Perception and Planning

Week 7

Behavioural

Architectures

Autonomous Control Systems

- Nature is filled with examples of autonomous creatures capable of dealing with the diversity, unpredictability, and rapidly changing conditions of the real world.

Autonomous Control Systems

- Such creatures must make decisions and take actions based on incomplete perception, time constraints, limited knowledge about the world, cognitive abilities, reasoning and physical capabilities, in uncontrolled conditions and with very limited cues about the intent of others.
- Consequently, one way of evaluating intelligence is based on the creature’s ability to make the most of what it has available to handle the complexities of the real world.

Uncertainty in Robot Systems

- Robot systems in unstructured environments have to deal with sensor noise and uncertainty
  - Sensor uncertainty
    - Sensor readings are imprecise and unreliable
  - Non-observability
    - Various aspects of the environment can not be observed
    - The environment is initially unknown
  - Action uncertainty
    - Actions can fail
    - Actions have nondeterministic outcomes

Situated Robotics

- The science of designing autonomous machines for complex, challenging and dynamically changing environments is called *situated robotics*.
  - I.e. where behaviour is strongly affected by the environment.
- Robots that exist in static, unchanging environments are not thought to be situated.
  - I.e. assembly robots operating in highly structured, fixed, and predictable environments.
- The predictability and stability of the environment has a direct impact on the complexity of the robot that must operate in it.
  - This makes situated robotics a considerable challenge for control engineering.

Autonomous Control Systems

- Autonomous robot control is the process of taking information about the environment through the robot’s sensors, processing it as necessary and making decisions about how to act.
- The complexity of the environment (i.e. the level of situatedness) has a direct impact on the complexity of the control system which is also directly related to the robot’s task.

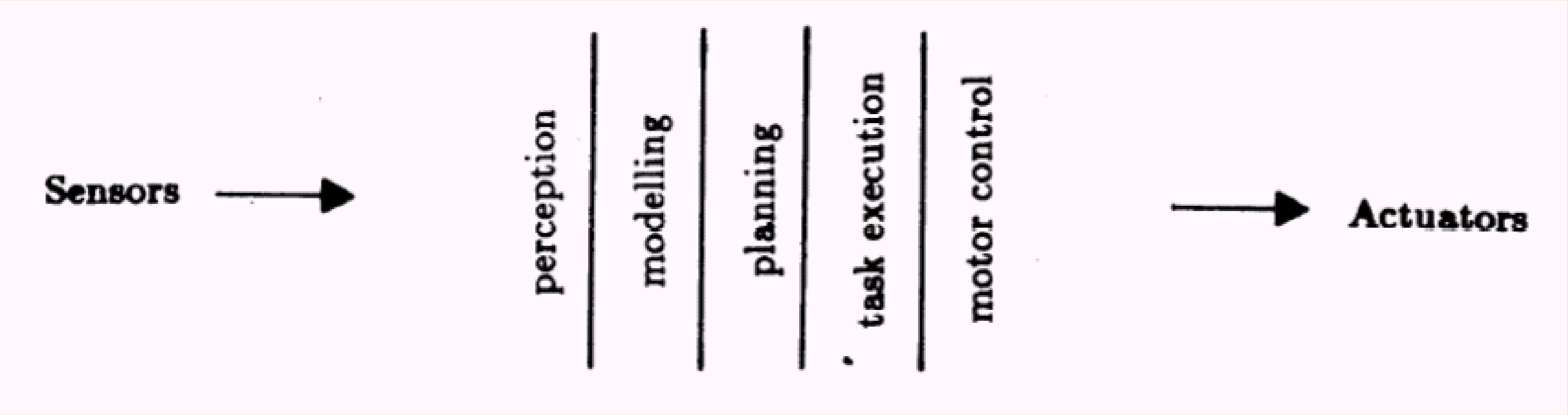
Autonomous Control Systems

- While there is an infinite number of ways to program a robot, there are fundamentally four classes of robot control methodologies:

1. Deliberative *(think, then act.)*
2. Reactive *(don't think, just react)*
3. Hybrid *(deliberative and reactive)*
4. Behaviour based *(parallel competitive bahaviours)*

