CS4501
Robotics for Soft Eng

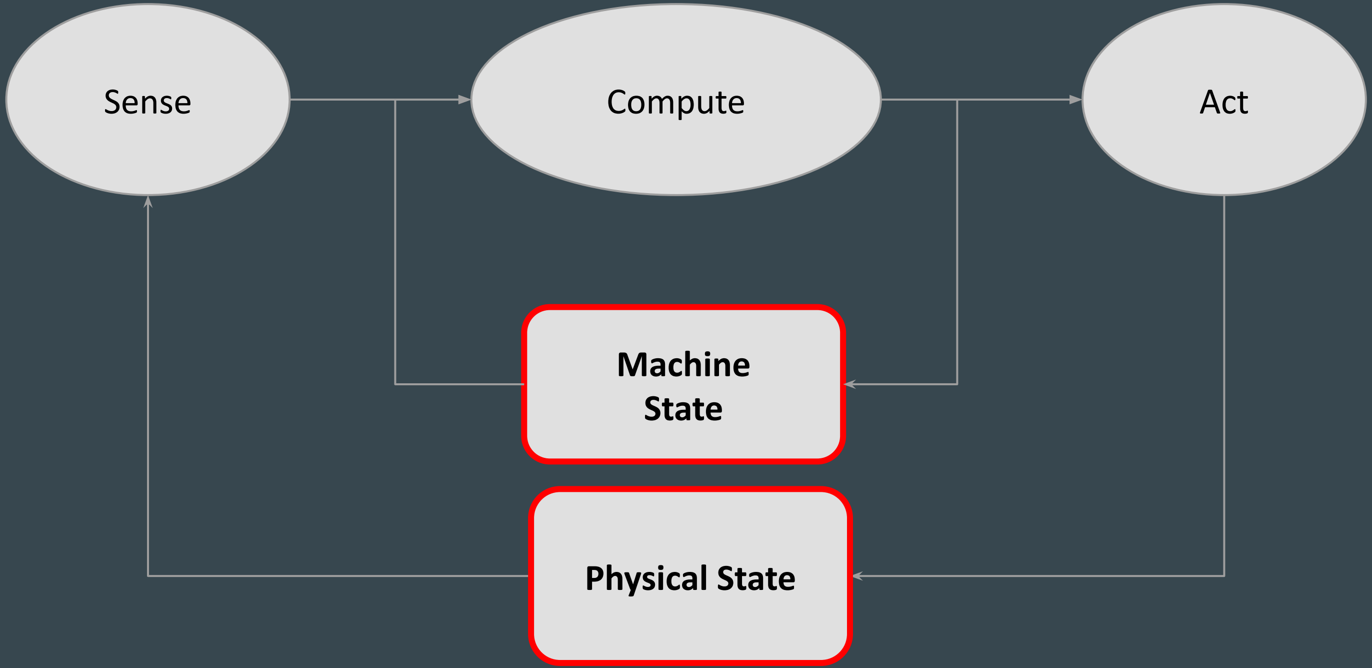
•••

Robot Architectures and Machinery

Robot Systems Architectural Attributes

- Asynchronous, event-driven -- world operates that way
- Decoupled -- parallelization, reuse
- Abstraction -- manage complexity
- Close loop -- need to assess/respond to changes

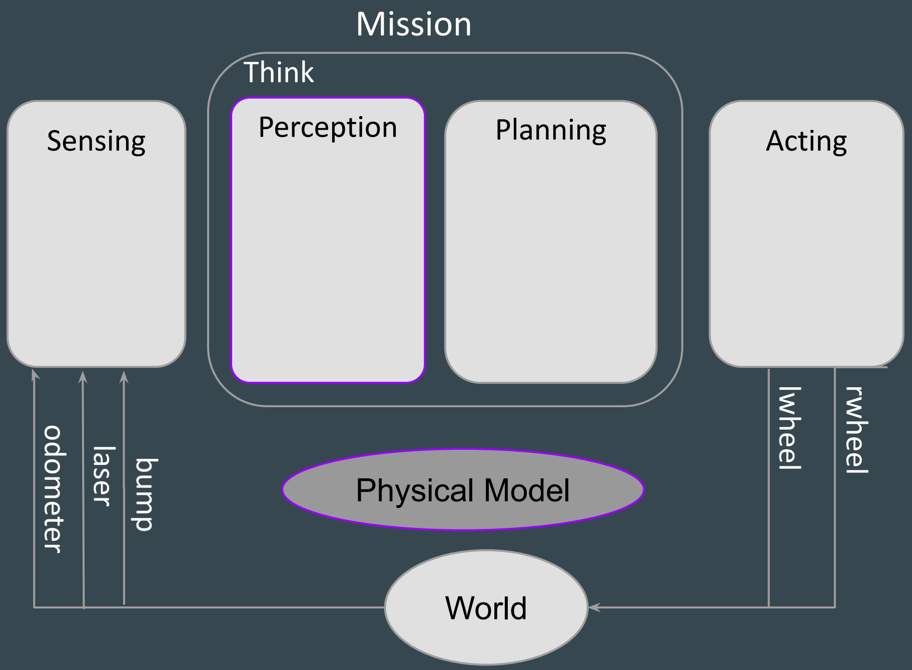
Conceptual Architecture - Structural Design



Physical State

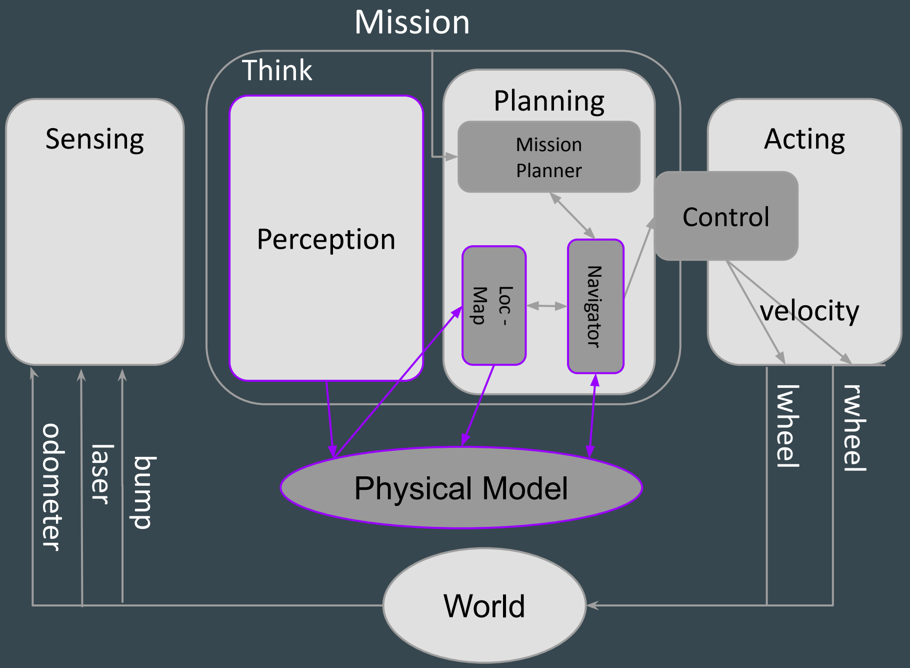
- Physical attributes that may change over time
- Some are sensed and some are estimated
- Robot State Examples
  - Roomba: senses odometry and velocity, estimates location
- World State Examples
  - Roomba: sense obstacles, estimates their location

Hierarchical/Deliberative my "Roomba"



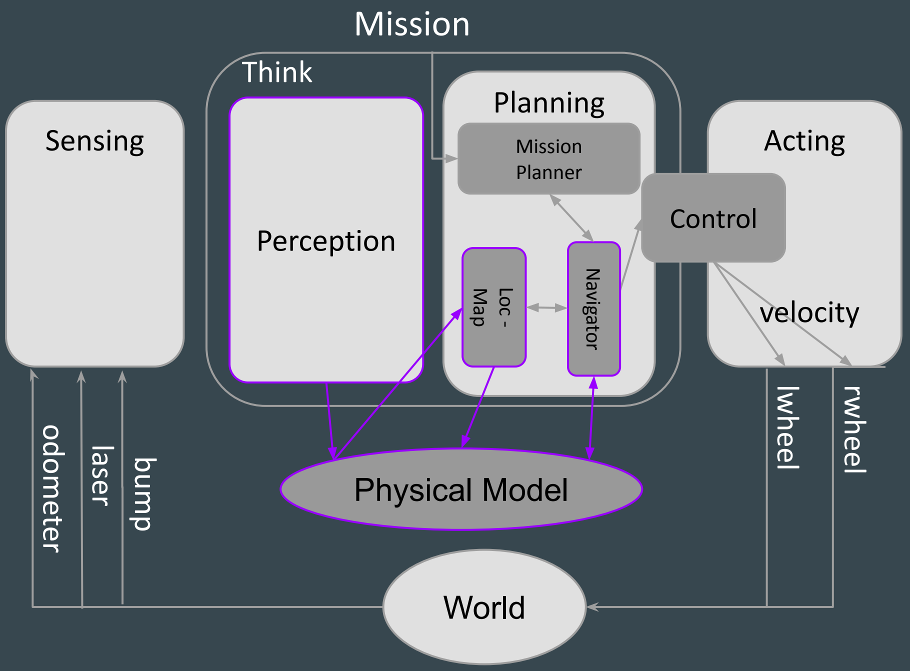
1. Senses world through multiple sensors
2. Perception updates interpretation of the world
3. Planning defines safe trajectory
4. Acting generates motor commands

