Cognitive Robotics

Jeff Krichmar

Lecture 7

Outline

- Pfiefer & Bongard Chapter 12
  - How the Body Shapes the Way We Think.
- NXC Programming
  - Multiple tasks & Mutex.
  - Integer math
  - Arrays
- Vehicles

Steps Toward a Theory of Intelligence

- Diversity-Compliance
  - Diversity implies that the agent has a number of behaviors available to act appropriately to current situation.
  - Compliance implies that the agent must follow its ecological niche.

Steps Toward a Theory of Intelligence

- Frame of Reference
  - Define standpoint from which we view the agent's behavior.
  - Behavior is emergent through system-environment interaction.
  - Apparent complexity of behavior does not necessarily imply complexity of the underlying mechanism.

Steps Toward a Theory of Intelligence

- Physical embodiment
  - Embodied agents:
    - Subject to the laws of physics.
    - Generate sensory stimulation as they interact with the world.
    - Complex dynamical systems.
    - Perform morphological computation.

Steps Toward a Theory of Intelligence

- Agent Design Principles
  - Three constituents.
    - niche, task, agent.
  - Complete agent.
  - Parallel, loosely coupled processes.
  - Sensory-motor coordination.
  - Cheap design.
  - Redundancy or degeneracy.
  - Ecological balance.
  - Value.

Steps Toward a Theory of Intelligence

- Design Principles for Development
  - Integration of time scales.
  - Development as an incremental process.
  - Discovery.
  - Social interaction.
  - Motivated complexity.

Through Engineering to Science

- Engineering
  - Design.
  - Construction.
- Science
  - Analysis.
  - Understanding existing systems.
- Interdisciplinary
  - Synthetic methodology.
  - Understanding by building.

Intelligence as It Could Be

- Artificial Life's motto
  - "Life as it could be".
- Synthetic methodology
  - "Intelligence as it could be".

Information Self-Structuring through Sensory-Motor Coordination

- Through physical interaction with the real world, sensory stimulation is induced in different sensory channels.
  - Our brains are great at correlating multimodal information.
  - Sensors are positioned in morphologically "clever" locations.
  - Morphology assists in the sensation.
- Rich sensory information can be acquired with little effort.
- Complexity of information is reduced through sensory-motor coordination.

Morphological Computation

- Agents must off-load neural processing into their morphology and environment.
  - Running exploits elastic properties of muscles and tendons.
  - Layout of insect photoreceptors compensate for motion parallax.
  - Morphology of the hand:
    - Deformable fingertips.
    - Elasticity.
    - Distribution of touch sensors.

The Brain Does Not Control the Body

- The body has its own dynamics and the dynamics of the nervous system must match.
  - Lamprey CPGs.
  - Running on varied terrain.

Exploiting Intrinsic Dynamics

- Exploiting intrinsic dynamics can lead to the achievement of tasks for free with no control.
  - Passive walker robots.
  - Quadruped puppy.

Embodiment as a Prerequisite of Cognition

- Memory, categorization, language, and consciousness are all reflected in the body schema.
- The world is continuous, fuzzy, and fluid.

Seeing things differently

- Swiss robots.
- Passive walkers.
- Legged locomotion.
- Social interaction does not require rules.
- Ant navigation.
- Creative computers.

```
#define BUMPER_PORT S1
#define BUMPER      SENSOR_1
#define BAT_PORT    S4
#define BAT_THRESHOLD 30

mutex motorMutex;

task Move() {
  while (true) {
    Acquire(motorMutex);
    OnFwdSync(OUT_AC, 75, 0);
    Release(motorMutex);
    Wait(500);
  }
}

task WatchUltra() {
  while (true) {
    int dist = SensorUS(BAT_PORT);
    if (dist < BAT_THRESHOLD) {
      Acquire(motorMutex);
      PlayTone(440, 500);
      OnRevSync(OUT_AC, 40+Random(60), 0);
      Wait(500+Random(500));
      OnFwdSync(OUT_AC, 40+Random(60), 100);
      Wait(500+Random(1000));
      Release(motorMutex);
    }
  }
}

task WatchBumper () {
  while (true) {
    if (BUMPER) {
      Acquire(motorMutex);
      PlayTone(880, 500);
      OnFwd(OUT_A, 40+Random(60));
      OnFwd(OUT_C, 60+Random(40));
      Wait(500+Random(1000));
      Release(motorMutex);
    }
  }
}

task main() {
  SetSensorTouch(BUMPER_PORT);
  SetSensorLowspeed(BAT_PORT);
  Precedes(WatchBumper, WatchUltra, Move);
}
```

NXC Programming

- Integer math

```
int x = 60;
int y = 40;
int z = x/y;
```

- Arrays

```
int a[10];
for (i = 0; i < 10; i++) {
    a[i] = i*i;
}
a[2] = ?;
```