Derliberative – Think, then act

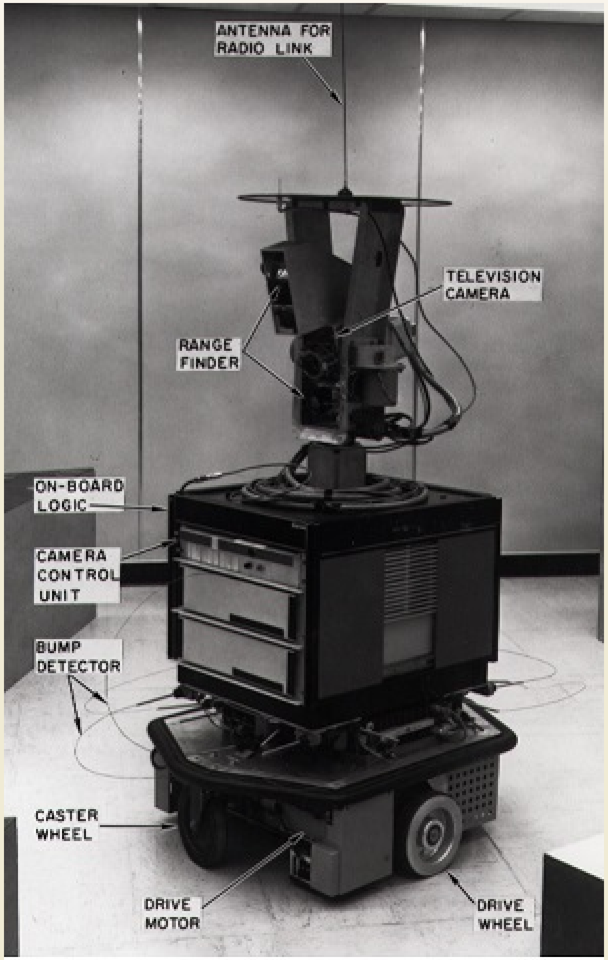
- In a deliberative control architecture the robot first plans a solution for the task by reasoning about the outcome of its actions and then executes it



  - Control process goes through a sequence of sensing, model update, and planning steps.

Derliberative – Think, then act

- Shakey (SRI – Stanford 1969) one of the earliest examples of a deliberative system.
  https://www.youtube.com/watch?v=7bsEN8mwUB8
- Shakey used a Strips planner implemented in Lisp and could avoid obstacles, manipulate objects, and navigate based on vision data using propositional logic.
- The Strips planner would search for a sequence of actions that could achieve the specified task.



STRIPS: Stanford Research Institute Problem Solver

Derliberative – Think, then act

- Deliberative control systems use all the robot available sensory information and all of the internally stored knowledge to reason about what action to take next.
- The control system is a functional decomposition of decision-making processes that are orgainzed into modules:
  - sensory processing module,
  - modelling module,
  - planning module,
  - value judgement module, and
  - execution module.

Derliberative – Think, then act

- Functional decomposition allows complex operations to be performed, but implies strong sequential interdependencies between the decision-making modules.
- Reasoning is typically in the form of planning, requiring a search of possible state–action sequences and their outcomes.

Derliberative – Think, then act

- Planning is a major component of deliberative control systems and is a computationally complex process.
- The planning process requires the robot to perform a sequence of sense–plan–act steps. e.g.
  1. combine the sensory data into a map of the world
  2. use the planner to find a path in the map
  3. send steps of the plan to the robot's wheels
- The robot must construct and evaluate all possible plans from the current position until it finds one that enables it to reach its goal.

Derliberative – Think, then act

- Planning requires the existence of an internal, symbolic representation of the world, which allows the robot to look ahead into the future and predict the outcomes of possible actions in various states so as to generate plans.
- The internal model must be kept accurate and up to date.
- When there is sufficient time to generate a plan and the world model is accurate this approach allows the robot to act strategically by selecting the best course of action for a given situation.
- However, being situated in a noisy non-deterministic world usually makes this impossible.

Derliberative – Think, then act

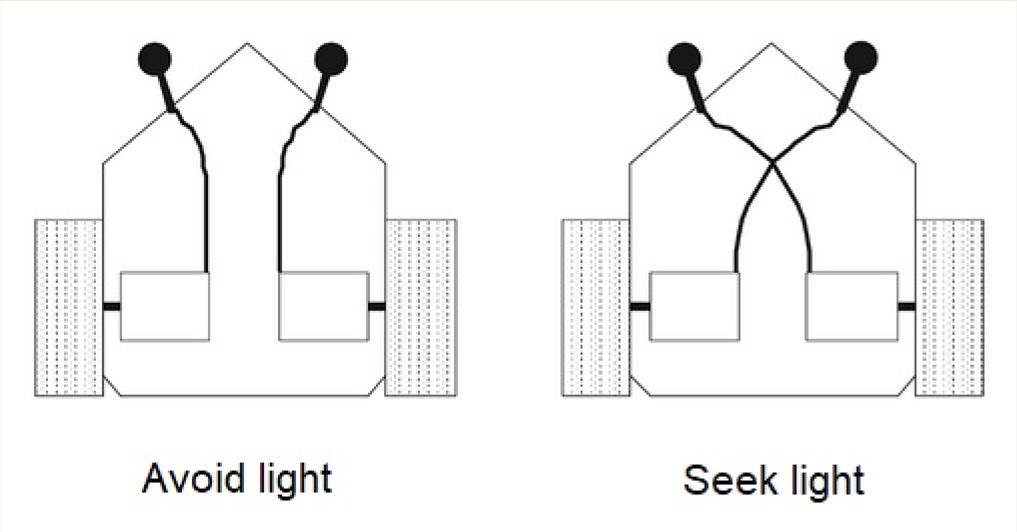
- Advantages
  - Reasoning about contingencies
  - Computes solutions to the given task
  - Goal-directed strategies
  - Robustness to noise and dynamics
- Problems
  - Solution can be fragile in the presence of uncertainty
  - Requires frequent re-planning
  - Reacts relatively slowly to changes and unexpected situations
  - Situated robots cannot be purely deliberative

Reactive – Don’t think, just react

- Reactive control architectures are driven by the need for faster actions in response to the demands of complex and dynamically changing real-world environments.
- Reactive control is a powerful and effective control method that abounds in nature: insects, which vastly outnumber vertebrates, are largely reactive.
- Reactive control has sensory inputs tightly coupled to actuators with no intervening reasoning.
- This allows the robot to respond very quickly to changing and unstructured environments.

