LECTURE 5-2

The Hybrid Deliberative/Reactive Paradigm

Introduction to AI Robotics (Ch. 7)

Quote of the Week

Asimov's Robot Laws

A robot may not injure a human being, or, through inaction, allow a human being to come to harm

A robot should obey a human being, unless this contradicts the first law

A robot should not harm another robot and protect its own existence unless the contradicts the first or second law.

From Handbook of Robotics, 56th Edition, 2058 A.D., as quoted in I, Robot.

ANNOUNCEMENTS

- Quiz 10 on Ch. 7, Lec. 5-2 on *Monday, 4/19/10*
- Lab 6 due on *Tuesday, 4/20/10*
- Lab 6 memo and code is due on Angel by midnight on *Thursday, 4/22/10*

OBJECTIVES

Upon completion of this lecture the student should be able to:

- Describe the hybrid deliberative/reactive paradigm in terms of the three robot primitives and sensing organization
- Name and evaluate on representative hybrid architecture
- Give a list of responsibilities and where they belong in the reactive or deliberative layer
- List the 5 basic components of a hybrid architecture

HYBRID CONTROL

- Recall that *reactive control* is fast but inflexible, there was no memory or planning
- *Deliberative control* is smart but slow
- *Hybrid control* exploits the best features of reactive and deliberative control
- *Hybrid control* involves the combination of reactive and deliberative control within a single robot control system
- Therefore, short and long time scales must work together
- No representatives and explicit and elaborate world models must be made to work together effectively

HYBRID CONTROL

- Hybrid control is the best general architectural solution for robotics because
  - Asynchronous processing techniques allow deliberative functions to operate independent of reactive behaviors
  - A planner can slowly compute the next goal for a robot while the robot reactively navigates to the current goal with fast update rates
  - Good software modularity allows subsystems or objects to be mixed and matched (purely reactive or entire architecture)

FUNCTION OF TIME

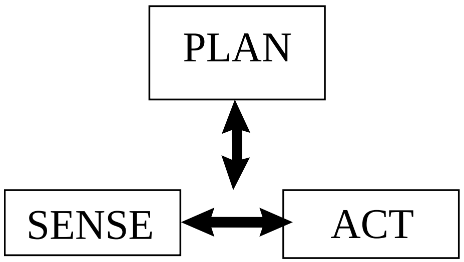
- Reactive Layer
  - Exists in the ***PRESENT*** or real time response to sensory input
- Deliberative Layer
  - Reasons about the ***PAST*** to create plans
  - Makes projections into the ***FUTURE*** for planning

REPRESENTATION

- Behaviors can interact with each other within the robot to store representation
- Behaviors can serve as a basis for learning and prediction which means they can achieve the same things as hybrid systems in a different way. The difference is how representation is used

THREE-LAYER ARCHITECTURE

- A hybrid system has three components:
  - A reactive layer
  - A planner
  - A layer that links the above two together

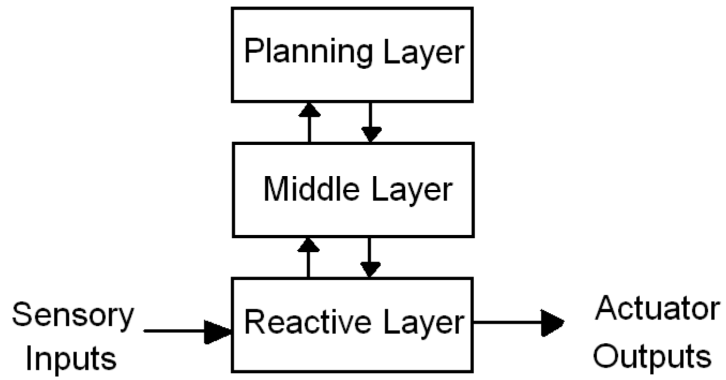


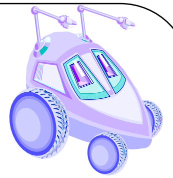
- The middle layer has to
  - Compensate for limitations of both the planner and the reactive system
  - Reconcile the two different time scales
  - Deal with their different representations
  - Reconcile any contradictory commands they may send to the robot