Hierarchical/Deliberative my "Roomba"



1. Senses world through multiple sensors
2. Perception updates interpretation of the world
3. Mission planner sets high-level objectives based on mission
4. Loc/Map reads model to infer where we are and builds/refines map
5. Navigator
   - Reads world to get map
   - Compute paths to meet objective
   - Tells planner when mission is complete or if objectives need revision
6. Controller transforms waypoint in path into motor commands

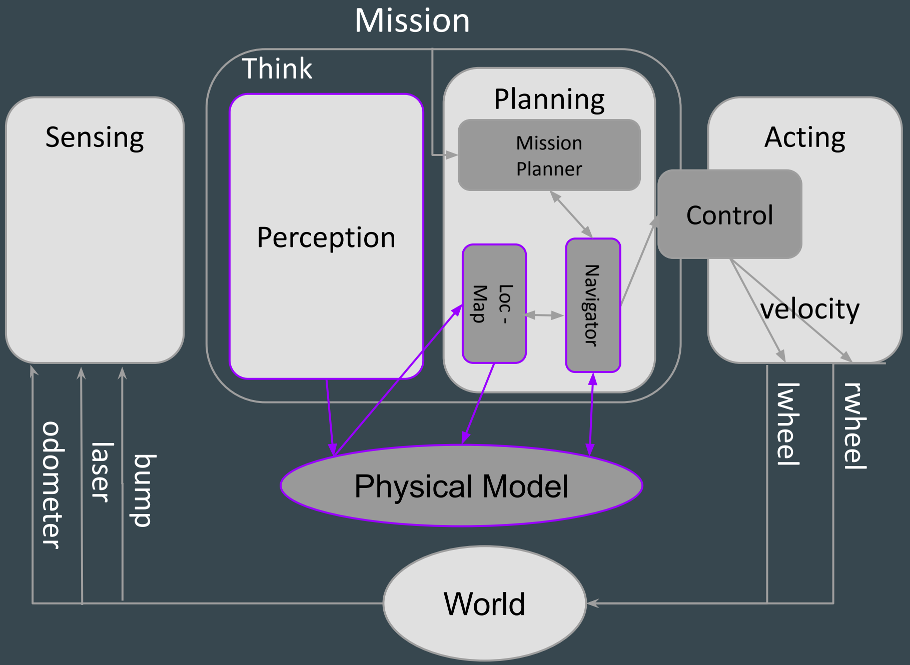
Hierarchical/Deliberative my "Roomba"



1. Senses world through multiple sensors
2. Perception updates interpretation of the world
3. Mission planner sets high-level objectives based on mission
4. Loc/Map reads model to infer where we are and builds/refines map
5. Navigator
    - Reads world to get map
    - Compute paths to meet objective
    - Tells planner when mission is complete or if objectives need revision
6. Controller transforms waypoint in path into motor commands

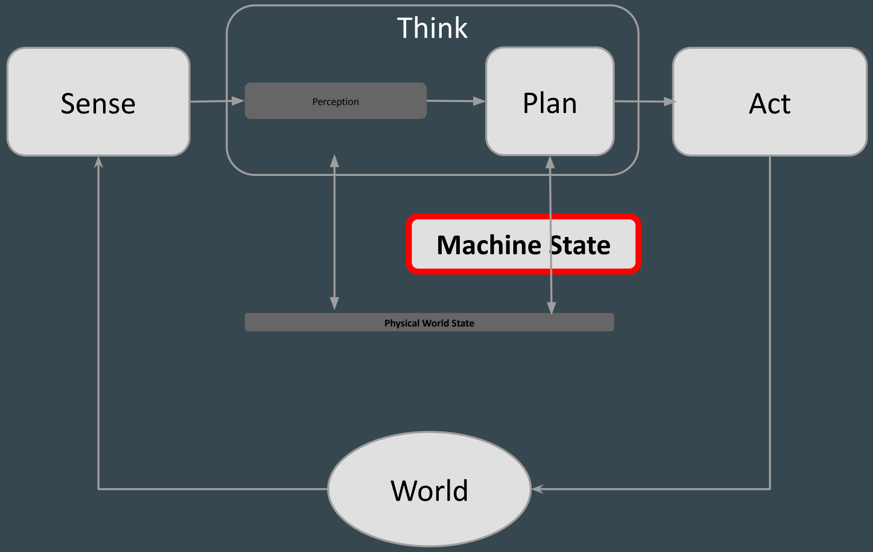
What can go wrong? - 2 min

Hierarchical/Deliberative my "Roomba"



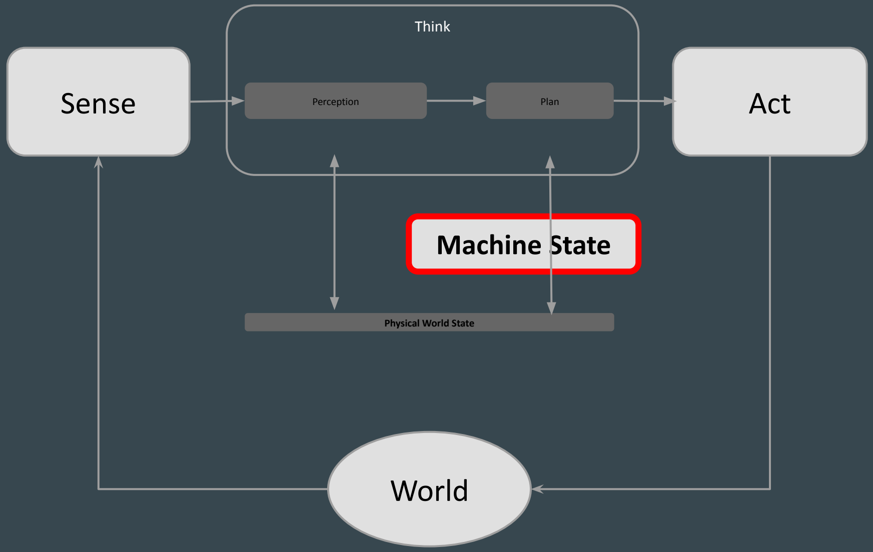
- World is too complex to model accurately / completely
- World changes faster than we can plan
- Difficult to extend functionality due to layers dependencies

Dominant Architectural Types: Reactive



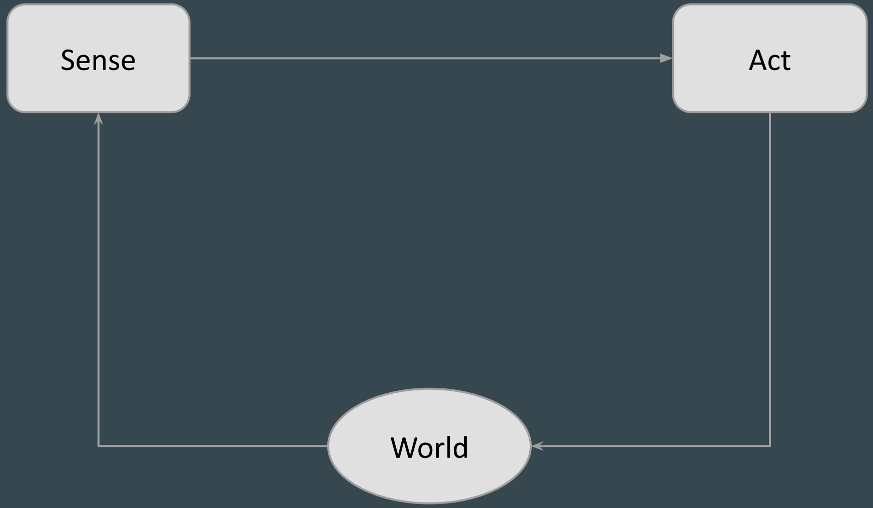
- Bio-inspired -- think insects
- No/Less reliance on model

