Adaptive Systems

Ezequiel Di Paolo
COGS

Behaviour-based Robotics

An action-oriented approach

- **Behaviour-based robotics** describes a design methodology for robots based mainly on Brooks' layered architecture (1986), but applicable also to other areas (such as schema-based design). As other action-oriented approaches it emphasizes the relevance of situatedness, embodiment and dynamics.
- Roughly inspired in biology.

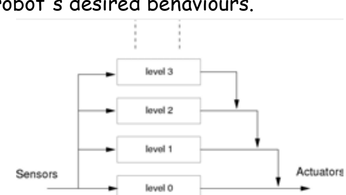
- Everyday action in real environments (low level, but interesting capabilities, e.g., safe navigation),
- Fast, reactive (or quasi-reactive) behaviour.
- Interaction between levels of competence
- Questioning the assumptions behind the traditional design methodology
- Avoiding complex processing involved in maintaining world models
- Precursors: Grey Walter, Moravec, Raibert

Instead of …

Why not …

Levels of competence

- Conceptual design representing a specification of the robot's desired behaviours.



- They separate what the robot should be able to do into levels. E.g., robot should be able to avoid obstacles, but also to explore, pick up interesting objects, etc.

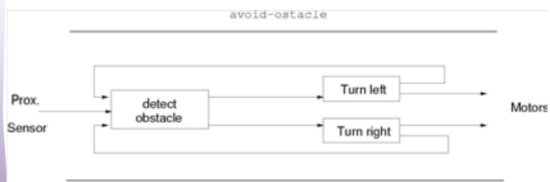
Key points

- Each *level of competence* is implemented as a *control layer* in the robot architecture
- Layers can be built incrementally from the most basic level up.
- Competences remain present all the time, even though they can be inhibited/modified under special circumstances.

Layers of control

- Layers composed of simple asynchronous modules (finite state machines in original cases).
- Build more basic layers first.
- Make them work. (Debug them.)
- "Freeze" them
- Build next layer on top of existing ones

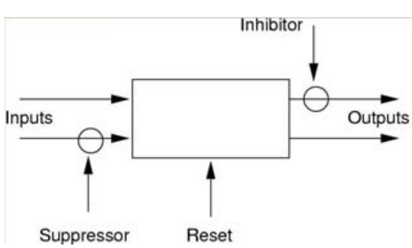
Obstacle avoidance



Coordination

- Coordination between layers through:
  - **Inhibition:** Outgoing signals from a module are inhibited
  - **Reset:** The internal state of a module is restarted
  - **Suppression:** the input to a module is suppressed and replaced by a different signal

Coordination



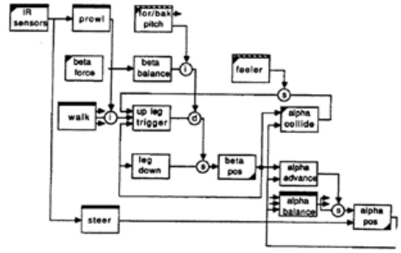
Coordination

- Low-bandwidth internal communication
- Message passing via machine registers
- Fixed topology (no plasticity)
- Output of lower layer accessible by higher levels (but not the other way around in principle)
- Effective communication through the world itself.

Hexapod robot

- Genghis (1989): walks over rough terrain, follows people. 12 motors, 12 force sensors, 6 pyroelectric sensors, 1 inclinometer and 2 whiskers.

- Distributed control, layers working in parallel.



- (Triangles at the top = sensor, triangles at the bottom = effectors, bars = coordinating modules for all 6 legs. Alpha = advance, Beta = balance)

Schema-based robotics

- R. Arkin *Behaviour-based robotics*, MIT Press, 1998).
- Reactive control -- Schema-based
- **Schema:** Basic unit of motor behaviour from which complex actions can be constructed.
- Motor schemas (behaviours) are selected to enable the robot to interact successfully with unexpected events while satisfying higher-level goals.

