Adaptive Systems

Ezequiel Di Paolo
Informatics

Evolutionary Robotics
2. The Reality Gap

Overview

- Artificial Evolution of robot controllers implies the repetitive testing of thousands of individual architectures.
- Success in simulated evolution does not guarantee successful transfer to reality.
- Simulations cannot model everything, nor can they accurately model anything.

Overview

- More accurate and encompassing simulations are costly, problem-specific, and slow.
- A methodology using clever, fast and cheap simulations can come to the rescue.
- It breaks with traditional engineering view of uses for simulations.
- It works.

Challenges for ER

- Mataric & Cliff (1996)
- Evolution in physical robots:
  - Real time → prohibitively long
  - Limited battery lifetime, tethering not always feasible
  - Limited robot lifetime

- Evolution in simulations:
  - Accurate simulations are costly and slow
  - Abstractions and simplifications may be reliably exploited by evolution
  - Careful empirical validation is needed
  - More accuracy implies less general validity and re-usability
- Not clear that ER is a viable alternative to human design for complex behaviours beyond the proof of concept level.

The Reality Gap

- **Successful transfer from simulation to reality:** The robot controller performs the task it's been evolved to perform in reality. Subjective but unambiguous criterion. (No real need to compare fitness values between simulations and reality as this comparison may not be significant)
- Consensus seems to be that success depends on careful construction and empirical validation of a simulation with the right amounts of noise (equivalent to the real case). Typically performance degrades after transfer.

What's a simulation for?

- Simulations have different aims.
- **Engineering:**
  - Used as stand-in for real devices/processes.
  - Accuracy often an aim.
  - Testing of designs, control strategies, training, etc.

- **Science:**
  - Used as modelling tools.
  - Aiming not at fine details but at capturing essential factors.
  - Explanatory purpose.
- **What is a simulation for in the case of ER?**
  - Often none of the above

Better simulations?

- Simulations cannot model everything.
  - Even constrained to factors that are likely to affect the behaviour of a robot, a simulation will always leave some things out.
- Simulations cannot accurately model anything.
  - Factors that are included in a simulation will not be an accurate replica of those same factors in reality.
  - Classical engineering approach: recognise these limitations but still strive for maximum accuracy and detail within cost constraints.

Minimal simulations

- Work by Nick Jakobi (1997/98), methodology currently used by Sussex group.
- **Main problem with evolving using simulations:** evolution relies on implementation details that remain constant in the simulation but are likely to vary in the real world.
- **Motivation:** build cheap, easy-design and fast simulations for robot evaluation.

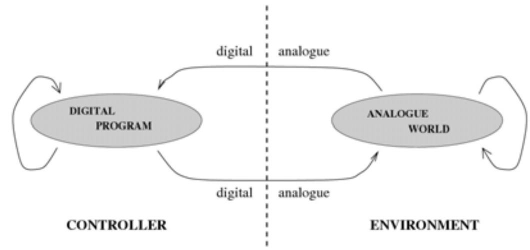
- What the simulation should do is **not** to accurately model the real world but to model *non-relevant* aspects of the world crudely and with large amounts of variation so that the only effective evolutionary strategy is to ignore those aspects.
- A base set of robot-environment relations that are deemed sufficient to underlie the desired behaviour must be identified. These must be kept relatively constant (but including noise and between trial variation to enhance robustness).
- Every other implementation detail must be subjected to **large** amounts of variation from one trial to the next.

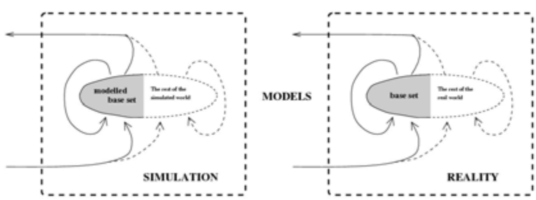
Methodology

- Precisely define behaviour (both tasks to be performed and relevant environmental conditions)
- Identify real-world base set (relevant aspects that affect behaviour and how they interact)
- Model base-set factors and how they react to control
- Model how base-set factors affect controller's input

Methodology

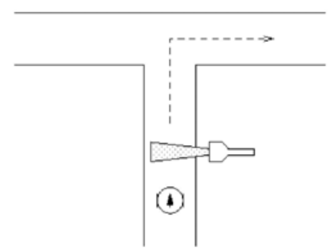
- Design suitable fitness test
- Ensure controllers are base-set exclusive (i.e., they do not rely on any implementational factor; radically vary those factors from trial to trial)
- Ensure controllers are base-set robust (introduce moderate trial-to-trial variation on base-set factors)





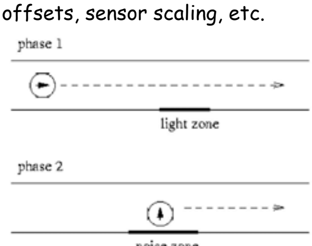
T-Maze

- A delayed response task: make a turn in the same direction as where the light came from.