Dominant Architectural Types: Reactive



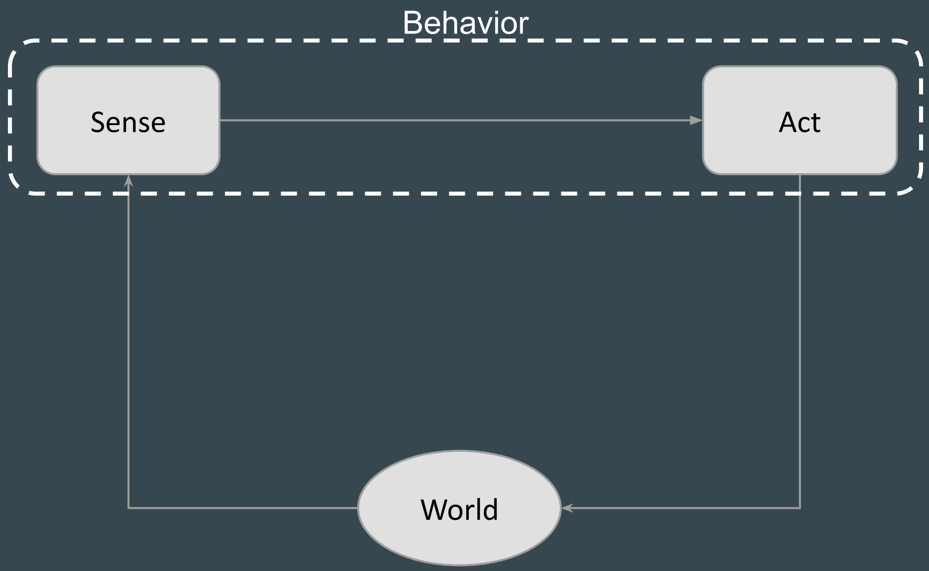
- Bio-inspired -- think insects
- No/Less reliance on model
- No thinking, more like intuitive reactions

Dominant Architectural Types: Reactive



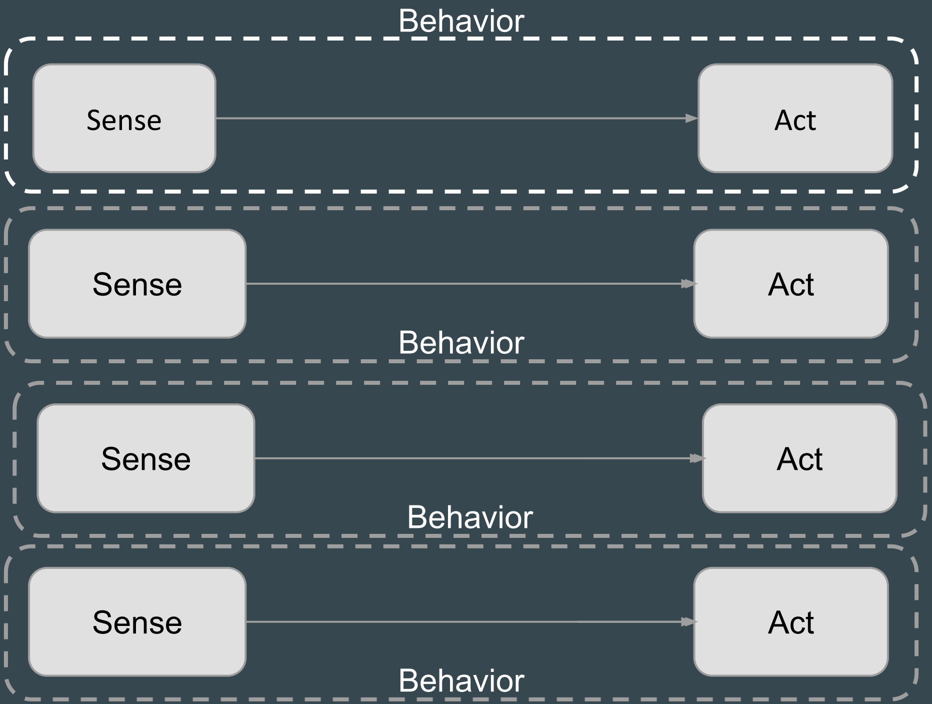
- Bio-inspired -- think insects
- No/Less reliance on model
- No thinking, more like intuitive reactions
- Fast acting

Dominant Architectural Types: Reactive



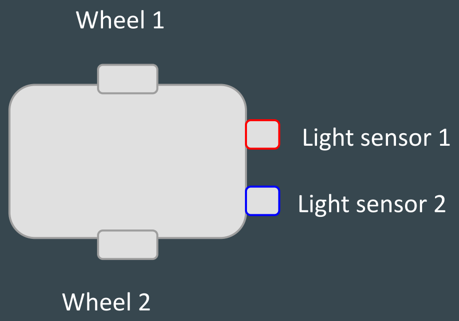
- Bio-inspired -- think insects
- No/Less reliance on model
- No thinking, more like intuitive reactions
- Fast acting
- Decomposition of behaviors

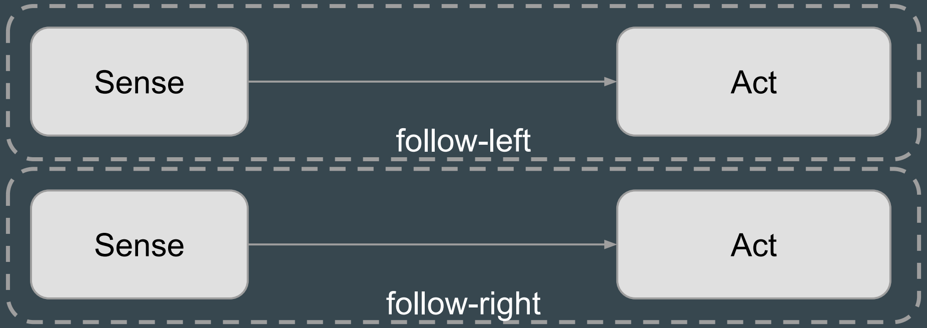
Dominant Architectural Types: Reactive



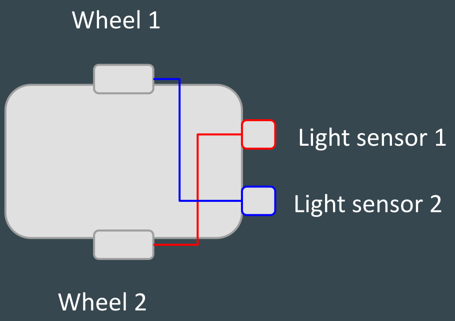
- Bio-inspired -- think insects
- No/Less reliance on model
- No thinking, more like intuitive reactions
- Fast acting
- Decomposition of behaviors

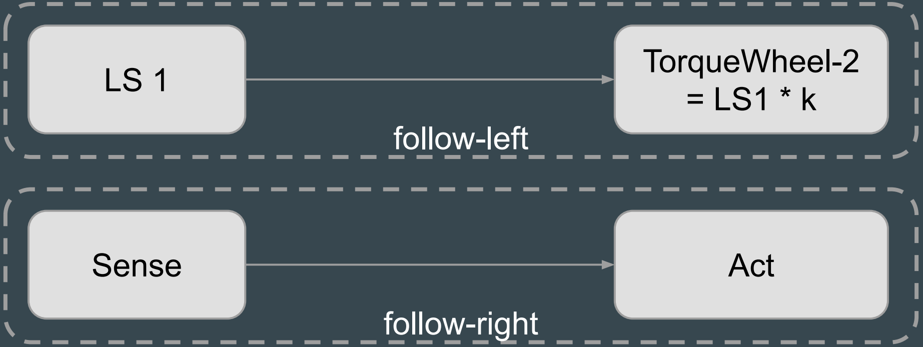
Dominant Architectural Types: Reactive “Moth”



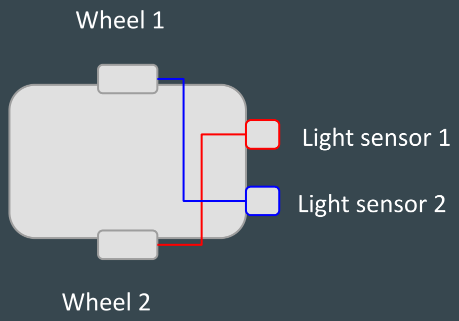


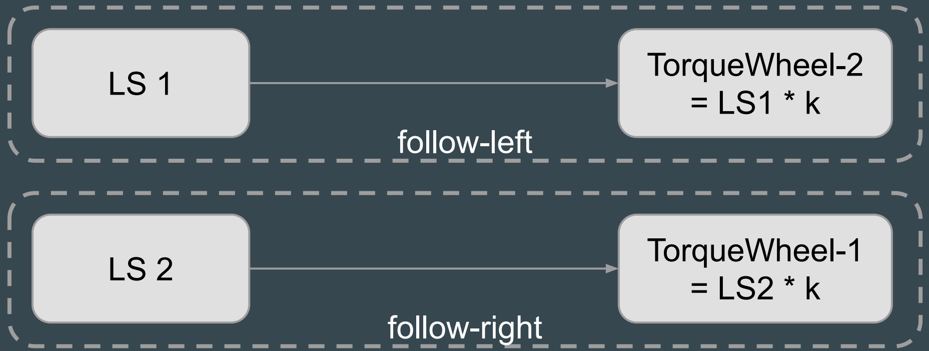
Dominant Architectural Types: Reactive “Moth”