Reactive – Don’t think, just react

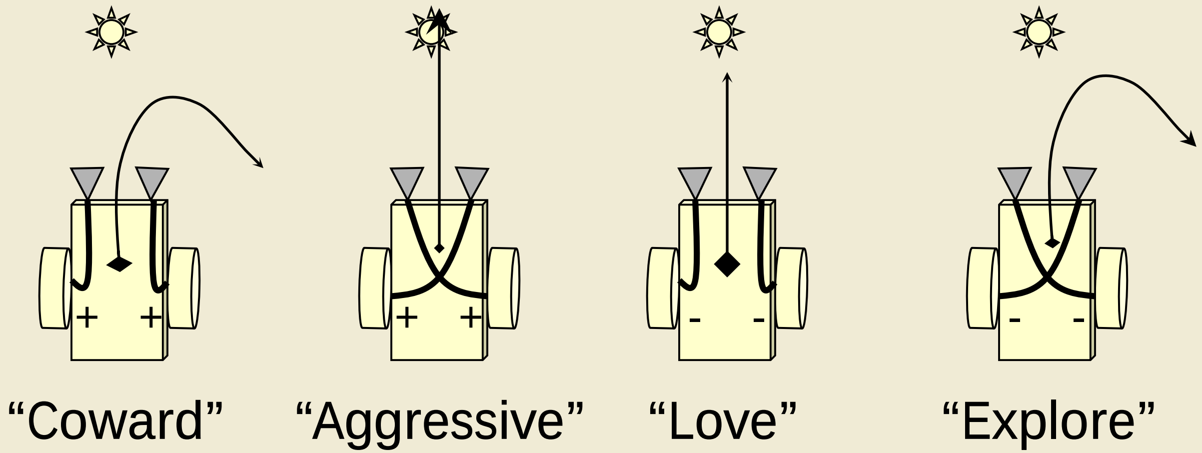
- Reactive control is inspired by the biological notion of stimulus–response. It does not require the acquisition or maintenance of world models as it does not rely on the types of complex reasoning processes utilised in deliberative control. e.g.



Braitenberg Reactive Robots

Reactive – Don’t think, just react

- Braitenberg vehicles: differential drive mobile robots with two light sensors.
- They show that complex behaviors can be achieved using very simple reactive control mechanisms.



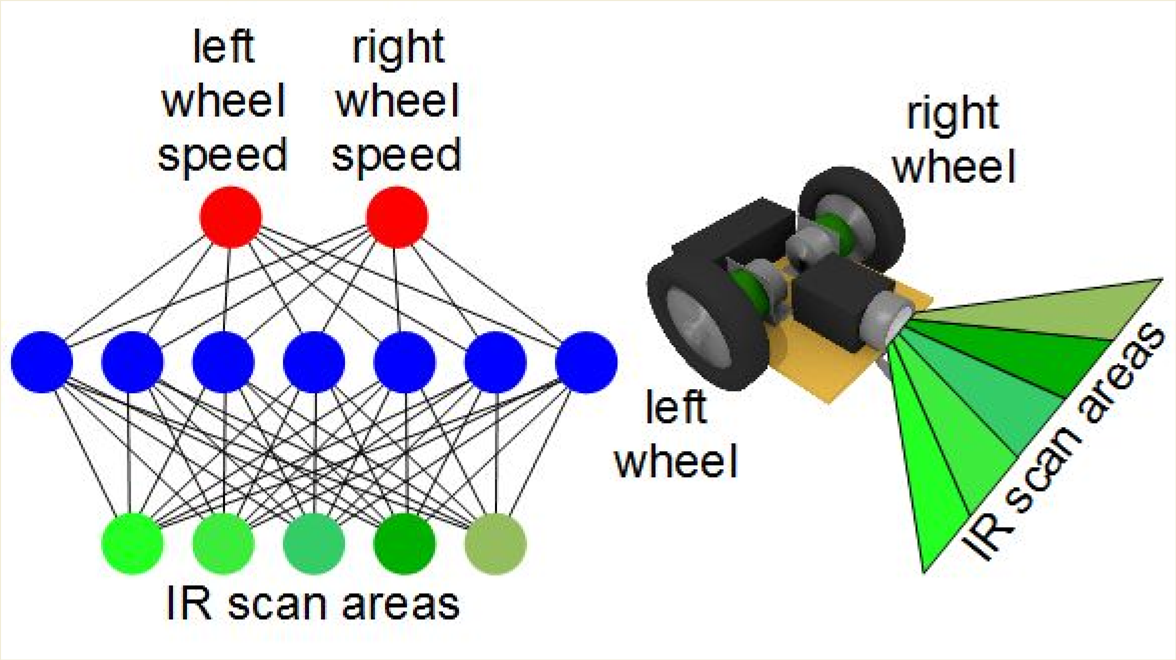
https://www.youtube.com/watch?v=NJo5HEdq6y0&t=242s

