Adaptive Systems

Inman Harvey
Informatics

Lecture 12 + 13:
Evolutionary Robotics Part 2

This lecture …

- Simulations
- Minimal simulations
- Examples: T-maze, 8 legged walking
- Physics engine, ODEs, etc.
- Evolving plastic controllers.
- Evolving body morphologies/models.
- Minimally cognitive behavior.
- Evolving spiking neurons/STDP.

Overview

- Evolution in simulation and transference to reality:
  - Success in simulated evolution does not guarantee successful transfer to reality.
  - Simulations cannot model everything, nor can they accurately model anything.
- This applies even in today's world of super physics-based engines and the like.

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. Typically performance degrades after transfer.

Minimal simulations

- Accurate and encompassing simulations are costly, problem-specific, and slow (though see Physics Engines!).
- A methodology using clever, fast and cheap simulations can help.
- It breaks with traditional engineering view of uses for simulations.
- It works.

What's a simulation for?

- **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.

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 still very valuable.
- **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.

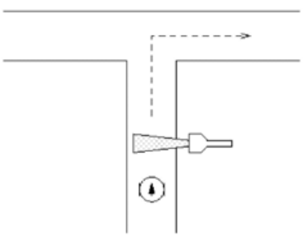
- What the simulation should do is **<u>not</u>** to accurately model the real world but to model *non-relevant* aspects of the world crudely and with large amounts of variation so that the only effective evolutionary strategy is to ignore those aspects.
- A base set of robot-environment relations that 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 **<u>large</u>** 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

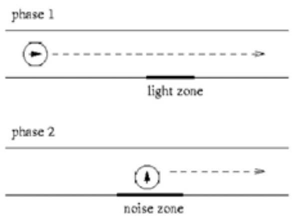
Example: T-Maze

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



Example: T-Maze

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



T-Maze

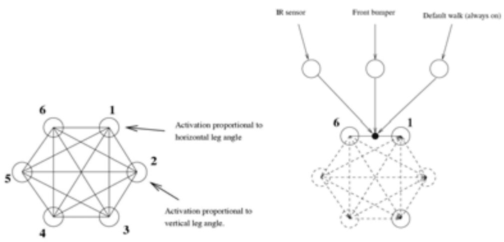
Fitness function:
Distance travelled along each corridor plus 100 bonus when turning in right direction.

Results:
1000 generations in 4 hours (in 1997!)

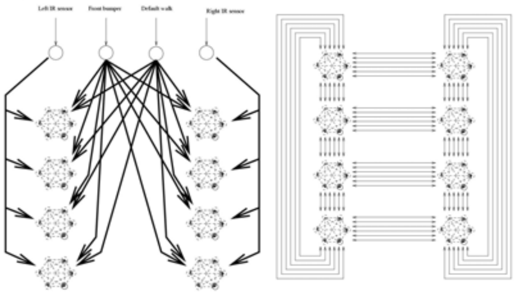
Example: Octopod

- 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

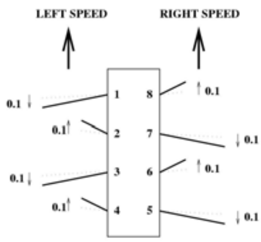


Octopod minimal simulation

- 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



Octopod video: “Maggie”

Minimal simulation challenges

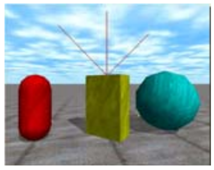
- Base set may not be so easily identifiable.
- Large amounts of variations make the search difficult. May even make it so difficult that it is not possible to evolve successful controllers unless variation is reduced.
- Evolution is deprived of its opportunistic ability. (Makes engineering sense, but maybe not scientific sense, it depends.)

Open Dynamics Engine (ODE)

- Cross-platform rigid body physics simulator.
- Compatible with openGL, written in C++/C.
- Stable, accurate simulation. Not the fastest, but becoming popular.

Open Dynamics Engine (ODE)

- Objects constructed out of component shapes (geoms).
- Simulation automatically handles underlying physics: collision detection, sensing (via rays), friction, gravity, joints, hinges, motors, springs, damping, etc.