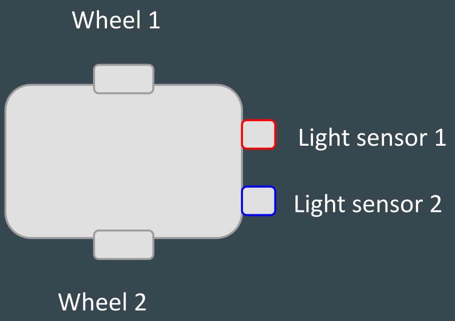


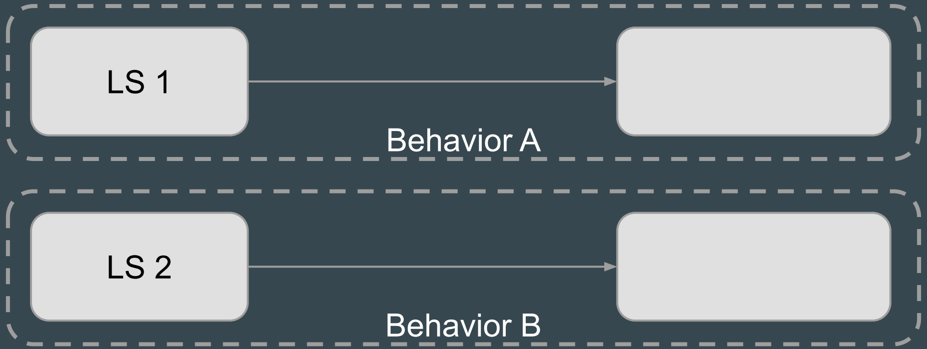
Dominant Architectural Types: Reactive “Moth”





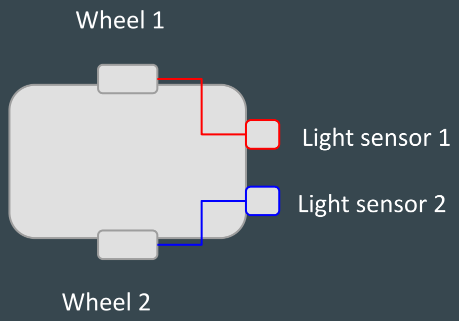
Dominant Architectural Types: Reactive

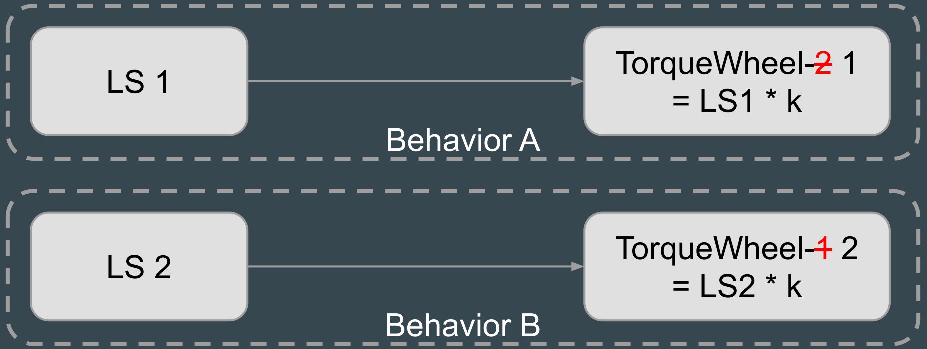




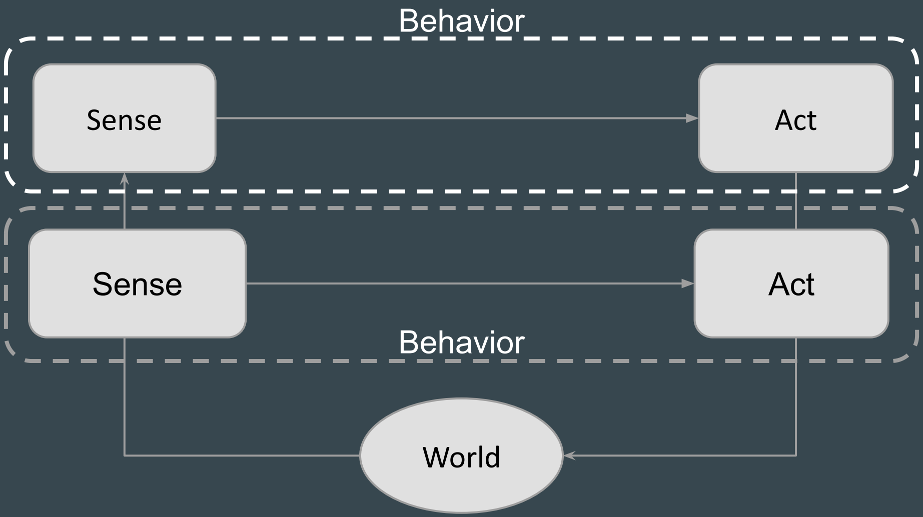
Change to “Cockroach” - 1 min

Dominant Architectural Types: Reactive "Cockroach"





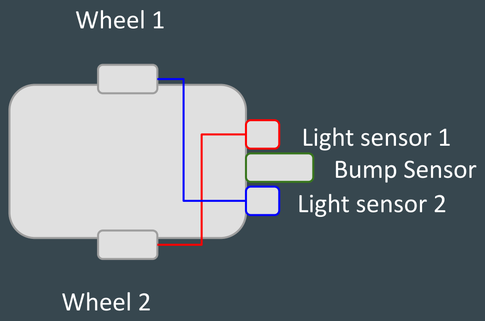
Dominant Architectural Types: Reactive

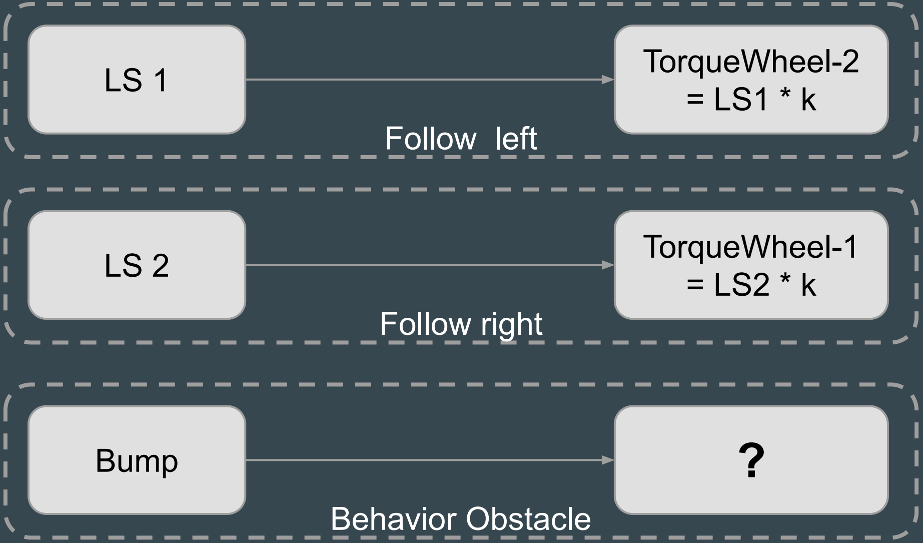


- Bio-inspired -- think insects
- No/Less reliance on model
- No thinking, more like intuitive reactions
- Fast acting
- Decomposition of behaviors