Reactive – Don’t think, just react

- Scaling up reactive robots can be difficult.
- More complex reactive robots use fast rule-based methods with no internal representations or knowledge of the world.
- Real-time reactive responses can be achieved by using a collection of pre-programmed concurrent condition-action rules with minimal internal state (e.g., if bumped, stop; if stopped, back up) - typical in maze solving robots.

Reactive – Don’t think, just react

- Artificial neural networks (ANNs) can also be used to achieve appropriate responses to input stimulus.
- ANNs can be trained via remote control (i.e. learning via demonstration), or evolved (using genetic algorithms).



Reactive – Don’t think, just react

- Benefits:
  - Reactive control is well suited to dynamic and unstructured worlds where having access to a world model is not a realistic option.
  - Minimal computation means reactive systems are fast, robust and suitable for minimal hardware platforms.
  - Able to respond in a timely manner to rapidly changing environments.
  - Capable of optimal performance in unstructured environments on tasks that can be characterised a-priori (in advance).

Reactive – Don’t think, just react

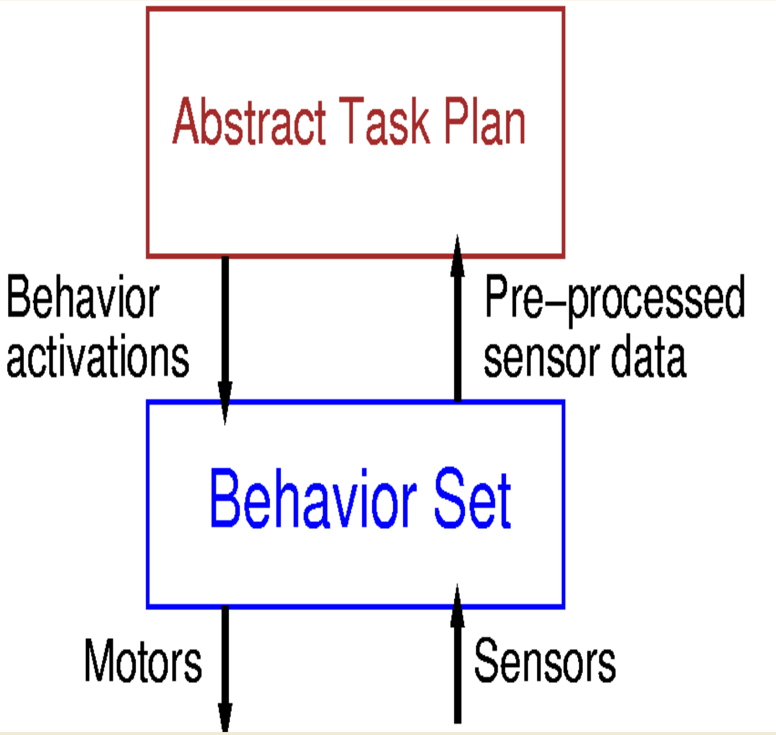
- Disadvantages:
  - Reactive control trades off complexity of reasoning for fast reaction time.
  - Unsuitable for tasks where internal models, memory, and learning are required.
  - Inability to store high-level environment information (e.g. maps), and therefore can’t plan, learn and adapt over time.

Hybrid – think and act concurrently

- Hybrid control aims to combine the best aspects of reactive and deliberative control
  - the real-time response of reactivity and
  - the rationality and optimality of deliberation.
- As a result, hybrid control systems contain two different components,
  - the reactive/concurrent condition–action rules and
  - the deliberative data structures, (which must interact in order to produce a coherent operation).

Hybrid – think and act concurrently

- Hybrid architectures combine reactive control with abstract task planning
  - Abstract task planning layer
    - Deliberative decisions
    - Plans goal directed policies
  - Reactive behavior layer
    - Provides reactive actions
    - Handles sensors and actuators



