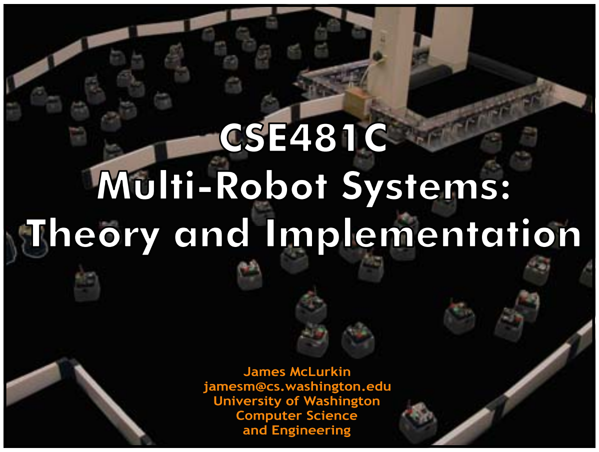


CSE481C Course Information
Multi Robot Systems: Theory and Implementation

People
- James McLurkin, jamesm@cs.washington.edu
- Ryan McElroy, arcanius@cs.washington.edu

Textbook
- No textbook, selected papers over the quarter

Grading
- Four labs with simple write-ups: 10%, 10%, 15%, 15%
- Final project proposal: 20%
- Final Project: 30%

Lab
- We will alternate lecture and lab during the class time. You must attend both classes each week.
- Labs are in CSE 003D
- There is a sign-up for staffed lab hours. Rank your choices.

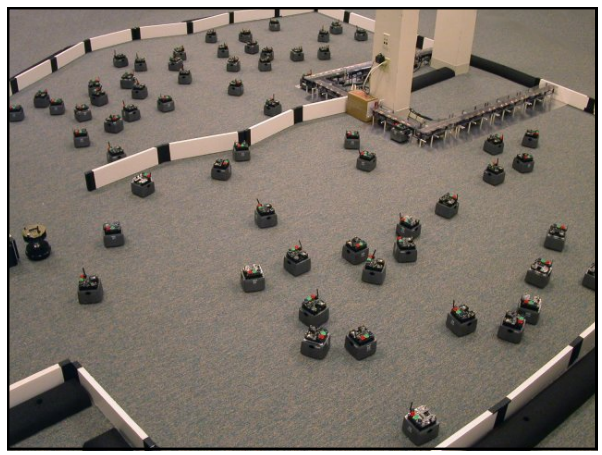
Course Goals:

Lectures:

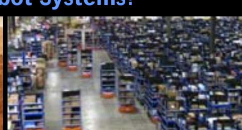
Introduce the key concepts of multi-robot systems:
- Robotics
- Communications
- Distributed Algorithms
- Configuration Control

Labs:
- Implement one topic from each concept
- System engineering of multi-threaded software
- Measure performance, analyze data

Final Project:
- Implement an algorithm of your design
- Write a clear, thoughtful proposal



Why Multi-Robot Systems?



Multi-Robot systems are best when you have a
- Large area to search, explore, or map
- Need to sense, compute, or manipulate in multiple places

Good applications
- Searching, exploration, surveillance

Future applications
- Distribution, agriculture, rescue

Course Overview

Course Outline

1. Robotics 101 (Lab 1)
   - Behavior-Based systems
2. Networked communications (Lab 2)
   - Ad-hoc networks
   - Broadcast and convergecast communications
3. Distributed Algorithms (Lab 3)
   - Consensus
   - Computational geometry
4. Configuration Control (Lab 4)
   - Dispersion, exploration, mapping
5. Other Issues in Multi-Robot Systems
   - Complexity Algorithm Execution
   - Biological systems
6. Final Project

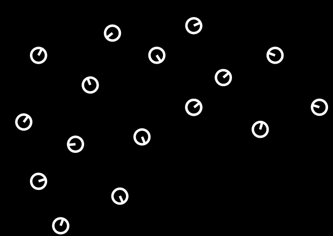
1. Robotics 101

A robot must Sense, Process, and Actuate
- Each of these three steps is difficult

We will only use simple robot controllers in this class
- Take the robotics capstone for more