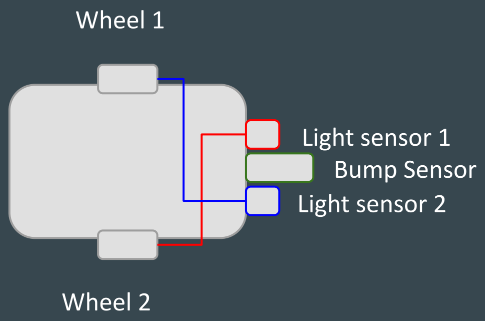
What can go wrong? - 2 min

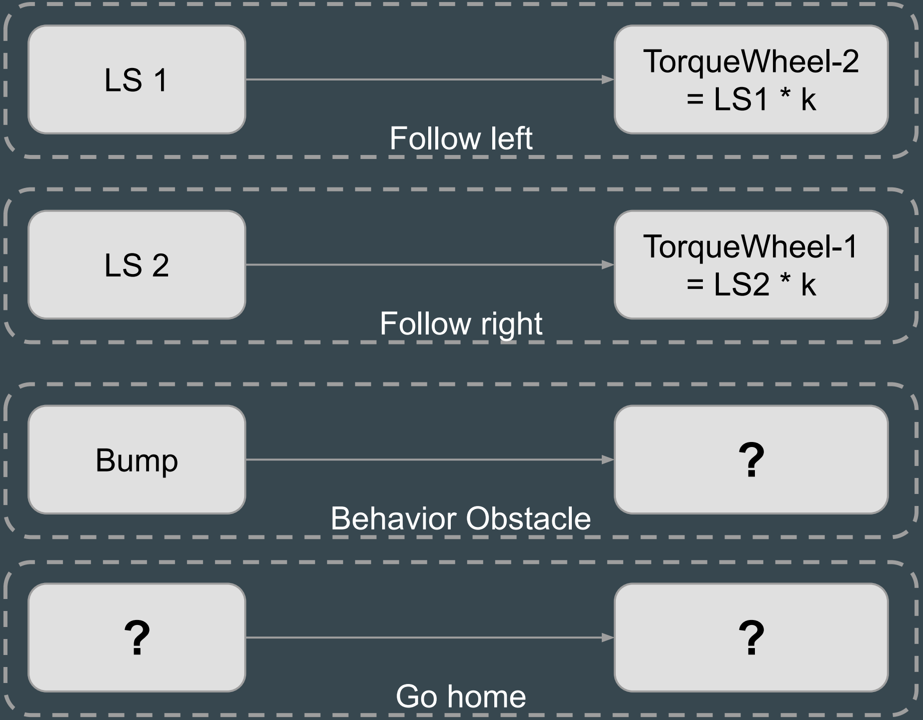
Dominant Architectural Types: Reactive "Light Follower"



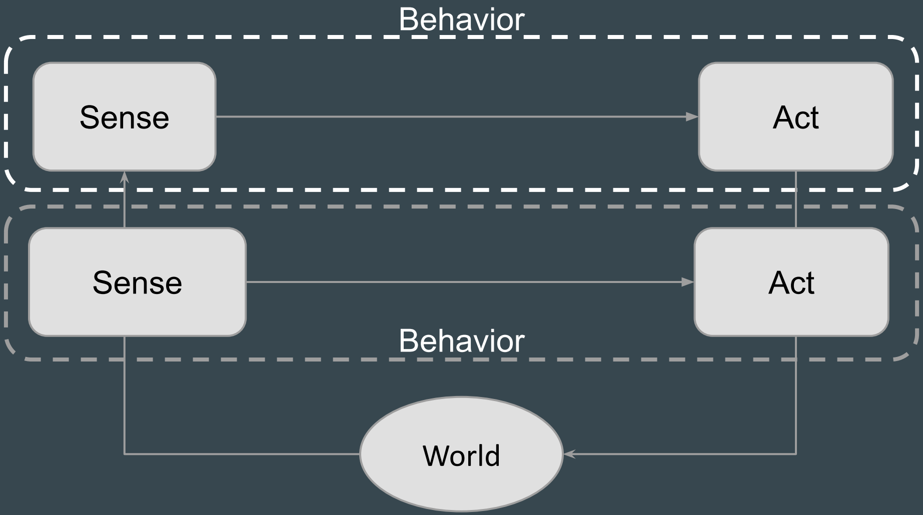


Dominant Architectural Types: Reactive “Light Follower”





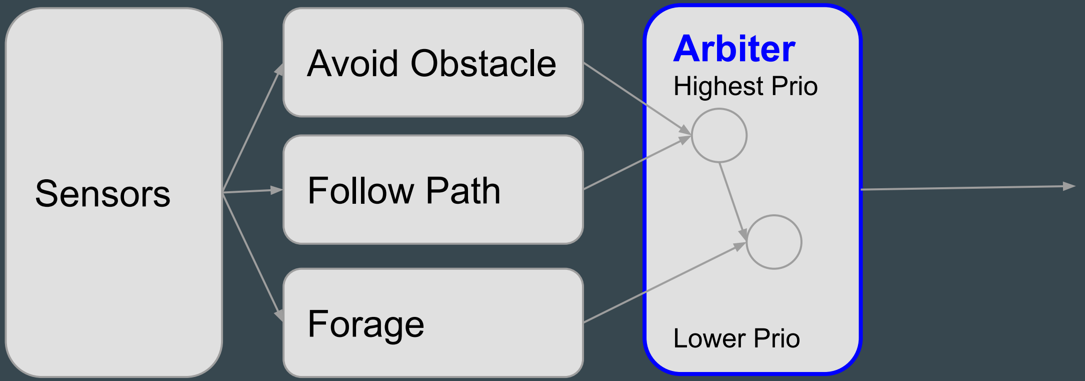
Dominant Architectural Types: Reactive



- Bio-inspired -- think insects
- No/Less reliance on model
- No thinking, more like intuitive reactions
- Fast acting
- Decomposition of behaviors

- Prioritizing behaviors and handling dependencies
- Achieving high level goals or complex behaviors

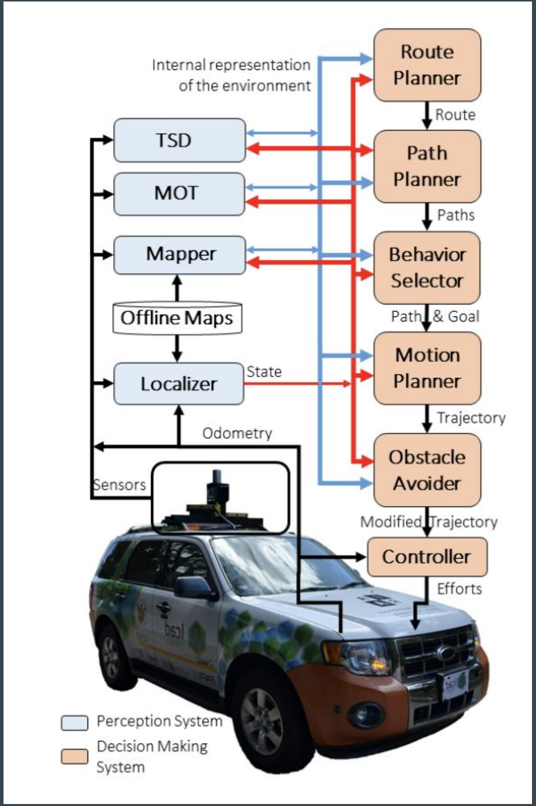
Dominant Architectural Types: Reactive



Handling dependencies with arbiters or additional logic

Reconsidering Architectures

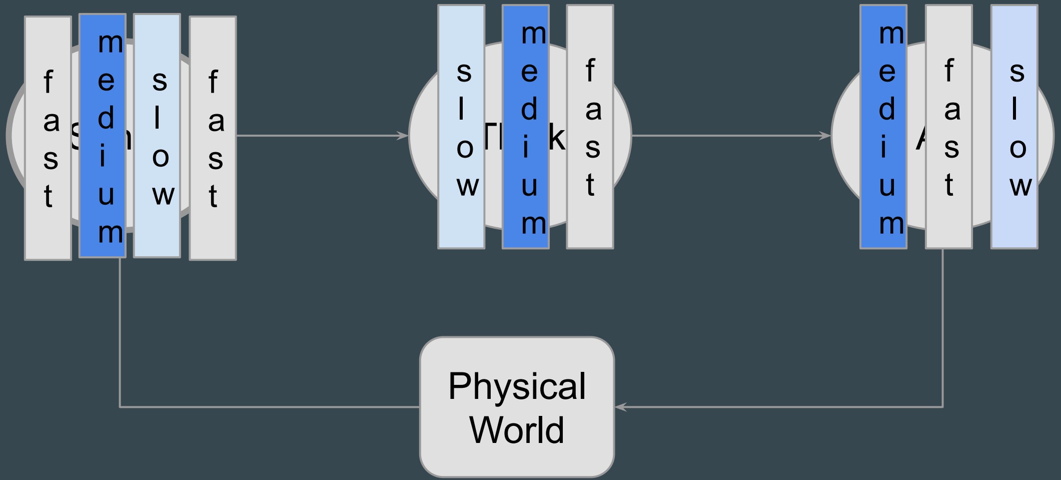
- Modular Decomposition is key
  - To develop and reuse
  - To test
  - To isolate failures
- Criteria
  - By features
  - Temporal