Hybrid – think and act concurrently

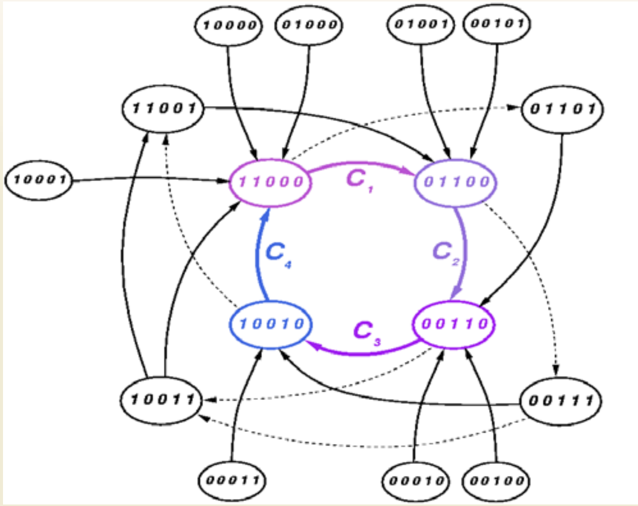
- Hybrid systems have three-layered architectures:
  - reactive (execution) layer,
  - intermediate (co-ordination) layer, and
  - deliberative (organisation/planning) layer.
- Three-layer architectures aim to harness the best of reactive control in the form of dynamic, concurrent, and time-responsive control, and the best of deliberative control, in the form of globally efficient actions over a long time scale.

Hybrid – think and act concurrently

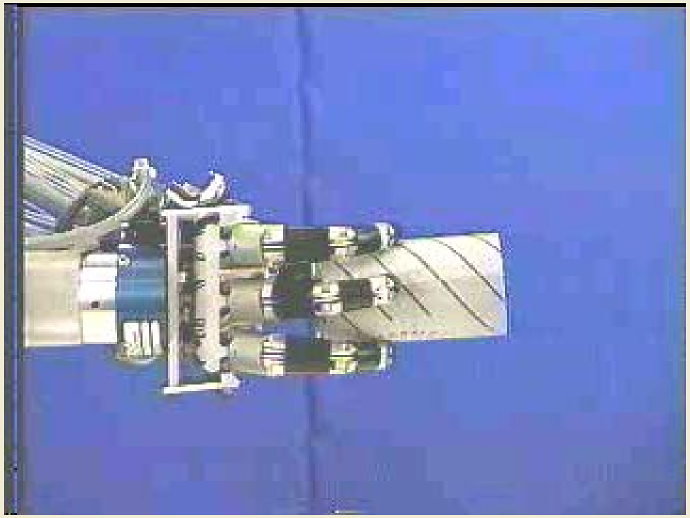
- The three layers are organised with increasing precision of control in the lower layers with decreasing intelligence.
- However, there are complex issues involved in interfacing these fundamentally differing components and the manner in which their functionality should be partitioned is not yet well understood.

Hybrid – think and act concurrently

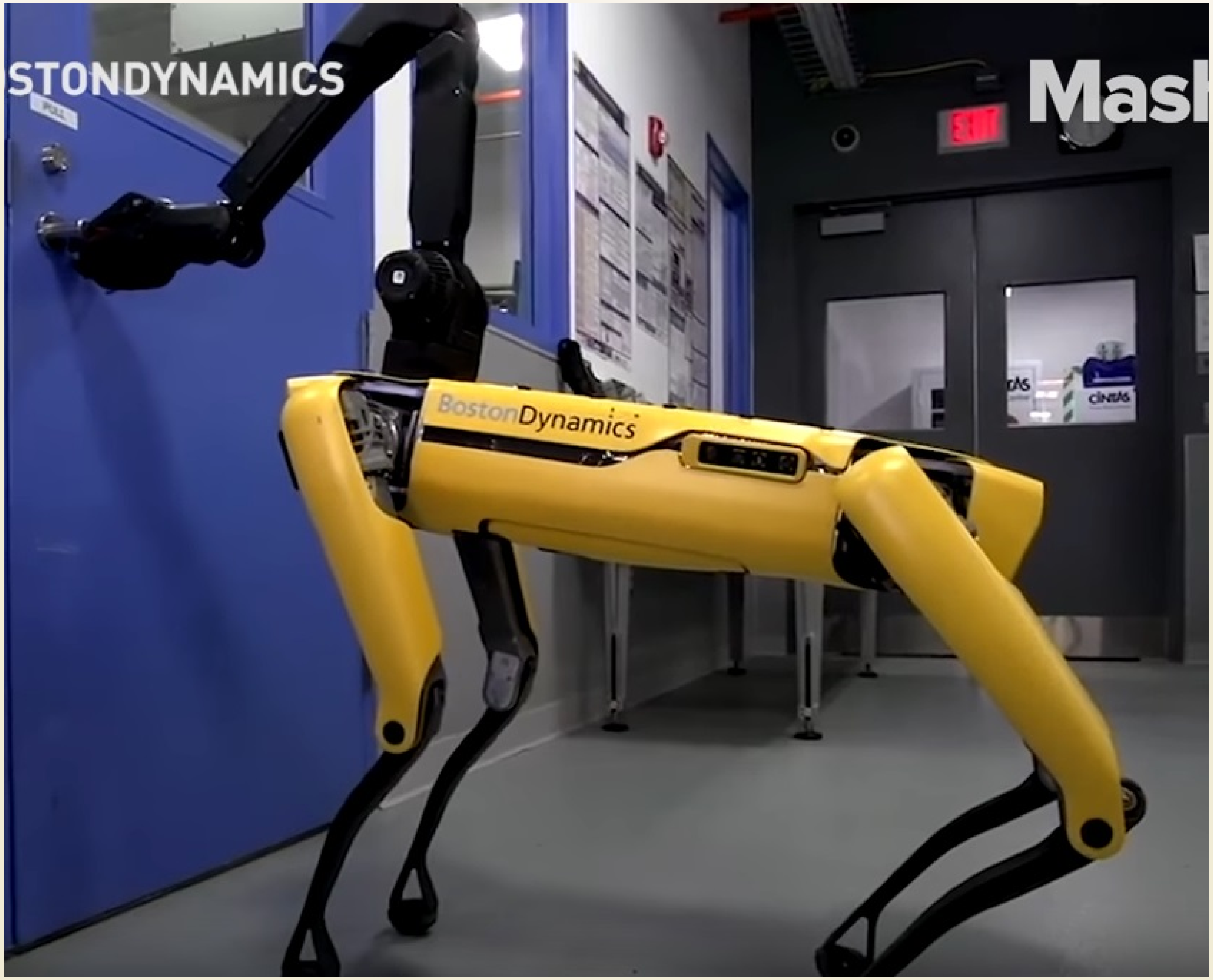
Task Plan:



