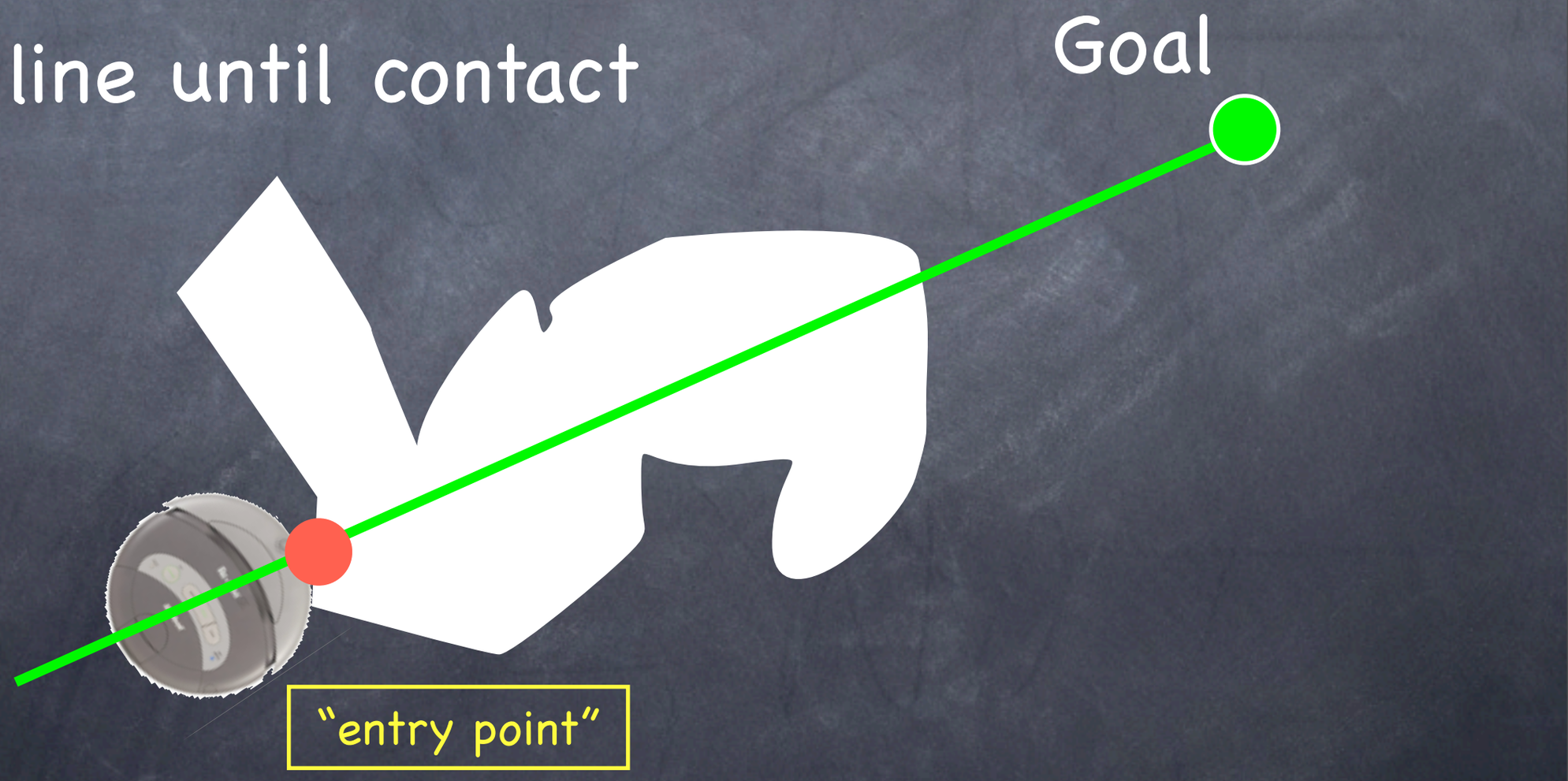
- Assume bumper/touch and localization or goal recognition

Draw straight-line path to goal

Bug Algorithm

- Assume bumper/touch and localization or goal recognition

Follow line until contact



Bug Algorithm

- Assume bumper/touch and localization or goal recognition

Follow boundary around obstacle

Goal

termed "wall-following"