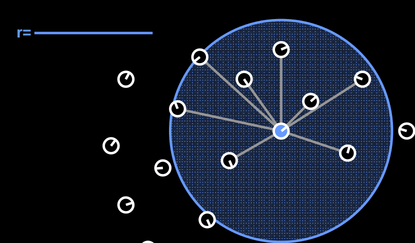
2. Network Communications

If we assume limited communications range, robots must form ad-hoc networks



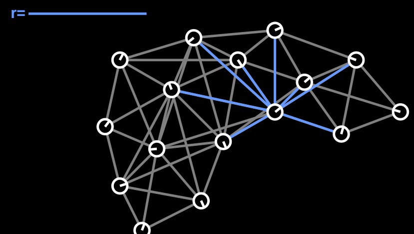
2. Network Communications

If we assume limited communications range, robots must form ad-hoc networks



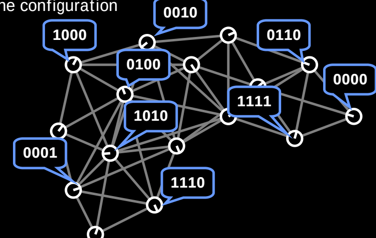
2. Network Communications

If we assume limited communications range, robots must form ad-hoc networks



3. Distributed Algorithms

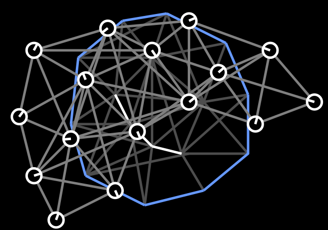
Combine local processing to produce group results
- This applies to algorithmic processing, and the physical output of the robots
- Use *computational geometry* to reason about the arrangement of the configuration



4. Configuration Control

A *configuration* is a group of robots and their internal states

Configuration control algorithms modify the configuration to produce some desired configuration



Example Applications and Multi-Robot System Comparisons

Our Canonical Multi-Robot Application has…

…high robot mobility

…short-range local communication &
multi-hop global communication

…a noisy sensor on each robot to measure the positions of other robots

…algorithms that are robust to population changes and robot failures

Multi-Robot Applications are Hard Because of…

…high robot mobility

…short-range local communication &
multi-hop global communication

…a noisy sensor on each robot to measure the positions of other robots

…algorithms that are robust to population changes and robot failures

Exploration: Small Group of Complex Robots

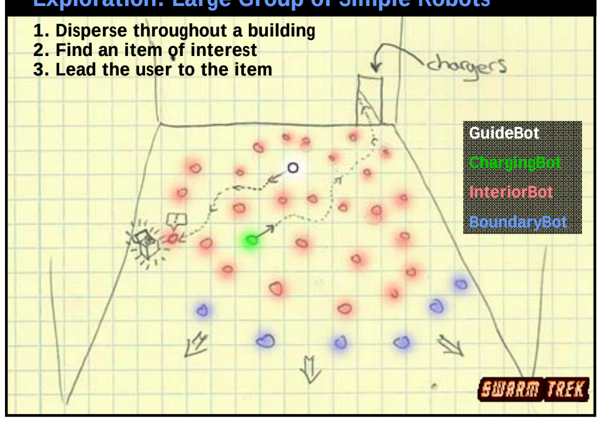
Coordinated exploration with three robots
from unknown start locations

The robots are fully autonomous.
All computation is performed on–board.

Shown is the perspective of one robot

D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schulz, and B. Stewart. "Distributed Multi-Robot Exploration and Mapping", 2006

Exploration: Large Group of Simple Robots



J. McLurkin, J. Smith. "Distributed Algorithms for Dispersion in Indoor Environments using a Swarm of Autonomous Mobile Robots", DARS 2004

The Perils of Small Robots in Big Buildings

Comparisons

Single Robots
- Robotics Capstone

Small Populations (< 10ish)
- Some single-robot techniques can be adapted
- Often with complex robots, good sensors, lots of processing

Medium Populations (< 1000ish)
- Centralized data structures and control become less practical
- Switch to distributed algorithms
- Often with simple robots, limited sensing, limited processing

Large Populations
- Must look to nature for inspiration
- Insect colonies can reach sizes of 20,000,000!