THREE-LAYER ARCHITECTURE

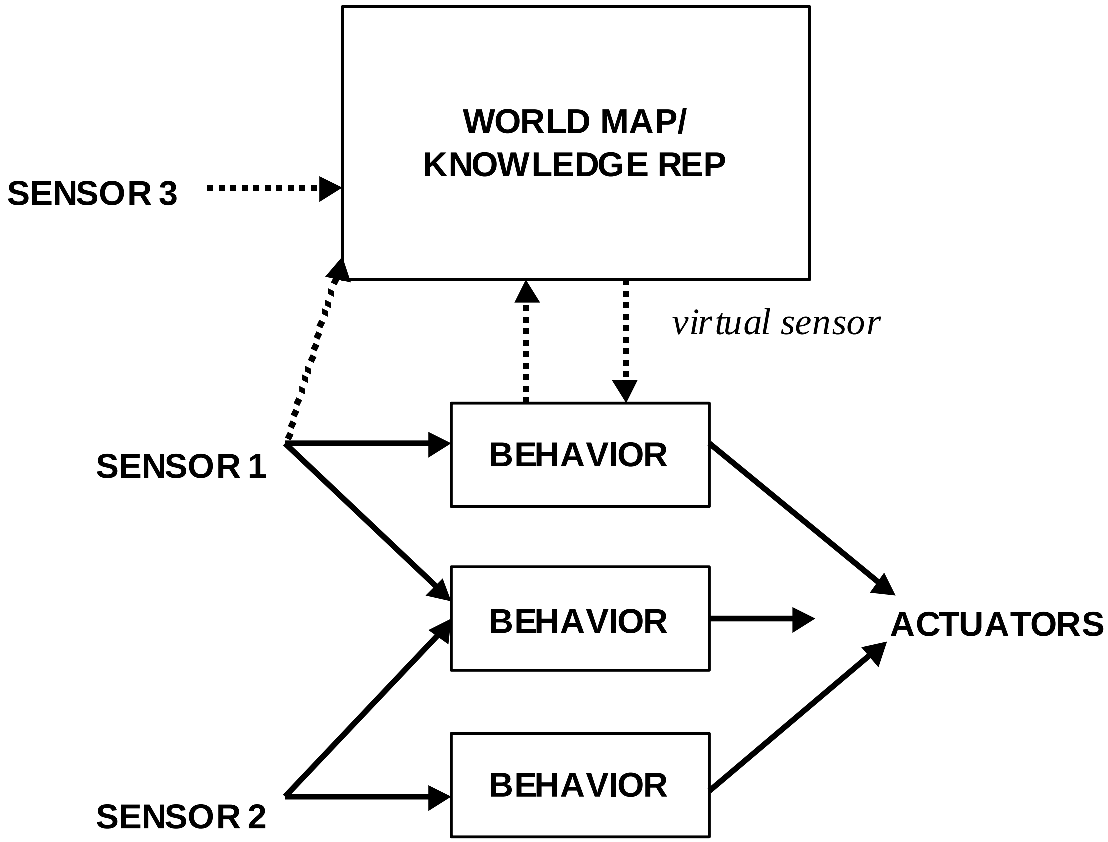
- When there is a change in the world, map or task the middle layer informs the deliberative layer to update its representation of the world to generate more accurate and useful plans
- *Dynamic replanning* occurs when the reactive layer discovers it cannot proceed. The planner can tell the robot to stop moving through the reactive layer

- One solution is to have some stored plans based upon common initial and goal states. They are not reactive because they have multiple steps and not plans so they reside in the middle layer





SENSING ORGANIZATION



Deliberative functions
**Can "eavesdrop"*
**Can have their own Sensors*
**Have output which Looks like a sensor Output to a behavior (virtual sensor)*

LOCAL VERSUS GLOBAL

- *Local* is typically associated with reactive control and *global* with hybrid control
- However, this may not always be true, i.e.
  - path planning and map making require a global world model
  - *Behavioral management* requires knowledge about the current mission and the current (projected) state of the environment (global and local)
  - *Performance monitoring* determines if a robot is actually making progress toward its goal (global)
- Deliberative may not need a global world model but know about the robot's internal state on a global scale

BEHAVIORS IN THE HYBRID ARCHITECTURE

- Behaviors are typically more complex and abstract than in the reactive paradigm
- The hybrid architecture uses assemblages of behaviors sequenced over time rather than primitive behaviors
- There is also more diversity in combining the output from concurrent behaviors in order to create complex emergent behaviors

ON-LINE AND OFF-LINE PLANNING

- *Off-line planning* is done to plan for all of the situations that might come up
- This takes place while the robot is being developed
- *On-line planning* takes place while the robot is busy trying to get a job done and achieve goals

- A *universal plan* is a set of all possible plans for all initial states and all goals within the state space. This is a *reactive robot* because the planning is done off-line not at run time