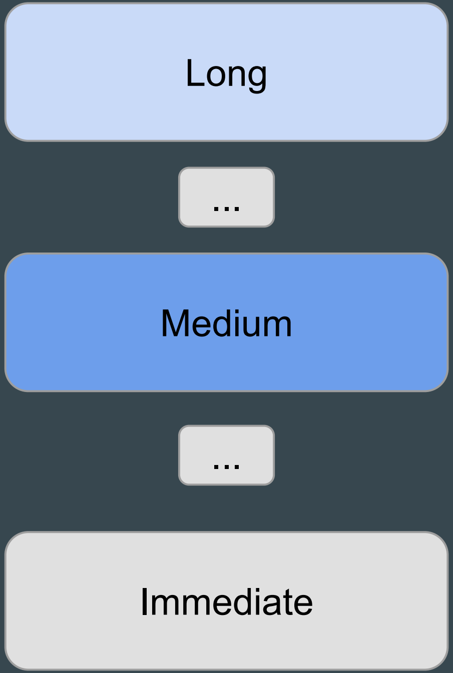
https://arxiv.org/pdf/1901.04407.pdf

Architectures: Temporal decomposition

- Time to sense
- Time to think
- Time to act



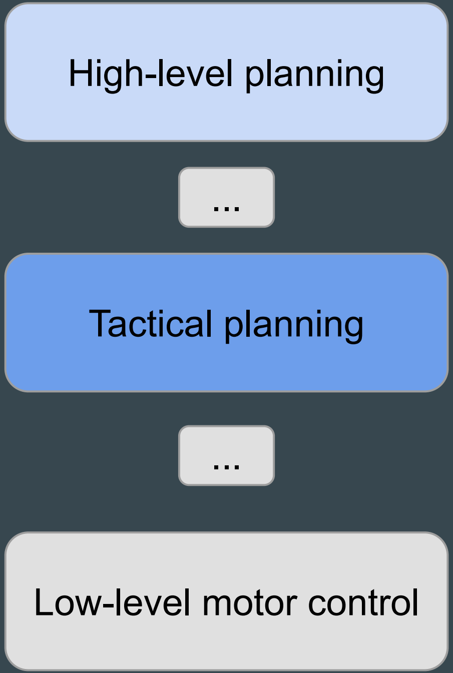
Architecture: Temporal decomposition



As slow as deliberative speeds

As fast as sensing speeds

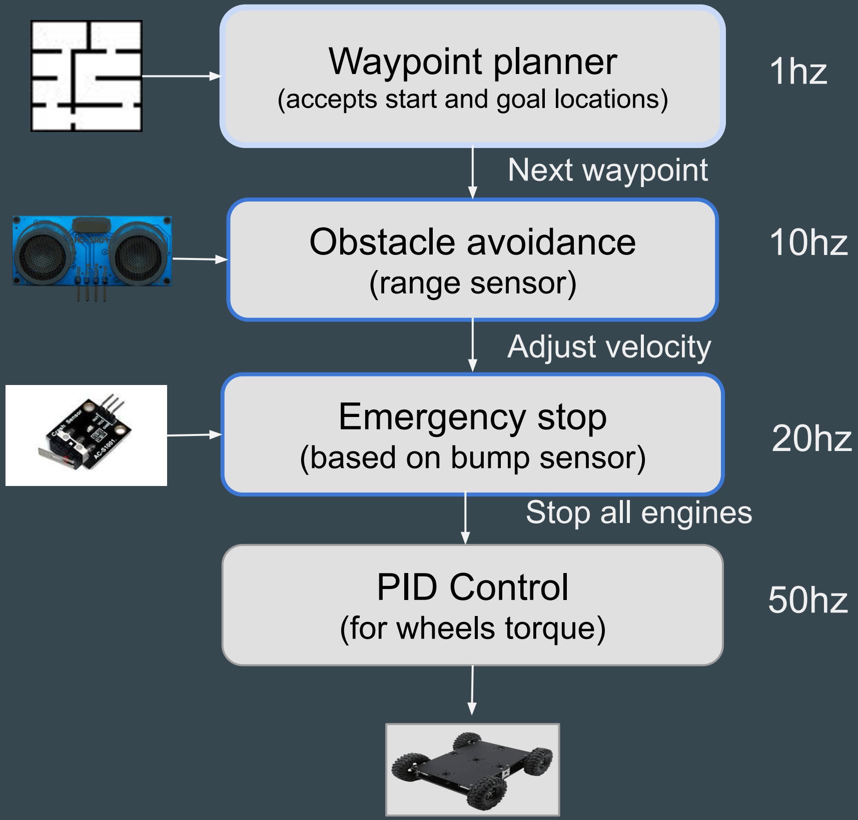
Architecture: Temporal decomposition



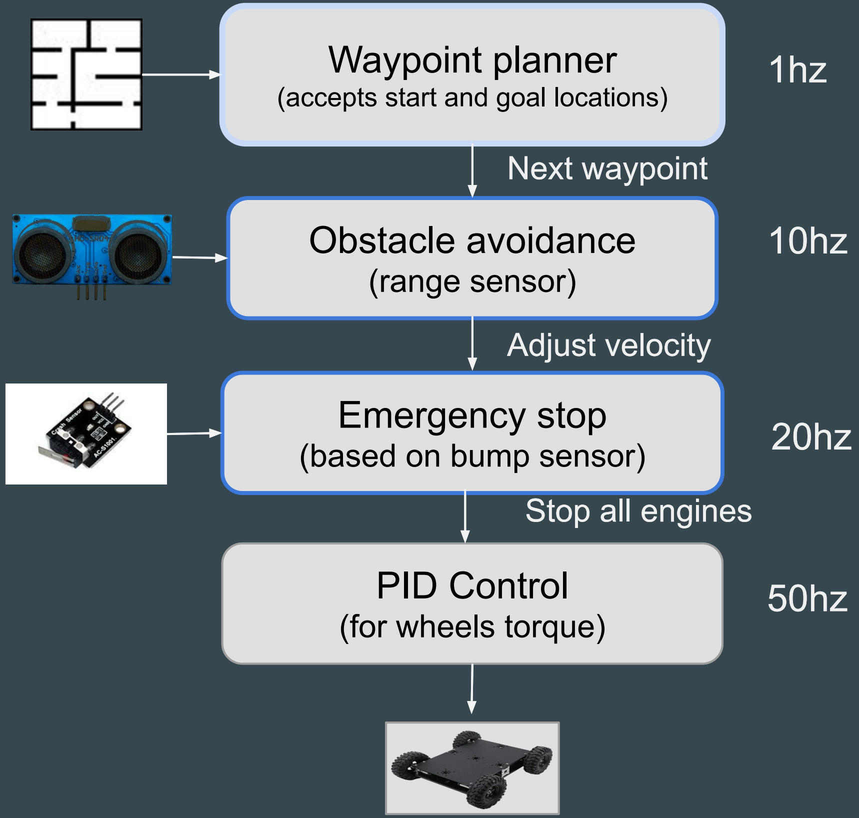
As slow as deliberative speeds

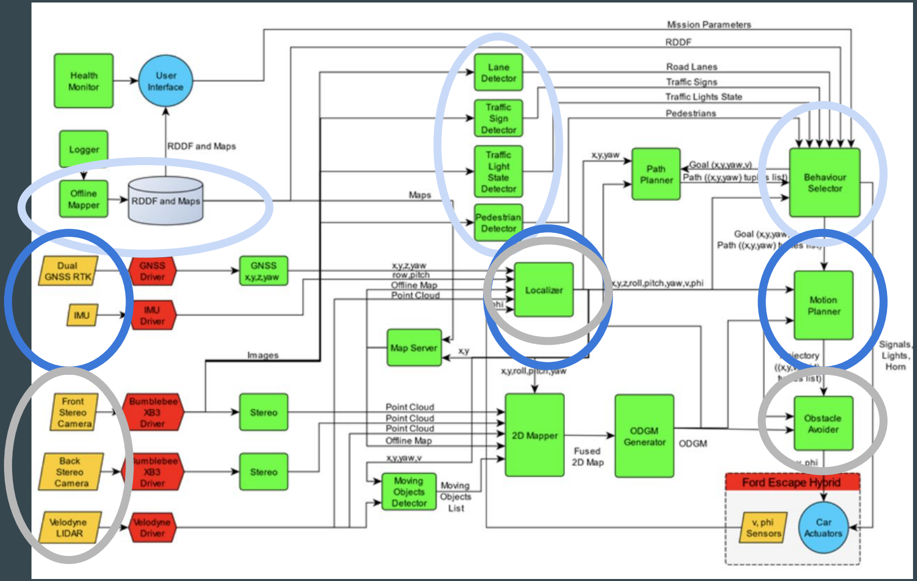
As fast as sensing speeds

Architecture: Temporal decomposition



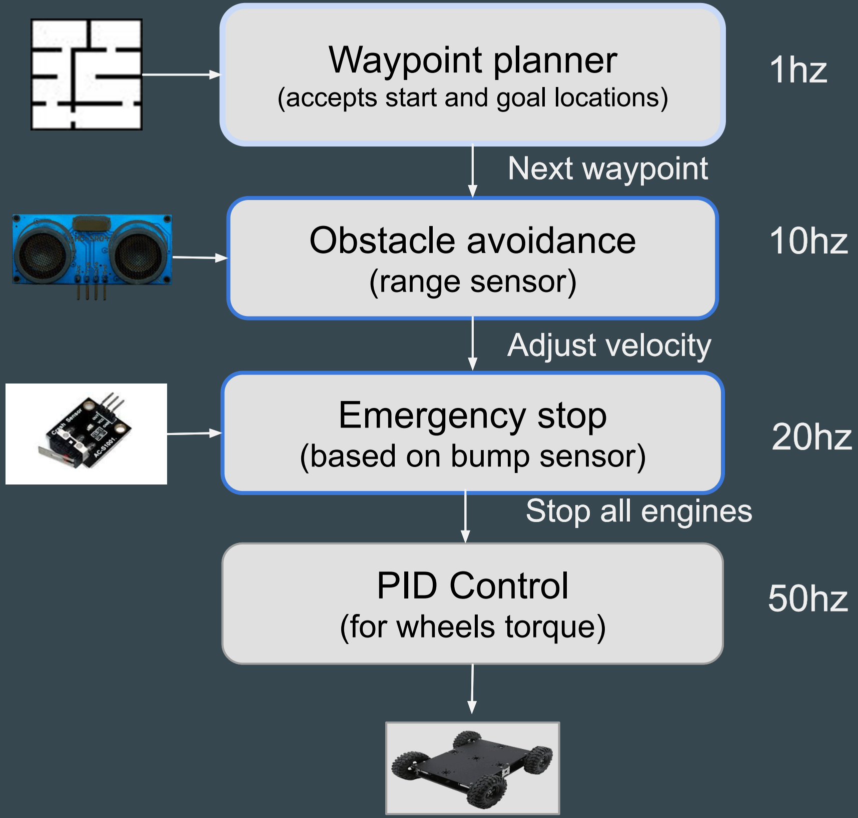
Architectures: Temporal decomposition