Movies

Full side back

Biped balance

Forward Backward

‘Hot topics’ in ER

- Plastic controllers
  - Evolving plastic rules
  - GasNets
- Evolving body morphology
  - Golem project
  - Continuous self-modeling
- Minimally cognitive behaviour
- Spiking neurons & STDP

Evolving plastic rules

- Work by Floreano and Urzelai (EPFL, Lausanne).
- Instead of genetically specifying values for weights in neural networks, genotypes encode rules for plastic change. Weights are initialised at random and plasticity acts during the lifetime of the robot.
- Fewer parameters to evolve; more robustness; cross-platform transfer etc.

Encoding and rules

- 4 Rules. (You could think of other possibilities as well.)
- Rules can be encoded for each connection, or the same rule for all incoming synapses to each neuron
- Each 100 ms weights are updated according to:

$$w_{ij}^{t} = w_{ij}^{t-1} + \eta \Delta w_{ij} \qquad w_{ij} \in [0,1]$$

- Hebb: Strengthen synapse in proportion to correlated activity.

$$\Delta w = (1-w)xy$$

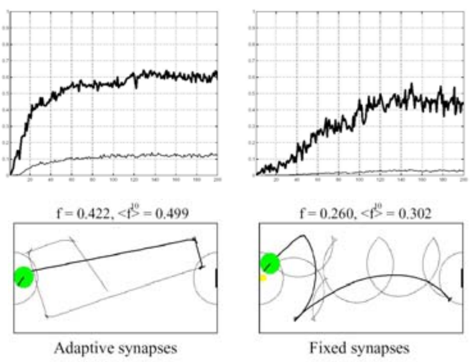
- Postsynaptic: As Hebb, but weakens synapses when postsynaptic node is active and presynaptic node is not. (Presynaptic: same but invert x-y).

$$\Delta w = (1-w)xy - w(1-x)y$$

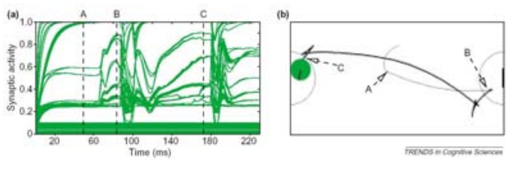
- Covariance: stronger when neurons have synchronous activity and weaker otherwise.

A sequential task

- A Khepera robot gains fitness by staying on grey area while light is on. But must switch on light first by stepping onto black area at the other end. IR, ambient light, and vision system. Discrete recurrent neural network as controller.



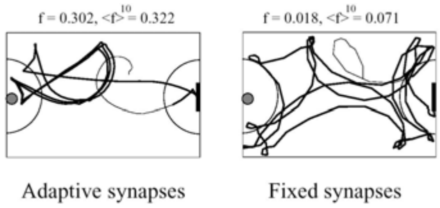
Some weights remain static, some change at a similar timescale as that of behaviour. Different "phases" can be identified



Cross-platform transfer

- Test Khepera controller in Koala robot (larger robot, six wheels, two motors, 16 IR sensors) without further evolution.

- Fairly good transfer for adaptive synapses. No transfer at all for fixed synapses



PAUSE !!

- Need a gap between the 2 lectures …

Evolutionary Robotics: So far

- ER for engineering vs ER for science
- Design of fitness functions
- CTRNNs/co-evolution
- Minimal simulations/the reality gap
- Detailed simulations/ODE, PhysX, etc..
- Evolving plastic rules.

Evolutionary Robotics: ctd…

- GasNets.
- Evolving body morphologies
- Self-modelling
- Minimally cognitive ER (more Beer)
- Spiking neurons & STDP.

GasNets

- A different kind of plastic neural controller. Inspiration from contemporary neuroscience. (Husbands, O'Shea, Smith, Philippides at Sussex)
- Real nervous systems can be viewed as involving several interacting dynamical processes each with different characteristics: electrical, short range chemical, long range chemical, etc.
- NB: some people are sceptical about validity of ***some*** of the claims. Some of the comparisons with CTRNNs are open to question.