DOMAIN KNOWLEDGE

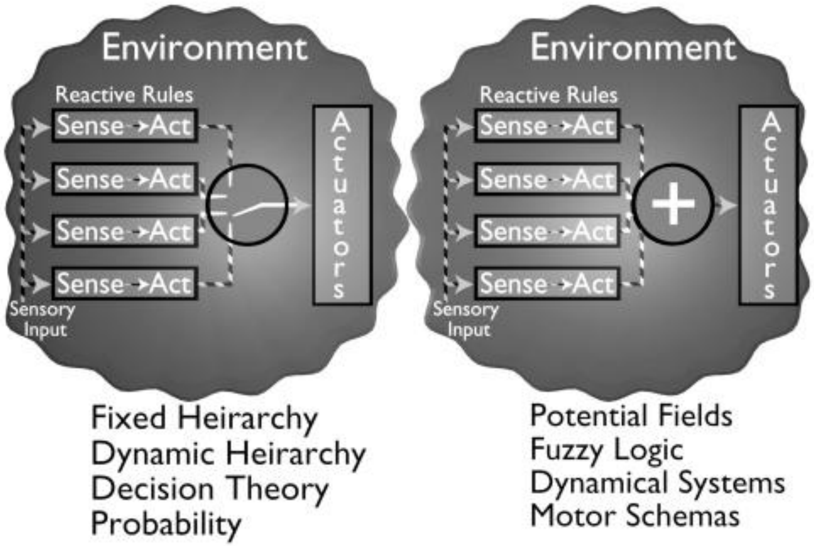
- Precompiled existing optimal plans can be put into the system in a clean, principled way.
- Information about the robot, the task and the environment is called *domain knowledge*
- It is compiled in a reactive controller so that it does not have to be planned on line in real time
- These plans become reactive rules to be looked up
- Drawback to situated automata
  - State space is too large for most realistic problems
  - The world must not change
  - The goals must not change

ACTION SELECTION OR BEHAVIOR COORDINATION

- The problem with *action selection* or *behavior coordination* is what action or behavior should be executed next.
- This problem must be resolved such that the robot always does the right thing over time in order to achieve its goals
- *Hybrid* and *Behavior-Based* systems must resolve the coordination problem
- In *Hybrid* systems, the middle layer handles coordination
- In *Behavior-Based* systems, the coordination may be distributed throughout the control system

BEHAVIOR ARBITRATION VERSUS FUSION

- *Behavior Arbitration* is picking one behavior /action
- *Behavior Fusion* is combining multiple behaviors /actions



ARBITRATION SELECTION

- *Fusion methods* are applied to lower-level representations of actions or behaviors (direction, velocity), i.e. potential fields
- *Arbitration methods* are used for higher-level ones such as (turn, stop, grasp), i.e. subsumption
- Many control systems are a combination of methods with fusion at one level and arbitration at another

BEHAVIOR ARBITRATION

- Arbitration-based behavior coordination is also called *competitive behavior* because candidate behaviors compete but only one can win
- In a *fixed priority hierarchy*, the behaviors have pre-assigned priorities
- In a *dynamic hierarchy*, the behavior priorities change at run-time
- *Subsumption architecture* uses a fixed priority hierarchy of behaviors. Some *hybrid systems* also employ fixed priority hierarchy

BEHAVIOR FUSION

- *Behavior fusion* is the process of combining multiple possible candidates into a single output action/behavior
- *Behavior fusion* is also called a cooperative method because it combines outputs of multiple behaviors to produce a final result
- This result may be an existing behavior or a new one (*emergent behavior*)
- There could be weighting or have some logic in the system to prevent certain combinations and outcomes

