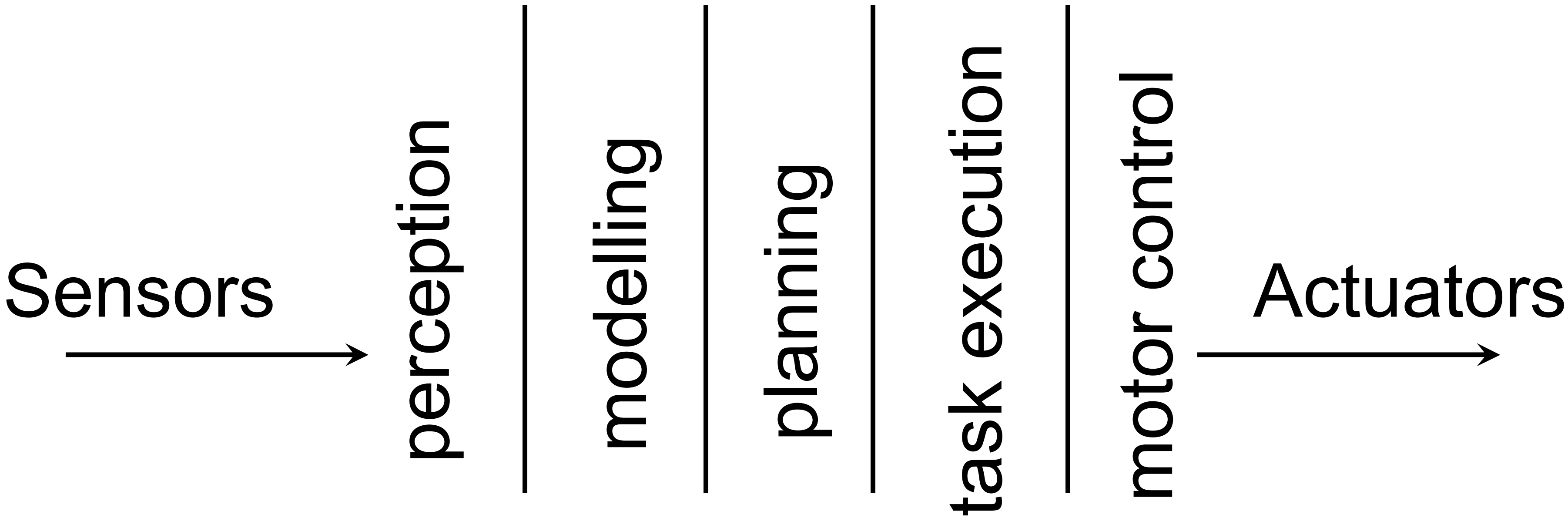
Behaviour-Based Control

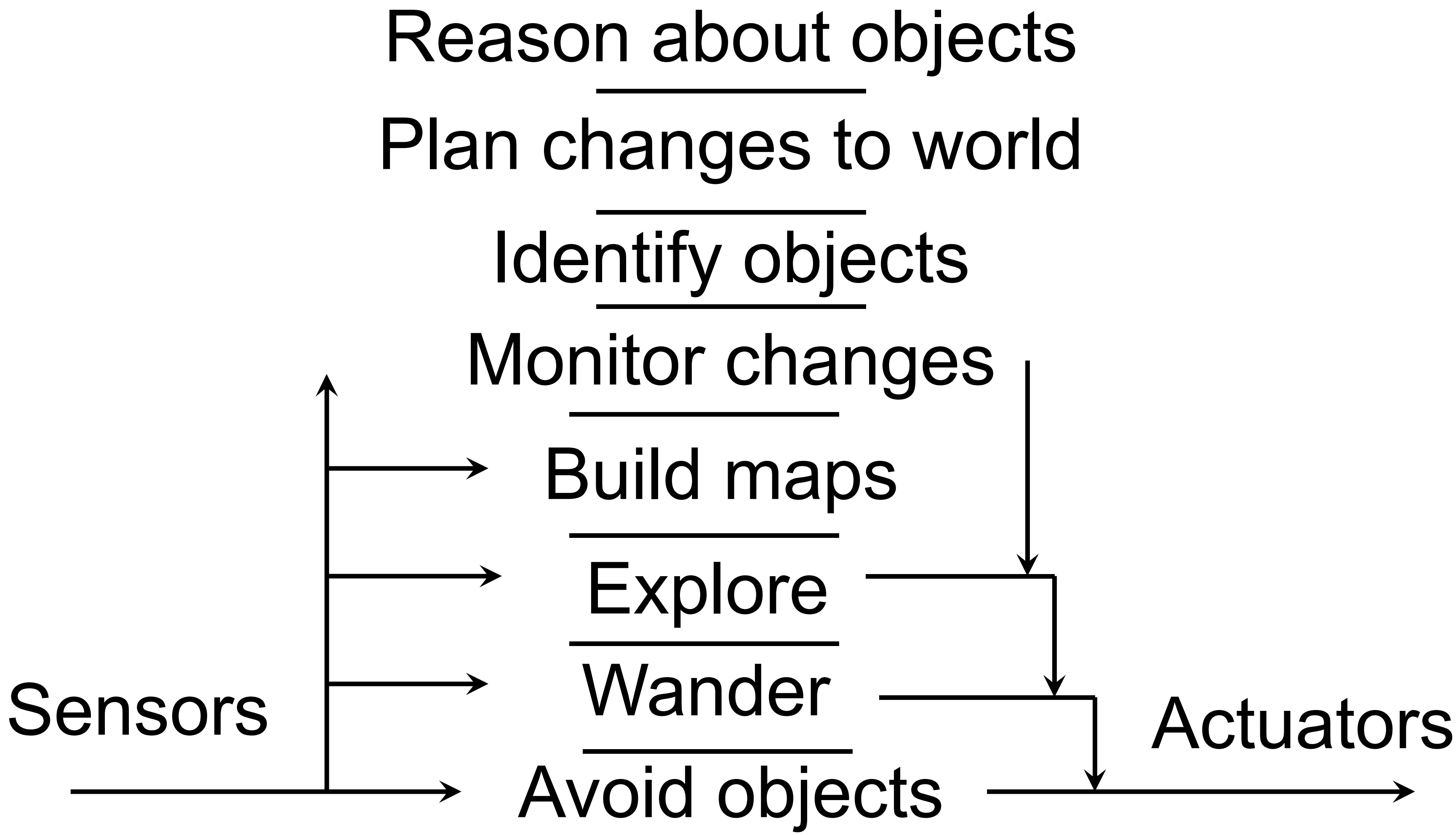
IAR Lecture 5

Barbara Webb

Traditional sense-plan-act approach suggests a ‘vertical’ (serial) task decomposition



Alternative is a 'horizontal' (parallel) task decomposition



Some advantages of the horizontal approach:

- Don’t have to build all parts before testing on robot
- Immediate appreciation of effects of embodiment and situatedness, possibility of reactive solutions
- Multiple goals pursued in parallel, late decision
- Multiple sensors without requiring fusion
- Each layer adds competence to already working robot – graceful degradation if higher level fails
- Can map onto hardware e.g. new processors for each new level of behaviour - additivity

Some problems of the horizontal approach:

- Hard to predict/verify the emergent outcome of the combined behaviours
- Have to decide how behaviours will interact:
    - Hierarchical inhibition (e.g. subsumption architecture)
    - Mixed outputs (e.g. motor schema architecture)
    - Action selection mechanism:
        - through bottom up emergent process
        - through top-down control (see hybrid control lecture)

The behaviour-based approach

- Robot architecture is designed around a collection of behaviours:
  - Typically, these are reactive, or use only local memory and minimalist representations
  - Exploit physics and environmental interaction
- Each behaviour should function on its own as a complete sensorimotor loop in the real world (i.e. modular, but always building a complete agent)

E.g. subsumption architecture proposed by Brooks (1986)

- Decompose problem into layers of competence
- Each layer uses sensors, actuators and control
- Build and debug lowest layers first
- Add new layers without changing lower ones
- New layers will ‘subsume’ the output of lower layers

