Background:

Autonomous Agents:

- Biological robots learn by interacting with the outside world.
- Simple biological agents – the cockroach and its walking behavior.

Genetic Algorithms:

- Controllers of robots – instances of the free parameters are used to evaluate the policy.
- The free parameters of the agent are treated as analogous to genes.
- Similar to the selection of generations:
    - Create the population of biological agents
    - Evaluate each agent's "fitness"
    - Chose the "fittest" agents to generate the next generations
- New generations are populated via copying, mutations and crossover in order to produce sufficiently fit agents.
- GAucsd, a software package, is used to evolve the locomotion controller for the cockroach robotic legs

Dynamical Neural Networks:

- Primitive way to simulate biological neuron.
- The networks of neurons are represented by a system of equations that include a weight matrix where the weight is attenuation on each interconnect.
- CTRNN – "continuous time recurrent neural networks" is used in this paper
    - accurate approximator of smooth
    - continuous dynamical functions

Goal:

- Creating a successful walking hexapod robot.

Problems:

- Can an agent be successfully evolved in simulation and then applied to a physical agent embedded in the real world?
    - Create a robot completely in and simulation, then put the network into a physical agent, what will happen? Good things and bad things?
- What impact does limiting sensory feedback have on evolving an agent which is both robust and efficient?
    - What happens if the robot is damaged or losing some sensors?
    - If the robot loses a leg, can it deal with that? Can it still walk?

The Agent:

- To loosely mimic the walking behavior of the common cockroach.
- A neural network is used to control the legs.
- Each leg can be up or down, but not in between.
- Standing
    - Agent's center of mass is inside the polygon of support formed by the positions of the supporting feet.
    - At least three feet down and the center of mass is inside the triangle.
- If the agent is standing, it can make forward progress; if falls, set to zero.
- The Neural Network
    - Each leg is controlled by a five node fully interconnected network; three of these are motor neurons that govern the movement of leg.
    - Each neuron receives a weighted sensory input from the legs angle sensor.
    - Each five node network is connected to the corresponding controllers in adjacent legs.

Potential Problems:

- Body Instantiation Problem
  - The problems when replacing a simulated robot with a real robot.
  - Is the simulation good enough?
- Controller Instantiation Problem
  - The difficulty to build a physical controller.
  - The controller is implemented on a PC, can it still work with weight 60 lbs?

The Robot

- Each leg can do rotation and extension and swing forward and backward.
- The CTRNN is run on a PC and connected to the robot.

Neural Network Evolution

- Took place under three different conditions: sensory feedback always, never, and available 50% of the time.
- Averaging the performance with and without sensors resulted in mixed controllers.
- Walking better with sensors; with sensors intact, mixed controllers exhibited a higher stepping frequency.

Controller and Body Interface

- Legs stretch as forward translation occurs; accounted by adjusting the radial length of the leg using the kinematic transform.
- Interface of controller output to torques generated by the motors – a constant of proportionality.

Trial Runs of Simulation – Sensor Loss

- Assumption: sensors fail partially over the range of angles.
- Result: the robot often fell down; legs with intact sensors moved faster than legs without sensors.

Trial Runs of Physical Robot

- With sensory information available, the robot walked with three feet up and 3 feet down a tripod gait.
- Without sensors, the robot was never observed to fall down.
- Similar results as the simulation.
- The controller was extraordinarily tolerant of sensor noise when installed in the robot.

Controller Instantiation

- Physically embedded Hopfield Network:
  - Adder
  - Integrator
  - Mapping circuit
- VLSI Implementation
  - Work and resistant to imperfections in the chip.
  - A fair amount of numerical evaluation can be reduced to simple logic and analog circuits.

Conclusions

- It is possible to evolve a dynamical neural network in simulation to control a physical hexapod robot.
- The robot is resistant to significant amounts of sensory damage without completely destroying its ability to walk.
- It is possible to implement the controller in circuitry and VLSI as opposed to the tethered arrangement.

Future Work

- Continue to create robots that more closely mimic the biology of the cockroach
- Robot II video

    - The robot is attempted to crawl rocks.
    - The robot has to make its next step without losing the balance.
- Robot IV video
    - Just be developed six month ago.
    - Unfortunately, since the robot is very young, the video only shows how the robot stands up.

Discussions

Q: Is the environment of the simulation close to the real world?
A: There are two sides. Certain behaviors of the cockroach are not incorporated, such as its escape mechanism which take all the sensory information and determine which way to run. On the other hand, nobody has ever tested simulated insect against walking on slats, yet it has successfully walked on the slats. They have not published any situation that the control breaks down in the environment they expected to work.

Q: Why the cockroach was chosen?
A: large amount of knowledge about how its nerve system works for walking and its neural circuitry is comparatively simple.