Behavioral Strategy:



Example: Door opening



https://www.youtube.com/watch?v=Pm8OZ2_-xWI

Example: Door opening

- This is a challenging task because the reactive component deals with the robot's immediate needs, such as moving and grasping while avoiding obstacles, and operates on a fast time scale using direct external sensory data and signals.
- The deliberative component uses highly abstracted, symbolic, internal representations of the world (e.g. door, map, path planning, etc), and operates on a longer time scale.

Hybrid – think and act concurrently

- As long as the outputs of the two components are not in conflict, the system requires no further co-ordination.
- However, the two parts of the system must interact if they are to benefit from each other.
- Consequently, the reactive system must override the deliberative one if the world presents some unexpected immediate challenge.

Hybrid – think and act concurrently

- Likewise, the deliberative component must inform the reactive one in order to guide the robot toward more efficient and optimal trajectories and goals.
- The interaction of the two parts of the system requires an intermediate component, which reconciles the different representations used by the other two and any conflicts between their outputs.
- The construction of this intermediate component is the greatest challenge of hybrid system design.

Hybrid – think and act concurrently

- Advantages:
  - Permits goal-based strategies
  - Ensures fast reactions to unexpected changes
  - Reduces complexity of planning
- Problems:
  - Choice of behaviors limits range of possible tasks.
  - Behavior interactions have to be well modeled to be able to form plans.
  - Complexity.

Behaviour-based Control – think the way you act

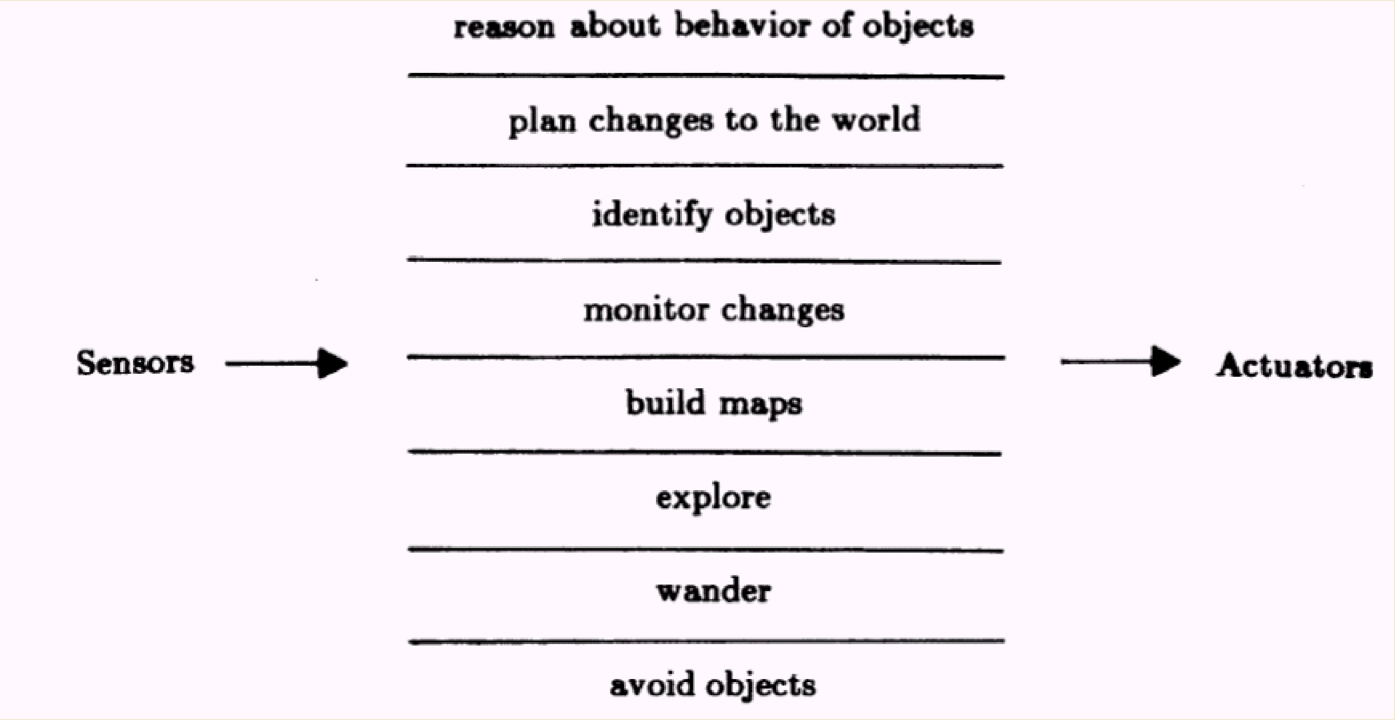
- Behaviour based control employs a set of distributed, interacting modules, called behaviours, that collectively achieve the desired system-level behaviour.

