CORC 3303 Exploring Robotics

Lecture E

Control

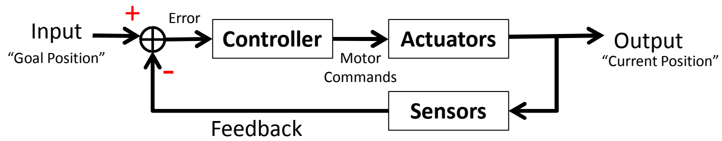
- **Topics:**
  1) Feedback Control
  2) Control Architectures
  3) Behavior-Based Control, Behavior Coordination

Feedback Control

- A means of getting a system to achieve and maintain a desired state (*set point,* or *goal state*), by continuously comparing it with the current state and minimizing the error
- Desired goals have two types
  – Achievement goal: no more work after final state
  – Maintenance goal: ongoing active work
- Goal state may be internal or external or the combination of the two
- Goal could be an unreachable one, as long as the robot keeps trying

Feedback Control, cont.

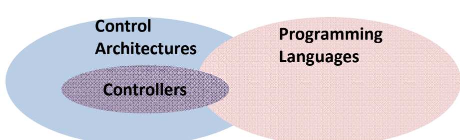
- A diagram of a typical feedback controller



- Types of Feedback Control
  - Proportional Control (P): gain is proportional to error, and it determines the amount of oscillating or damping
  - Proportional derivative control (PD)
  - Proportional integral derivative control (PID)
- Feedback control is a low level control governed by Control Theory and Mathematics. Higher-level goals require AI.

Control architecture

- What if there are multiple feedback controllers? How to decide what is needed, which part of the control system to use in a given situation? What priority to assign to it?
- A **control architecture** provides a set of principles for organizing a robot's control system
- Provides constraints, guides how programs are structured
- Refers to software control level, not hardware!
- Implemented in a programming language



Classes of robot control architectures

- Control architectures, no matter how different they may look, fall into one of the four categories.
  - *Deliberative:* Look-ahead; think, plan, then act
  - *Reactive:* Don't think, don't look ahead, just react!
  - *Hybrid:* Think but still act quickly, both work in parallel
  - *Behavior-based:* Distribute thinking over acting, think the way you act
- In most cases for simple robots, it's impossible to tell what control architecture a robot is using simply by observing its behavior.
- Architectures become more important for complex robots, and their choices are based on:
  - Time scale: how fast do things happen?
  - Modularity: what are the component of the control system and how they interact?
  - Representation: What does the robot know and keep in its brain?

Representation

- Representation is the form in which information is stored or encoded in the robot
  - What is stored, how it is stored, how it is used?
- Representation of the world is called a world model. E.g. a map in navigation, many ways of modeling exist:
  - Odometric path: go fwd for 3cm, turn left 90 deg…
  - Landmark-based path: turn left at the first junction…
  - Topological map (landmark-based map, the order of the junctions is not required)
  - Metric map: drawing the map with exact distances.
- The world model needs to be kept accurate and updated. Some are long-lasting, others are transient
- Representation comes with a cost (sensing, computation, memory), and the level of its use depends on the architecture chosen.

Deliberative control

- Classical control architecture (first to be tried)
- First used in AI to reason about actions in non-physical domains (like chess)
- Natural to use this in robotics at first
- Example: Shakey (1960's, SRI)
  - State-of-the-art machine vision used to process visual information
  - Used classical planner (STRIPS)
- Planner-based architecture
  1. Sensor (S)
  2. planning (P)
  3. Acting (A)
- Requirements
  - Lots of time to think
  - Lots of memory
  - (but the environment changes while the controller thinks)

Reactive control

- Operates on a short time scale
- Does not look ahead
- Based on a tight loop connecting the robot's sensors with its effectors
- Purely reactive controllers do not use any internal representation; they merely react to the current sensory information
- Collection of rules that map situations to actions
  - Simplest form: divide the perceptual world into a set of mutually exclusive situations, recognize which situation we are in and react to it
  - (but this is hard to do!)
- Example: subsumption architecture (Brooks, 1986)
  - Hierarchical, layered model

Hybrid control

- Use the best of both worlds (deliberative and reactive)
- Combine open-loop and closed-loop execution
- Combined different time scales and representations
- Typically consists of three layers:
  1. Reactive layer
  2. Planner (deliberative layer)
  3. Integration layer to combined them
  – (but this is hard to do!)

Behavior-based Control (BBC)

- Involving the use of “behaviors” as modules for control. Incorporating the best of reactive systems, but does not involve a hybrid solution
- What is a behavior?
  – Achieve, maintain particular goals (e.g. homing, wall-following).
  – Time-extended, not instantaneous
  – Take inputs from sensors, and other behaviors. Send outputs to effectors and other behaviors.
  – More complex than actions (e.g. find-object vs. stop-turn-left)

Behavior-based Control, cont.

- Principles for BBC design:
  - Behaviors executed in parallel/concurrently for immediacy
  - Networks of behaviors used to store states
  - Behaviors operate on compatible time-scales
- The internal, unobservable behavior and the external, observable behavior are not necessarily the same.
- The observable behaviors could be the result of the *interaction dynamics* between the internal control and the environment.
- Emergent behaviors: simple rules or behaviors interact to produce more complex outputs
  - Wall following
  - Flocking

Behavior Coordination

- What action/behavior should be executed next? Behavior coordination = Action selection
- There are two ways: *arbitration* or *fusion*
- Behavior Arbitration: selection one action/behavior from multiple possible candidates – competitive (one choice wins).
  - Fixed priority hierarchy (pre-assigned priorities)
  - Dynamic hierarchy (behavior priorities change at run-time)
- Behavior Fusion: combining multiple possible candidate action/behaviors – cooperative.

An Example of BBC

the Jungle Bug

- Each Bug has a repertoire of different behaviors: Wander, Sleep, Avoid-Left, etc.
- Each behavior is triggered by either the stimuli like a bumper being pushed, or changes in the internal state (e.g one behavior just finished)
- The selection mechanism is based on the concept of motivations.
- At any given instance, each behavior is associated with a motivation value based on sensor inputs. The behavior with the highest value wins and becomes the active behavior.
- E.g. motivation value for Wander is high during the day and low at night, and the value for Avoid-Left is higher than Wander when the left bumper is activated.

An Example of BBC, cont.

From: http://legolab.daimi.au.dk/Projects/JungleCube.dir/Chapter.html