Behavior-Based Control

Behavior-Based Control

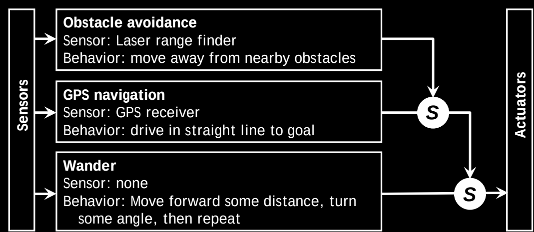
A behavior is a small program that reads sensors and controls an actuator
- They are *reactive*, and act like reflexes.
- Behaviors do not plan and have no state (or very limited state)

Example Behaviors for an outdoor driving program:
- GPS navigation
  - Sensor: GPS receiver
  - Behavior: drive in straight line to goal
- Obstacle avoidance
  - Sensor: Laser range finder
  - Behavior: move away from nearby obstacles
- Wander
  - Sensor: none
  - Behavior: Move forward some distance, then turn some angle, then repeat

Combining Behaviors

It is difficult to combine all of your tasks into a single behavior
- Hard to program
- Can produce unintended interactions

We need an abstraction to manage many behaviors:
- Combine behaviors, Sequence behaviors, or *Subsume* behaviors:



Genghis

The behavior-based poster child

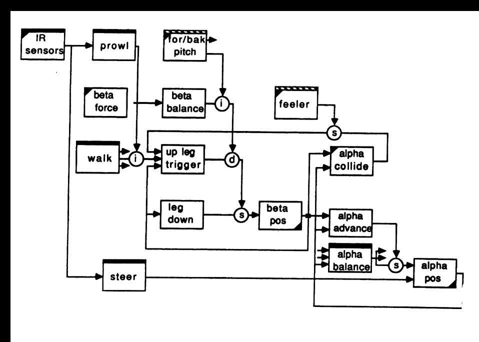
Hardware
- 6 legs, 2 motors per leg
  - α-motor for forward/back, β-motor for up/down
- 2 bump sensors (feelers)
- 2 ground detection sensors (switches)

Very limited computation

R. Brooks. "A Robot that Walks; Emergent Behavior from a Carefully Evolved Network", ICRA 1989

Gengis in Action

Subsumption Architecture



R. Brooks. "A robust layered control system for a mobile robot", ICRA 1986

Gak! Why so Complicated?

Why not just program it directly?
- That's what we would do now.
- Back then, there was a political point to make: *Intelligent-looking behavior can arise from simple processing that is tightly coupled to sensors (aka: the environment) and actuators.*
- Behaviors can be compiled from a high-level specification into machine code
- This whole process is called Subsumption Architecture (there was even a complier!)

For this course:
- The basic concepts of behavior design and arbitration continues in most modern robotic system
- But your behaviors will be at a higher level of abstraction

Lab 1: Behavior-Based Control

Lab 1: Behavior-Based Control

Goal

- Build a layered control system to guide a robot towards light and away from obstacles
- Measure the robot's estimate of its *pose* from odometry compared to ground truth

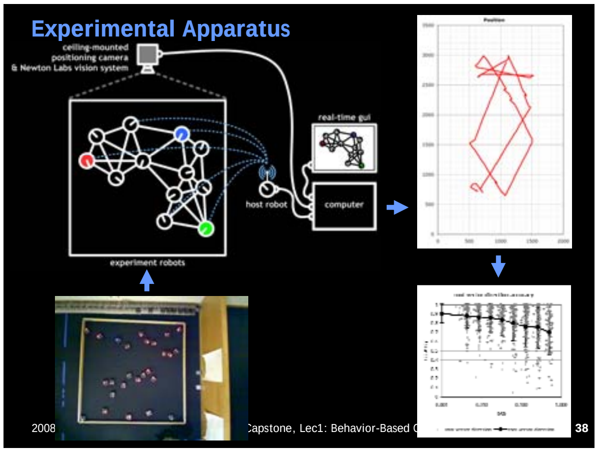
Experimental Apparatus: SwarmBot

Photo courtesy iRobot

Experimental Apparatus: Data Collection

Photo courtesy iRobot

Experimental Apparatus



2. Robot Demo!