IARA Software Architecture on Ford Escape

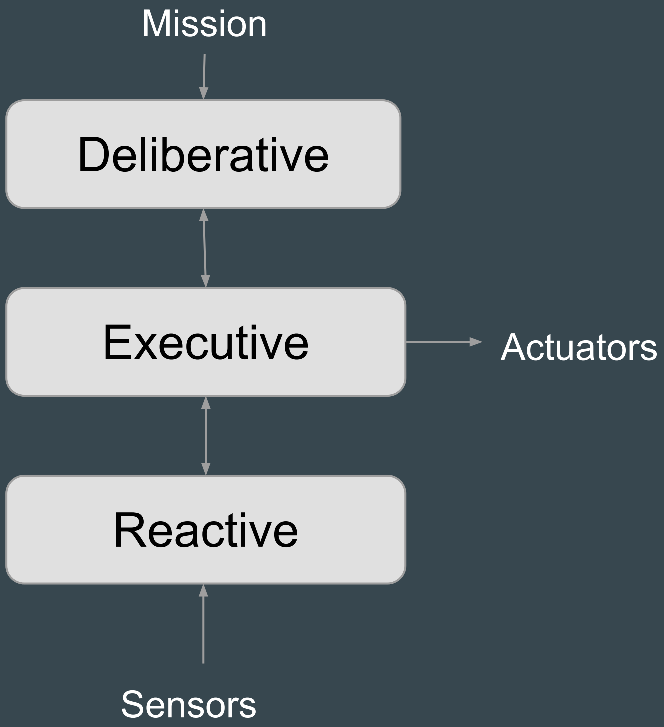
Architectures: Temporal decomposition



- Model for context
- Spatial locality
- Memory & lookahead
- Sensor speed

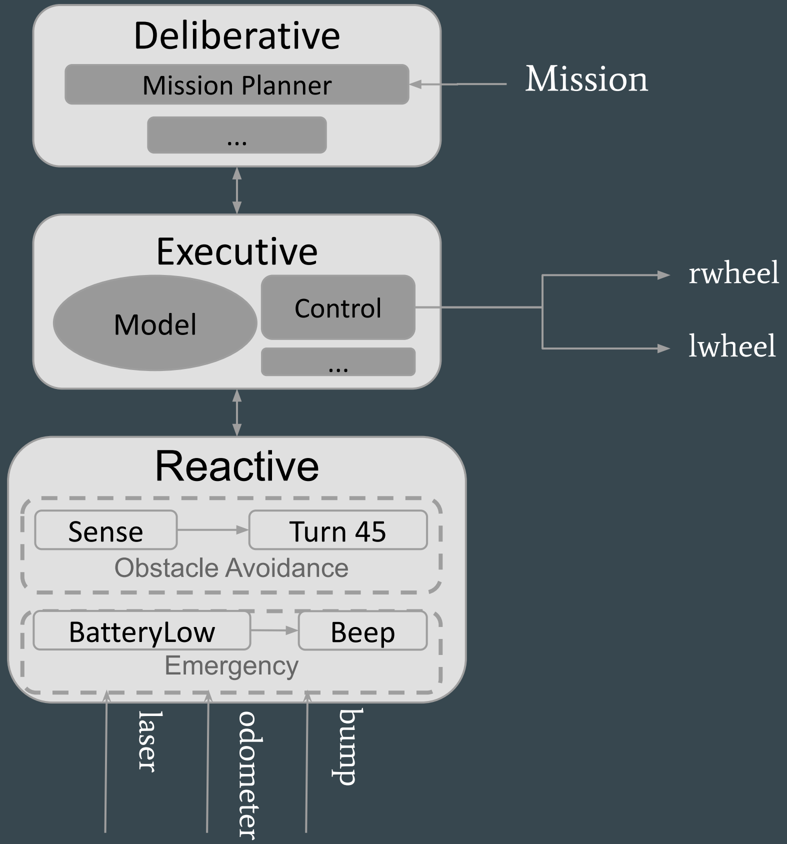
Decomposition Criteria

Dominant Architectural Types: Hybrid - 3 Tier

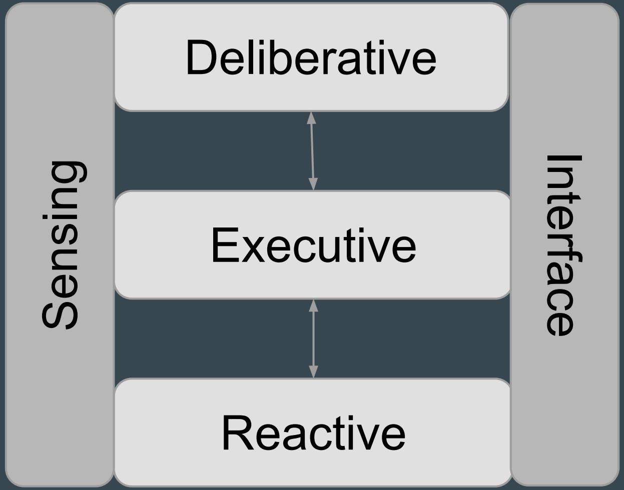


- Deliberative
  - Long term planning
  - Uses world representation
- Executive
  - Glue
  - Maintains world representation
  - Translates directives into lower level commands
- Reactive
  - Low level behaviors
  - Connects sensors-actors

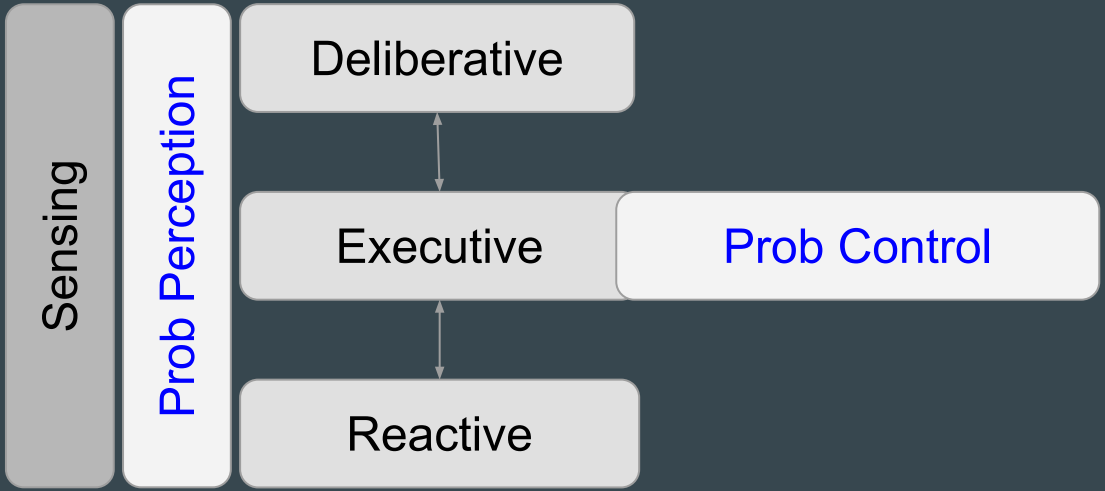
Dominant Architectural Types: Hybrid - 3 Tier Our Bot