E.g. ‘Polly’ – Horswill, 1989

Provide Tour

Recognise Person

Navigate

Avoid hazards

Motor schema architecture proposed by Arkin (1989)

- Schemas: independent asynchronous processes taking sensor inputs and generating velocity vectors
  - Move-to-goal: $V_{magnitude} = fixed\ gain, V_{direction} = perceived\ goal$
  - Avoid-static-obstacle:

$$O_{magnitude} = \begin{Bmatrix} 0\ for\ d > S \\ \frac{S-d}{S-R} * G\ for\ R < d \le S \\ \infty\ for\ d \le R \end{Bmatrix}$$

$$O_{direction} = line\ from\ obstacle\ centre\ to\ robot$$

  - Noise: $N_{magnitude} = fixed\ gain, N_{direction} = random\ change\ every\ p\ steps$
- Motor fusion: A weighted sum of the vectors determines the current reaction of the robot

Motor schema architecture

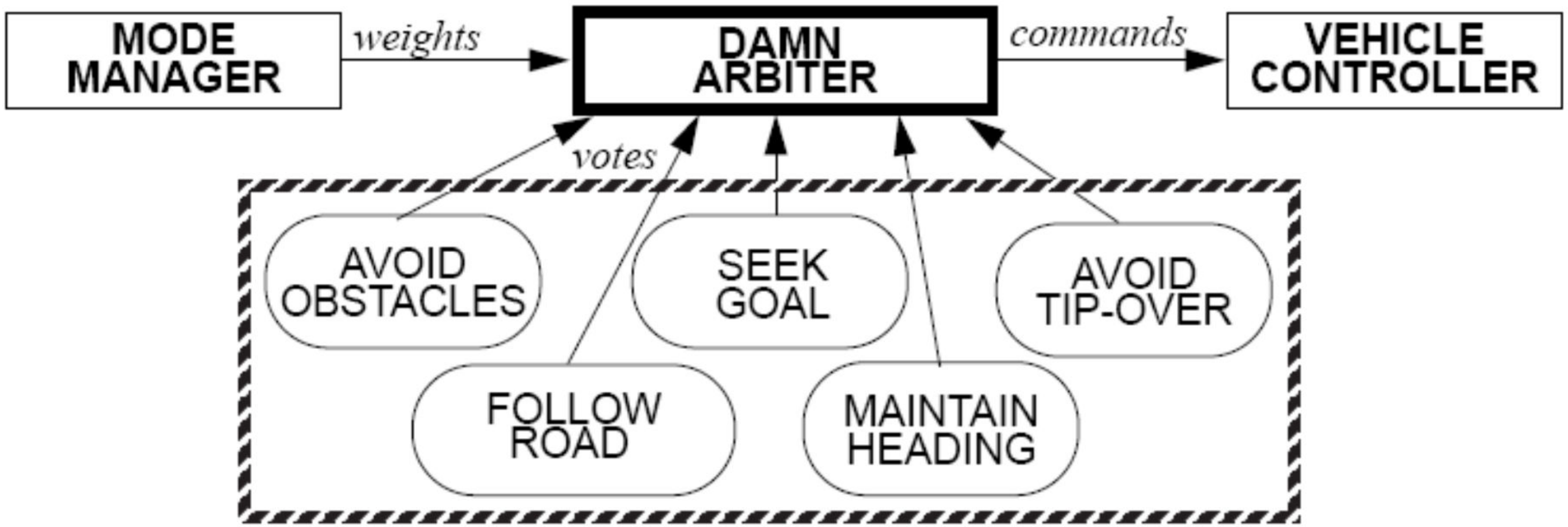
- Similar to potential field method, but only ever calculate the local vectors
- Relative gain of each behaviour can be varied according to robot's current mission state
  - E.g. high gain noise for exploring, low gain noise when approaching goal (keeping some noise as 'behavioural grease' to avoid local minima)