  Example: https://www.youtube.com/watch?v=oJq5PQZHU-I

- Each behaviour receives inputs from sensors and/or other behaviours in the system, and provides outputs to the robot's actuators or to other behaviours.
- There is no centralised world representation or focus of control.
- Instead, individual behaviours and networks of behaviours maintain their own required state information and models.

Behavior-Based Control

- In a behavior-based control architecture the robot's actions are determined by a set of parallel, reactive behaviors which map sensory input and state to actions.



Behaviour-based Control

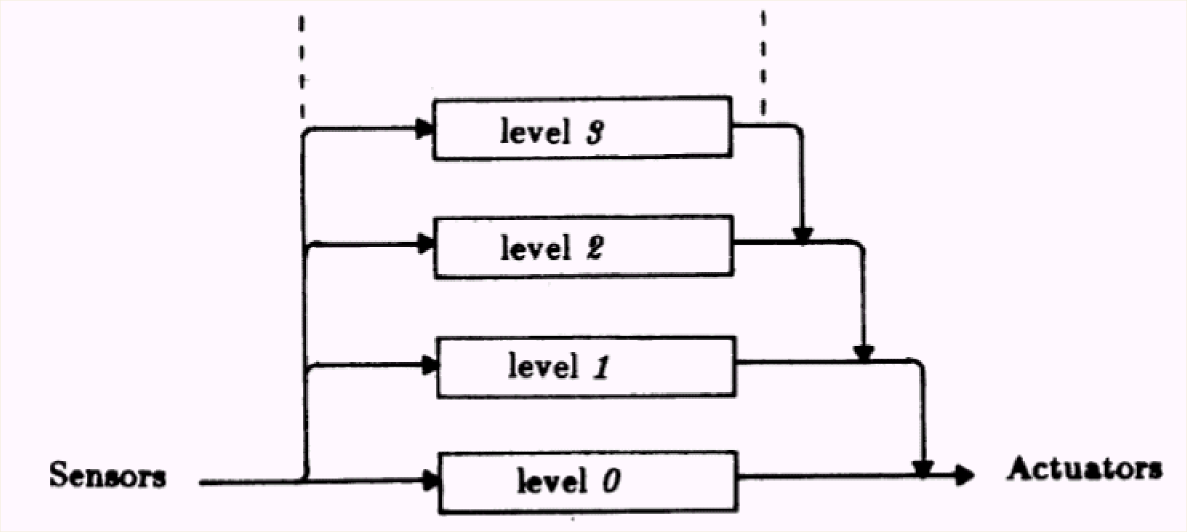
- To an external observer, behaviours are patterns of the robot’s activity emerging from interactions between the robot and its environment.
- To a programmer, behaviours are control modules that cluster sets of constraints in order to achieve and maintain a goal.

Behavior-Based Control

- Reactive, behavior-based control combines relatively simple behaviors, each of which achieves a particular subtask, to achieve the overall task.
  - Robot can react fast to changes
  - System does not depend on complete knowledge of the environment
  - Emergent behavior (resulting from combining initial behaviors) can make it difficult to predict exact behavior
  - Difficult to assure that the overall task is achieved