- Much interest has been generated recently regarding NO as a neurotransmitter. Unlike classical transmitters that act mainly at the synaptic site, NO can diffuse over large volumes of space and remain active for long periods, affecting many cells and synapses. It has been implicated in many modulatory roles
- Modulatory effects could be used to change synaptic or cell properties, as well as modifying adaptive processes such as Hebbian learning.
- Will controllers evolve faster with such mechanisms? Will they be more robust?

A GasNet controller

- Neurons are spread on a 2D plane. They can be connected by inhibitory or excitatory links.
- Some nodes, under certain circumstances, can emit 'gases' that change the transfer function (in a concentration-dependent way) of other nodes.

$$O_i = \tanh(k_i(\sum w_{ji}O_j + I_i) + b_i)$$

- All parameters genetically set. *k* is affected by the local gas concentration (in ways that are genetically set).

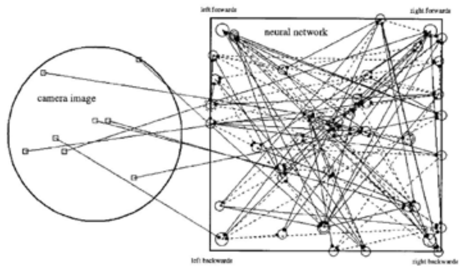
- Nodes can emit 2 'gases' under genetically set conditions:
  - node activation > *Te*
  - concentration of a gas in the vicinity > *Tc*
- Gas concentration:

$$C(d,t) = \begin{cases} C_0 e^{-2d/r} T(t) & d < r \\ 0 & else \end{cases}$$

- Radius of influence (*r*) genetically determined. *T(t)* linear build-up/decay function of time (slope genetically set). *Gas 1 increases k, Gas 2 decreases k.*

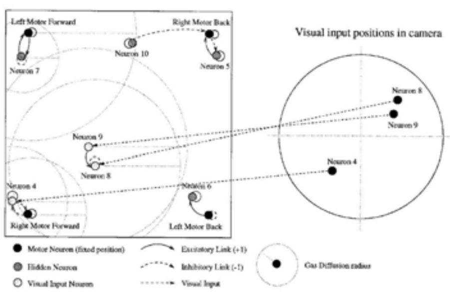
Results

- Without gas. About 6000 generations.



Results

- With gas. ~600 generations, 'simpler' structures.

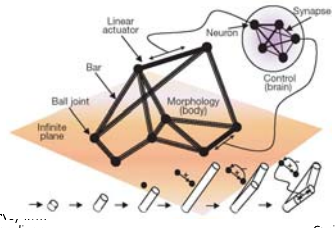


- More recently: looking genetically set 'receptors': each neuron may or may not respond to gas concentrations. Evolvability increases roughly tenfold again.
- Many open questions: why? what happens to fitness landscape? what about other mechanisms? How stable are these controllers? How robust?.
- What new understanding have we gained about neuromodulators in the brain?

Evolving morphologies:

- **Golem project:** Lipson and Pollack, Brandeis University, 2000.
- Evolutionary design of body morphology, actuators and controller + automatic fabrication of designs.

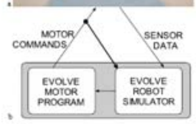
- Building blocks: bars, joints, linear actuators, neurons. Evolve in simulation (maximize distance travelled), build using rapid prototyping (extrusion of thermoplastic layer by layer), test in reality (with fairly good results). Mutations introduce new body elements or neurons and modify them. No sensors.