Action selection architectures

- A fixed hierarchy (as in subsumption) has to be predetermined and is not flexible to opportunities
- Motor fusion does not always produce the appropriate responses (particularly if behaviours conflict)
- Alternative is to have some form of ‘action selection’ mechanism between behaviours, e.g.:
  - Winner take all network
  - Fuzzy logic
  - Multiple objective optimisation
  - Reinforcement learning or other adaptive methods
- More recent/complex systems may use several selection mechanisms, according to task and situation

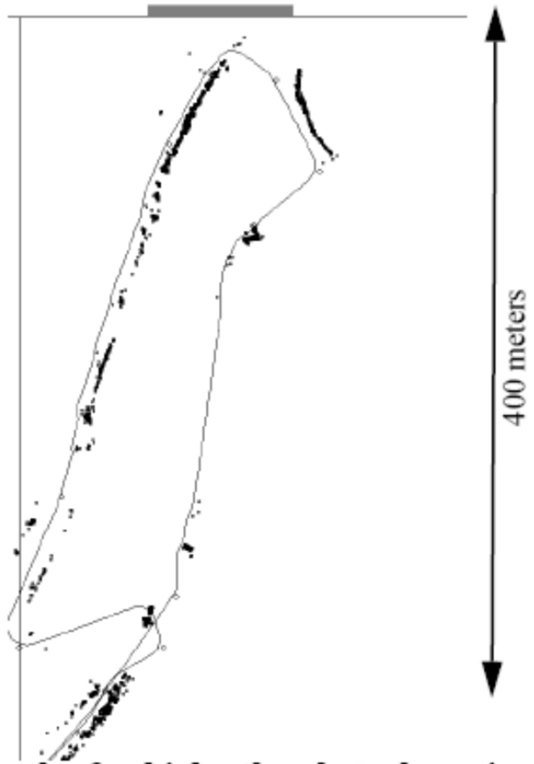
DAMN (Distributed architecture for mobile navigation) proposed by Rosenblatt (1995)

- As before have set of parallel asynchronous behaviours producing possible action outputs
- Final output determined by 'arbiters' that count the weighted votes for each action



DAMN used on NAVLAB

(a) Camera view of terrain with approximate path superimposed.



(b) Exact path of vehicle: the obstacle regions are shown as black dots; the intermediate goal points are shown as small circles.

Behaviour based approach: Conclusions

- Produced some very robust and successful robots:
  - Still very widely used in robot and agent approaches
  - But no continuous evolution to higher capabilities…?
- Wide influence across AI and related fields:
  - Importance of embodiment and situatedness; solving problems with physics and hardware as well as software
  - Possibilities for low-level sensorimotor coupling, exploiting environments, emergent behaviours
  - Use of world rather than internal representations
  - New focus on action selection as critical problem to solve

Test question

- What physical design issues are exemplified in the roomba?

Conclusions

- But arbitrary and difficult to design emergent behaviour for a given task.
  - Architectures do not impose strong constraints
- Options?
  - Build up toolbox of techniques
  - Use learning or evolutionary methods
  - Copy existing systems (i.e. biology)
  - Formalise interactions as dynamical systems
- Difficult to do some traditional (and useful) tasks.
  - Increasingly common to adopt ‘hybrid’ approach, e.g. classical planner operating on top of basic behaviours

References

Brooks, R. A., "Elephants Don't Play Chess", Robotics and Autonomous Systems (6), 1990, pp. 3–15.

Brooks, R. A. "A Robust Layered Control System for a Mobile Robot", IEEE Journal of Robotics and Automation, Vol. 2, No. 1, March 1986, pp. 14–23; also MIT AI Memo 864, September 1985.

Available from http://people.csail.mit.edu/brooks/publications.shtml

Horswill, I. (1993) **Polly: A Vision-Based Artificial Agent** Proceedings of the 11th National Conference on Artificial Intelligence (AAAI-93)

Arkin, R.C. (1989) Motor schema based mobile robot navigation. International Journal of Robotics Research, 8:92-112

Rosenblatt, J.K. (1997) DAMN: a distributed architecture for mobile navigation. Journal of Experimental and Theoretical Artificial Intelligence, 9:339-360