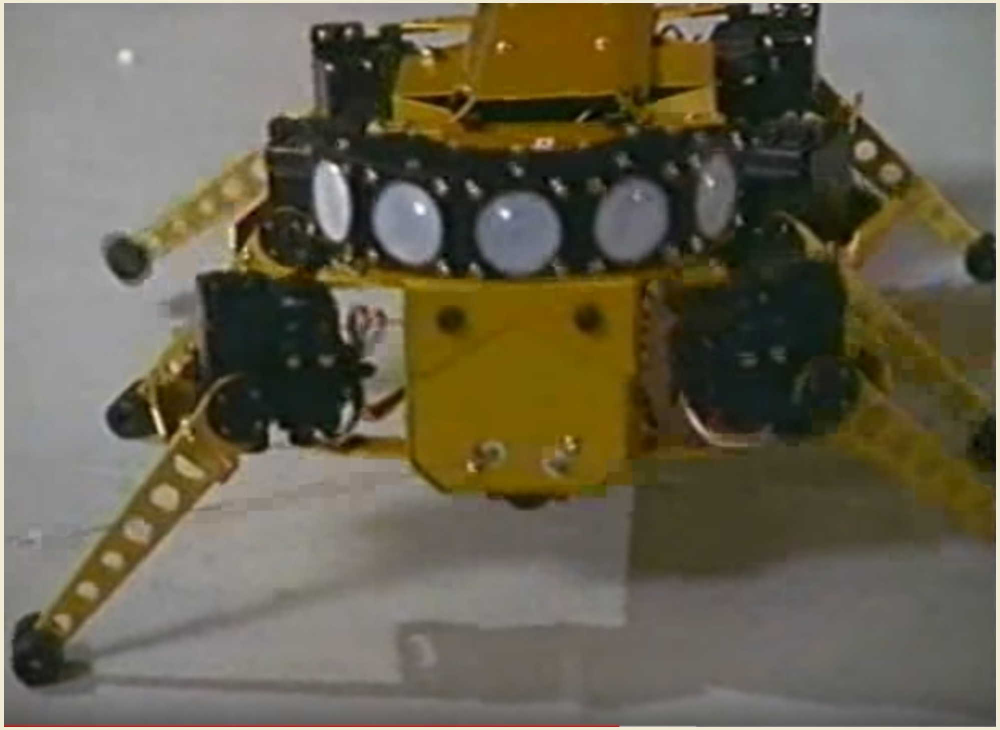
Behavior-Based Architectures: Subsumption Example

- Subsumption architecture is one of the earliest behavior-based architectures
  - Behaviors are arranged in a strict priority order where higher priority behaviors subsume lower priority ones as long as they are not inhibited.



Subsumption Example

- A variety of tasks can be robustly performed from a small number of behavioral elements



https://www.youtube.com/watch?v=9u0CIQ8P_qk

Behaviour-based Control – think the way you act

- Subsumption-based systems take advantage of the dynamics of interaction among the behaviours themselves, and between the behaviours and the environment.
- The functionality of behaviour-based systems can be said to emerge from those interactions and is thus neither a property of the robot or the environment in isolation, but rather a result of the interplay between them.

Behaviour-based Control – think the way you act

- Unlike reactive control, which utilises collections of reactive rules with little if any state and no representation, behaviour-based control utilises collections of behaviours, which have no such constraints.
- But behaviours can have state and can be used to construct representations thereby enabling reasoning, planning, and learning.

Behavior-Based Control

- Advantages
  - Reacts fast to changes
  - Does not rely on accurate models
    - “The world is its own best model”
  - No need for replanning
- Problems
  - Difficult to anticipate what effect combinations of behaviors will have.
  - Difficult to construct strategies that will achieve complex, novel tasks.
  - Requires redesign of control system for new tasks.

So which is best?

- Each approach to robot control has its strengths and weaknesses.
- All play important and successful roles in certain robot control problems and applications.
- Each offers interesting but different insights, and no single approach should be seen as ideal or absolute.
- The choice of robot control methodology should be based on the particular task, the environment, and the robot.

So which is best?

- For example, reactive control is the best choice for environments demanding immediate response.
- However, such speed of reaction comes at the price of being shorted sighted - not looking into the past or the future.
- Reactive systems are also a popular choice in
  - highly stochastic environments, and
  - environments that can be characterised as having an obvious reactive input output mapping (e.g. a light seeking robot).

So which is best?

- Deliberative systems are the only choice for domains that require a great deal of strategy, optimisation and planning.
- Such domains, however, are not typical of situated robotics, but more so of scheduling, game playing, and system configuration, among others.
- Hybrid systems are well suited for environments and tasks where internal models and planning are needed, and the real time demands are few, or sufficiently independent of the higher-level reasoning.

So which is best?

- Behaviour-based systems, in contrast, are best suited for environments with significant dynamic changes, where fast response and adaptivity are crucial, but the ability to do some looking ahead and avoid past mistakes is required.
- Those capabilities are spread over the active behaviours, using active representations if necessary.

So which is best?

- As a consequence of these inherent trade-offs, it is important to have different methodologies rather than having to fit all controller needs into a single methodology.
- Selecting an appropriate control methodology and designing an architecture within it is best determined by the situatedness properties of the problem, the nature of the task, the level of efficiency or optimality needed, and the capabilities of the robot, both in terms of hardware, world modelling, and computation.

Summary

- Situated robotics deals with autonomous machines in complex and dynamically changing environments
- There are four classes of robot control methodologies:
  - Deliberative (think, then act.)
  - Reactive (don’t think, just react)
  - Hybrid (deliberative and reactive)
  - Behaviour based (parallel competitive bahaviours)
- Subsumption architecture arranges behavious in a strict priority order where higher priority behaviors subsume lower priority ones as long as they are not inhibited.