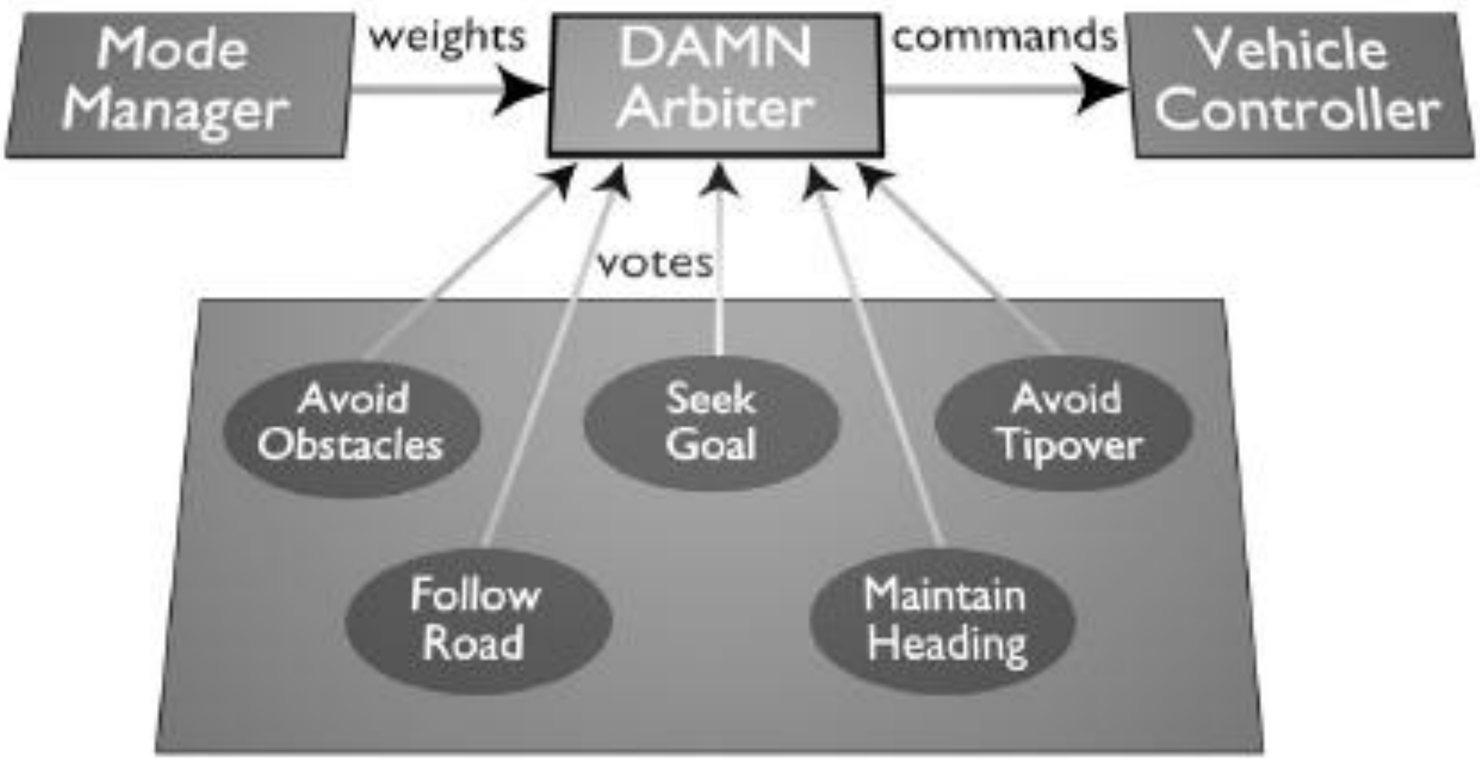
TYPES OF HYBRID ARCHITECTURES

- Types are distinguished by
  - How does the architecture distinguish between reaction and deliberation?
  - How does it organize responsibilities in the deliberative portion?
  - How does the overall behavior emerge?
- The two primary means for combining reactive behaviors are subsumption and potential fields
- These are limited so there are now 3 other mechanisms including
  - Voting – DAMN
  - Fuzzy logic – Saphira
  - Filtering - SFX

DYNAMIC AUTONOMOUS MOBILE NAVIGATION (DAMN)

- *DAMN* was used to control a van that autonomously drove on roads over vary large distances
- In *DAMN*, several low level actions are voting members and the result is a weighted sum of actions.
- This was an example of command fusion and was later improved with fuzzy logic
- Formal methods for command fusion include *potential fields*, *motor schema* and *dynamical systems*

DAMN ARCHITECTURE



HYBRID ARCHITECTURE COMPONENTS

- Generally, all hybrid architectures have:
  - *Sequencer* – generates a set of behaviors to use in order to accomplish a subtask
  - *Resource manager* – allocates resources to behaviors, i.e. sensors for ranging
  - *Cartographer* – creates, stores and maintains maps or spatial information
  - *Mission planner* – interacts with the human to create plan for the robot
  - *Performance monitoring and problem solving* – determines if the robot is making progress and requires robot self-awareness

THREE CATEGORIES OF HYBRID ARCHITECTURE

- Managerial
  - Divide the layers into scope of control for example the mission planner directs the path planner
- State Hierarchies
  - Use the robot’s state to distinguish between reactive and deliberative activities
- Model-oriented
  - Not clear delineation of roles, behaviors have access to the world model

DRAWBACKS TO HYBRID CONTROL

- The middle layer is hard to design and implement, very task and robot specific
- A hybrid system can degenerate into having the planner slow down the reactive system so reactive system may ignore the planner
- An effective hybrid system is difficult to design and/or debug