T-Maze

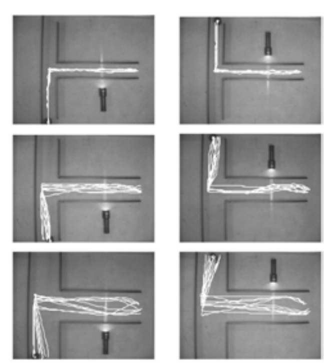
- Minimal simulation: 2 look-up tables, 2 phases, junction not modelled. Variations in corridor width, motor offsets, sensor scaling, etc.



T-Maze

Fitness function:
Distance travelled along each corridor plus 100 bonus when turning in right direction.

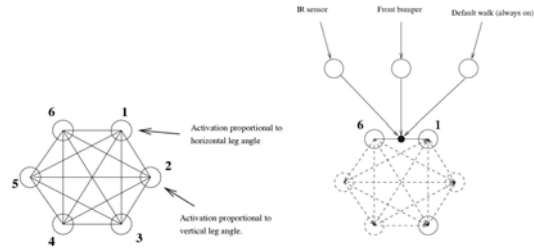
Results:
1000 generations in 4 hours equivalent to 14 months of real time



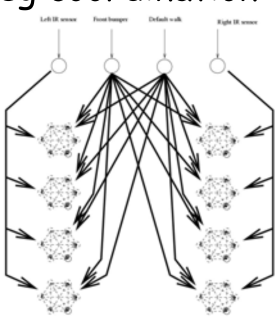
Octopod

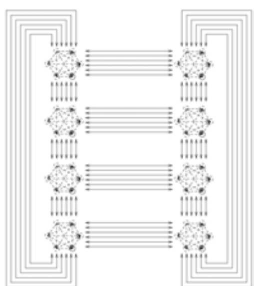
- Walk straight as fast as possible. Turn left on the spot when obstacle appears on right-hand side and vice versa.
- Walk backwards if front bumper is hit.
- Distributed control using Continuous-Time Recurrent Neural Networks with gating of synapses.
- A good simulation of all actuators would be very hard (model collisions, clashes, drag, friction, 3-D position of the body, etc.)

Leg controller



Leg coordination



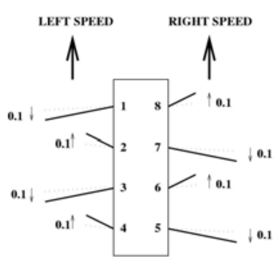


Octopod minimal simulation

- Two main variables: speed of right-hand side and speed of left-hand side.
- Robot stable if centre of gravity contained within polygon formed by legs touching the ground.
- Robot standing or dragging belly depending on height of legs
- Fitness penalises dragging body and instability.
- Each trial the four sensory scenarios are tested.

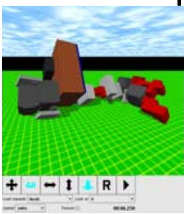
Octopod minimal simulation

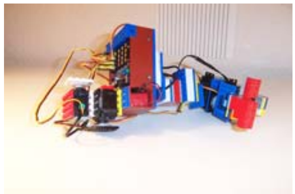
- Legs contribute to side velocity in proportion to how near the ground they are, (**totally unphysical** condition but when maximized leads to desired behaviour)
- Good results after 3500 generations (14 hours of simulation ~11 weeks of real time)



Octopod video

Recent Example





- Ian Macinnes work on brain body joint evolution
- Inter-trial variation in body and network parameters, body/neuron relative update rates, motor velocities, etc.

Others…

- Gantry, triangle-seeking robot.
- Evolving saccadic, tracking camera.
- GasNets
- Group behaviour (Matt Quinn)
- More general methodology for fitness function design for robust controllers

Conclusions

- **What is a simulation for in Evolutionary Robotics?**
- Neither scientific modelling, nor accurate reproduction. It's for *shaping an evolutionary process* by making implementational factors sufficiently unreliable so that they can be safely ignored when designing the simulation.
- The methodology of minimal simulations attempts to do precisely this.

Conclusions

- A minimal simulation does not attempt to reproduce reality. It often is quite unrealistic and unphysical.
- Successful behaviour relies only on base-set factors. Between trials variations makes implementation factors unreliable and controllers base-set robust.
- The aim of a minimal simulation is to be fast and cheap and yet constrain evolution in the right way.

Not problem-free

- Base set may not be so easily identifiable.
- Large amounts of variations make the search difficult. May even make it so difficult that it is not possible to evolve successful controllers unless variation is reduced.
- Method works, but practical aspects of methodology are still more like an art form.
- Evolution is deprived of its opportunistic ability. (Makes engineering sense, but maybe not scientific sense, it depends.)