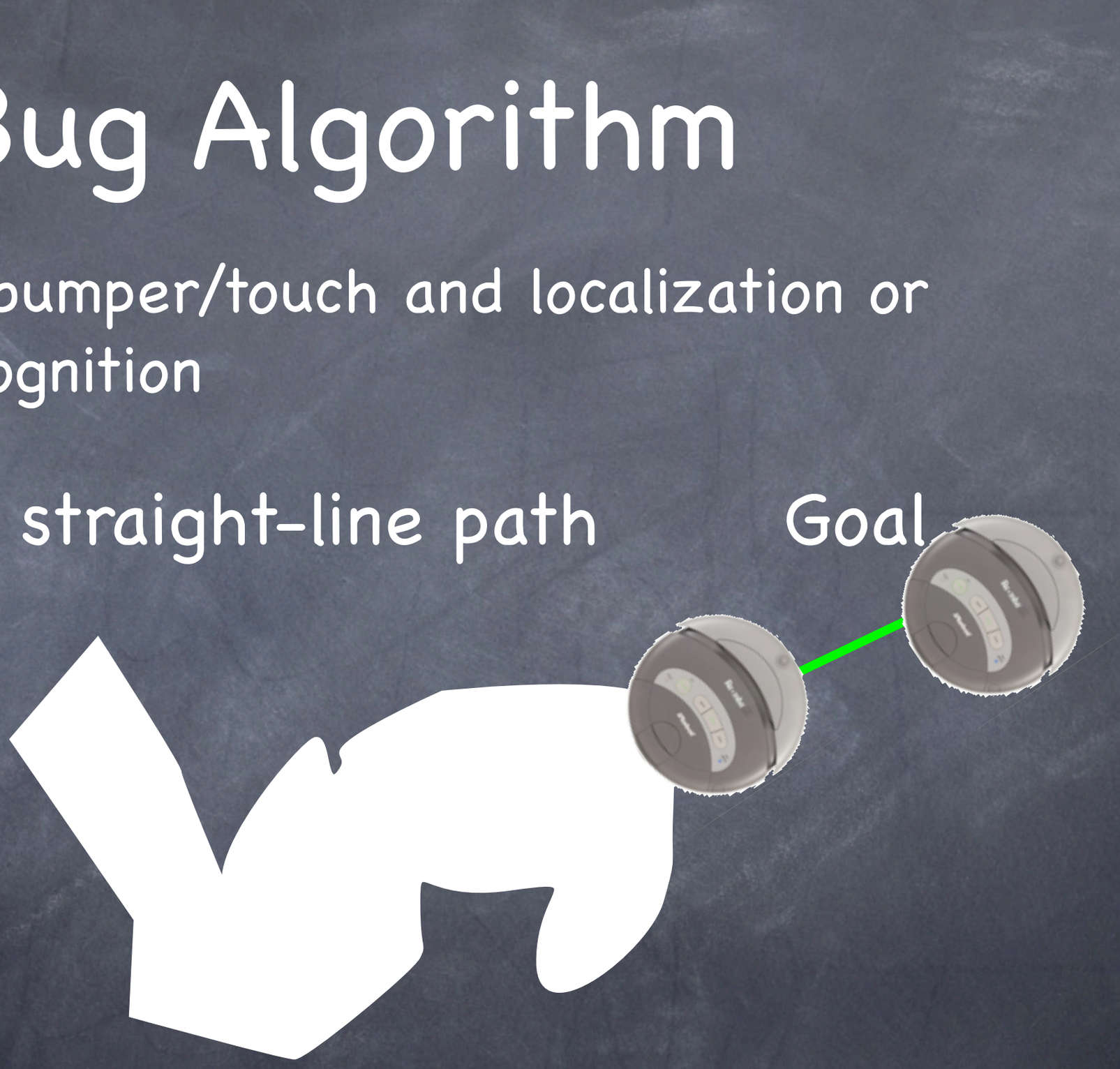
"exit point" along initial straight-line path

Bug Algorithm

- Assume bumper/touch and localization or goal recognition

Continue along straight-line path

Goal



Assumptions

- What is assumed for random walk controller?
- What is assumed by the bug algorithm?