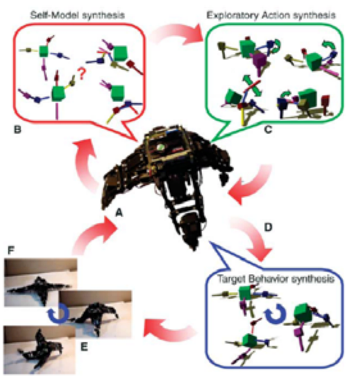
Examples

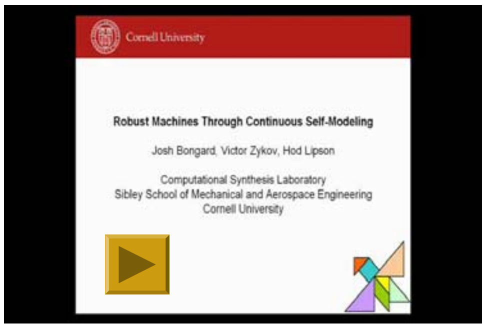
Automated synthesis of body schema

- Use co-evolution to develop a self-model of a robot body.
- These models can be used to drive behaviour.
- They can also be used to adapt to unforeseen morphological change (i.e., damage)



Bongard et al. 2006

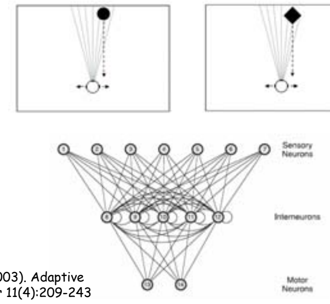




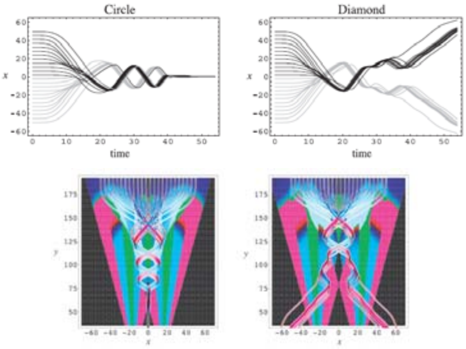
Minimally cognitive ER

- Randy Beer (again).
- Use ER to design minimal controllers for highly idealized cognitive behaviors.
- Analyze the dynamics of evolved controllers so as to extract general dynamical principles underlying cognitively salient agent-environment interactions.
- Treat the models as a sort-of Galilean 'frictionless plane'; an abstraction allowing insight into essential properties and mechanisms.

Example: Shape discrimination



Beer (2003). Adaptive Behavior 11(4):209-243



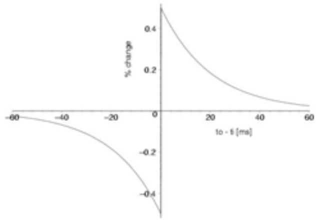
- Agent develops 'active scanning' strategy.
- Decision to 'catch' or 'avoid' is a temporally extended phenomenon, not a discrete event.
- Surprisingly rich dynamics even for such a simple (minimal) agent-environment system.
- Perception as a 'perturbation' to an agent, not as a representation of an external reality.
- How far do these results generalize?

Spiking neurons

- Interesting from the point of view of novel neural mechanisms. More plausible with increasing computing power. Very difficult to design functional controllers by hand. Evolutionary methods ideal for this.
- Floreano and Matthiusi (2001) evolved visually guided Khepera to navigate in an enclosure avoiding striped walls. New work aiming at flying robots.
- Di Paolo (2002/2003) Experiments evolving spike-timing synaptic plasticity (STDP).

STDP

- Synaptic plasticity depends on the precise relative timing between pre- and postsynaptic spikes.



Evolving STDP

- It is possible to evolve only the parameters of the STDP rules and other forms of plasticity.
- Controllers are highly robust. With low neural noise they rely heavily on spike timing.
- With higher levels of noise, synchronous firing happens but is not essential. STDP works (paradoxically) in the absence of precise spike timing.
- Comparisons with plastic CTRNNs show STDP to be more efficient in achieving stable controllers

Summary

- ER offers an (i) an automated engineering design methodology, and (ii) a scientific means of developing mechanistic models for dynamic, interactive behaviours.
- Engineering: Problems of scaling, introducing constraints, establishing operational ranges, finding and updating solutions. Etc.
- Science: Problems of abstraction from empirical data, of unconstrained modelling, of creating opaque models, of 'wall following' disease. Etc.
- Even so, ER models *can* create useful new designs and *can* challenge mechanistic assumptions.