Schema-based robotics

- No explicit planning
- Multiple active schemas present, each produces a velocity vector driving the robot in response to sensory stimulus.
- Vectors are summed to give a single velocity.
- Continually updated. No explicit arbitration between schemas (as in subsumption architecture)

Examples of schemas

`avoid-static-obstacle`

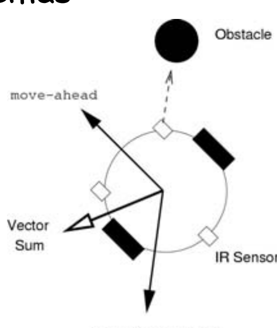
`move-to-goal`

`stay-on-path`

`noise` (move at random)

`probe` (move towards most open space)

`escape`

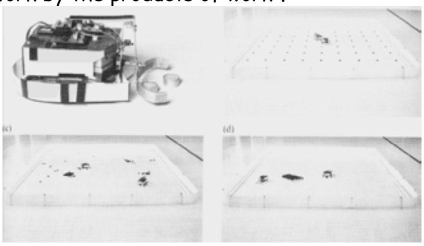


Parameters for schemas

- E.g., `move-to-goal`, `avoid-obstacle`, and `noise`, controlled by five parameters:
  - goal gain - strength for approaching goal
  - obstacle gain - strength with which move away from obstacle
  - obstacle sphere of influence - distance from obstacle at which robot is repelled
  - noise gain - amplitude of random wandering
  - noise persistence - number of time steps the noise vector is held constant
- Schemas largely hand-designed but GAs can be used to tune parameters effectively.

Collective action

- Beckers et al. 1994. Inspiration from social insects. Stigmergy (Grasse, 1959) "incitement to work by the products of work".



Collective action

- Basic behaviours:
  - Move in a straight line
  - Avoid hitting into other robots and walls by random turning
  - Push up to 3 pucks (Mind the body!)
  - Drop them if pushing more than three

Precursors

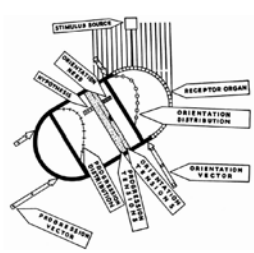
- Grey Walter Tortoises (50s) (Holland, 1996)
- Tolman's Sowbug (1939) (Endo & Arkin, 2001)
- Braitenberg vehicles (80s)

Grey Walter

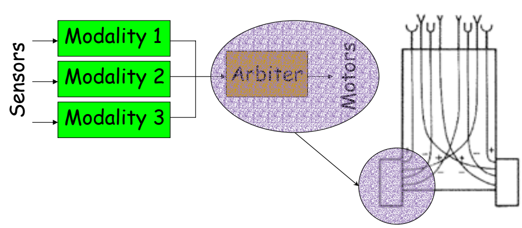
Principal behaviours:
- Cycloidal wandering (both motors)
- Phototaxis (driving motor only)
- Dazzle (both motors active at diff. level)
- Avoid obstacle (similar to dazzle)

Tolman's sowbug

- 1939! Braitenberg like architecture with internal states such as motivation
- Orientation and Progression vectors
- Able to do simple learning



B-Vehicles as schema-based



Is BBR reactive?

- Most of the time it is considered to be so.
- However, internal modules in the control layers may change state so there is no in-principle reason why the architecture as it is should be confined to reactive behaviour.
- It is mostly a question of timescales. Internal modules change state at a relatively fast timescale. Hence little chance of long-term plasticity.

Questions

- How will this methodology scale?
- Design gets harder as you try to scale up. Can't keep track of the effect of different parameters.
- How about human level cognition? (In fact, how about a dog, an iguana, a fly?)
- Should we care?

Questions

- D. Kirsh ("Today the earwig, tomorrow man?"): concepts equal symbolic computation, a non-symbolic approach cannot get you to human level intelligence.
- Brooks ("From earwigs to humans") recognizes problems, but bets on soundness of approach. Human intelligence requires a humanoid body, (COG project and others).
- Is artificial evolution a possible answer to some of these problems?