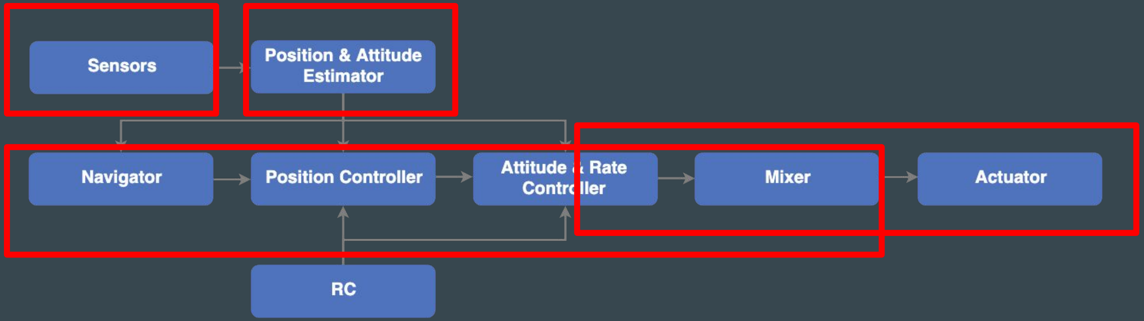
Dominant Architectural Types: Hybrid - Variations



Dominant Architectural Types: Probabilistic



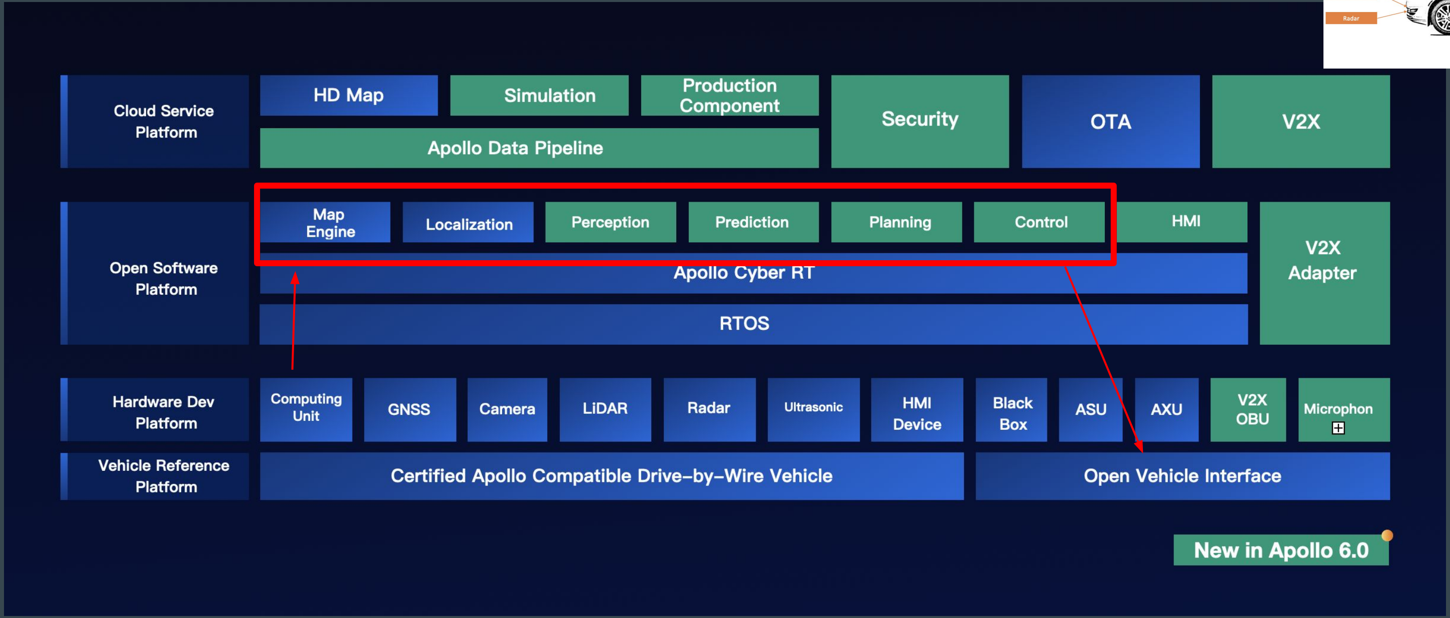
Reality is a bit messier

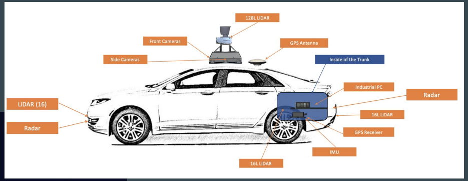


PX4 - Autopilot

https://docs.px4.io/master/en/concept/architecture.html

Reality is a bit messier

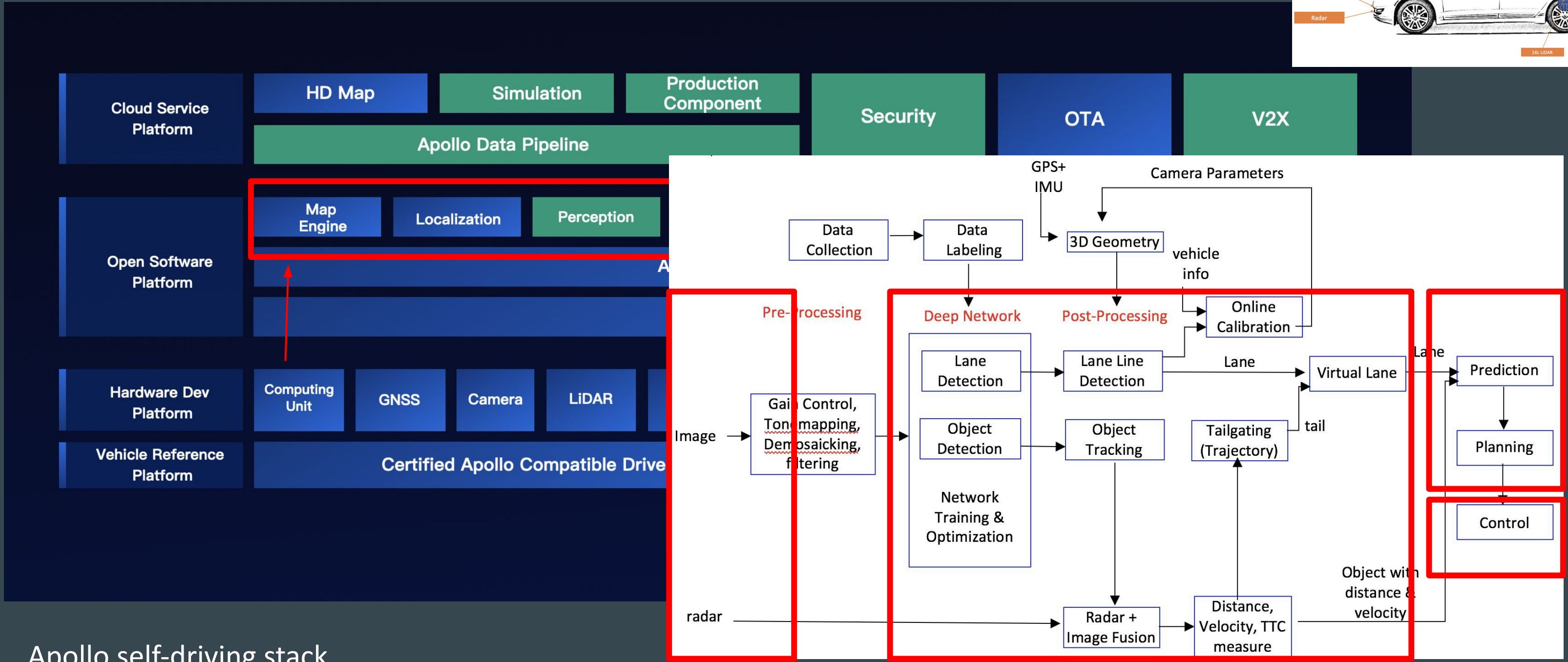


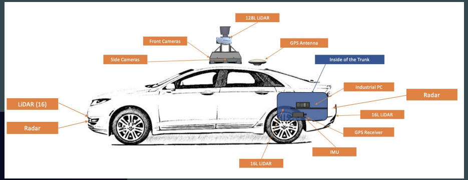


Apollo self-driving stack

https://github.com/ApolloAuto/apollo

Reality is a bit messier





Apollo self-driving stack

https://github.com/ApolloAuto/apollo

Taking stock

- **Deliberative**
  - Think hard, act later
  - Lots of states
  - Maps of the robot environment
  - Look ahead
- **Reactive**
  - Do not think, react
  - Less/No world states. Less/No maps. No look ahead
  - Reactive + state: Behavior, look ahead only while acting
- **Hybrid**
  - Think and act independently.
  - States. Look ahead in parallel to acting.
  - Combines long and short time scales