a) Wrapup of Learning

b) Multi-Robot Systems, Part I

October 29, 2002

Class Meeting 19

“The mob has many heads but no brains”. -- English Proverb.

Announcements

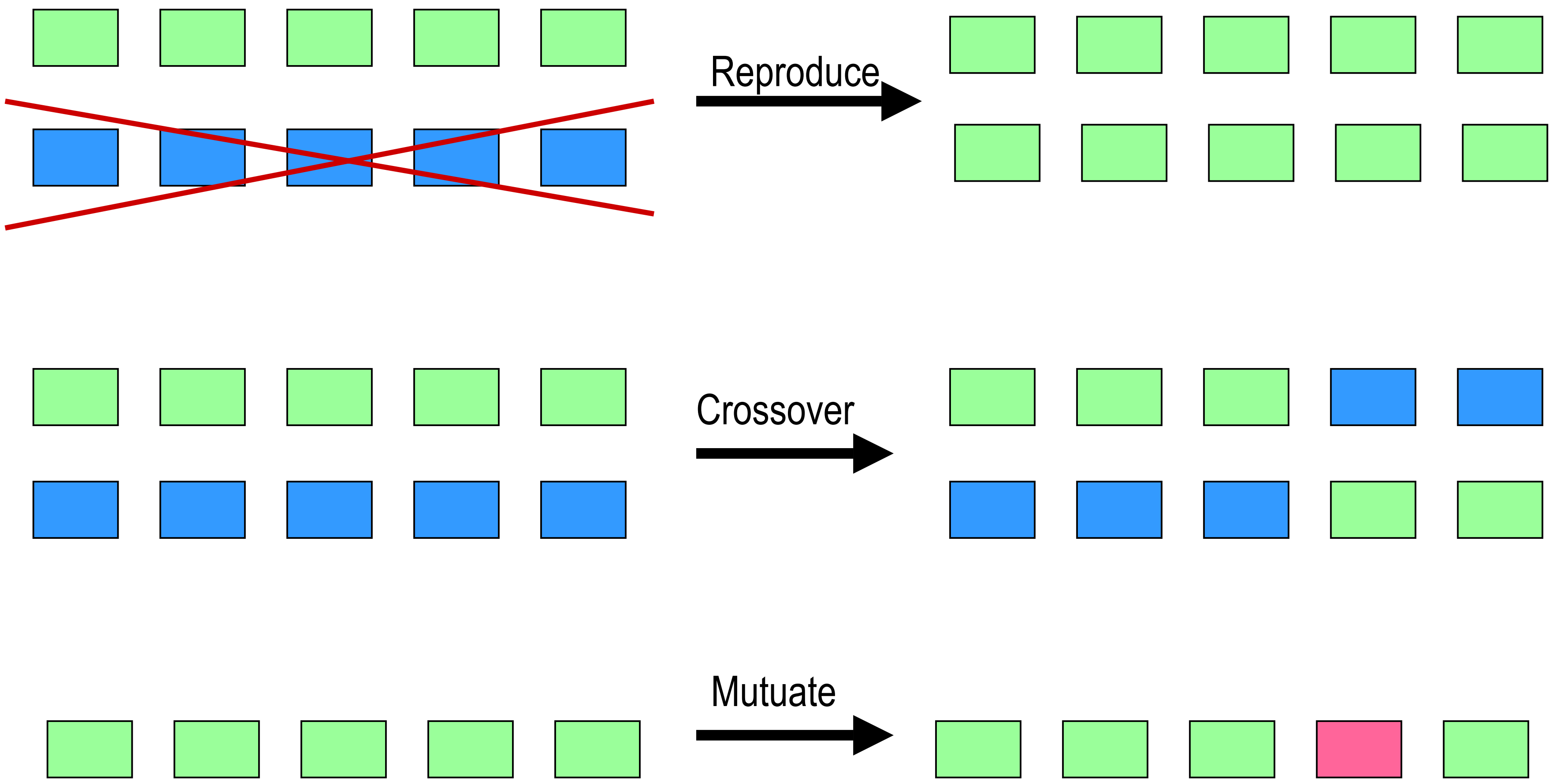
- Remember:
  - Assignment #5 Handed out today, DUE: November 7
  - Final Project Proposals DUE: This Thursday, Oct. 31
    - Final proposals (about 1 page) should include:
      - Statement of problem you are addressing
      - Your intended approach
      - Your planned experiments to evaluate your approach: VERY IMPORTANT
        - » This should include definition of a metric (or metrics)
        - » Data collection relevant to your metric(s), based on multiple runs that vary some aspects of problem (for example, varying distributions of beacons, varying numbers of robots, varying distributions/shapes/arrangements of obstacles, varying algorithms, etc.)
        - » Note that your final project report should include some sort of graph/plot/table, etc., that reports your findings based upon this data collection and analysis

Objectives

- Wrap up discussion on Genetic Algorithms for robot learning

- Multi-Robot Systems, Part I
  - Overview
  - Multi-Robot Communication

Genetic Operators



GAs for Robot Learning

- Prior to operator's application, each individual's fitness is computed using fitness function
- For robot learning, this may involve running a robot through a series of experiments, using the encoding of the behavioral controller represented by the particular individual bit string encoding being evaluated
- Fitness function returns a value capturing the robot's overall performance for the set of conditions being tested

GAs for Robot Learning (con't.)

- Reproduction operator:
  - Fittest individuals are copied exactly and replace less-fit individuals
  - This is done probabilistically, usually using roulette-wheel selection
  - Net effect: increase in ratio of highly-fit individuals relative to # of poor performers
- Crossover operator:
  - Exchange of information through transfer of info between two individuals
  - Process creates new individuals that may or may not perform better than parents
  - Which individuals to cross over and which bit string pairs to exchange is often done randomly
  - Net effect: increase in overall population

GAs for Robot Learning (con't.)

- Mutation operator:
  - Simple probabilistic flipping of bit values in encoding
  - Affects individual only
  - Probability of mutation is generally low
  - Net effect: ability to escape local minima
- Overall effect: changing population over time
- Final quality and length of time to obtain solution depend on nature of problem and parameters used

Example of GAs for Learning Behavioral Control

- GAs, although powerful, require some restrictions on implementation compared to previous learning approaches we've discussed

- Much of learning must be done off-line, since:
  - Significant population of robots needed for fitness testing
  - Robots must be tested over many, many generations

- Simulations, fortunately, can be run much faster than real-world testing

- If simulation has reasonable fidelity to real robot and environment, control parameters from fittest simulated individual can be transferred to actual robot for use

Example Robot GA Code

```
begin
  Obstacles.Create;
  Population.Build;
  for 1 to NUMBER_GENERATIONS do
        begin
               for 1 to RUNS_PER_GENERATION do
                      begin
                             for 1 to MAX_NUMBER_STEPS do
                                    begin
                                    ROBOTS.Move
                                    end
                             Obstacles.Recreate;
                      end
               Robots.Reproduce;
               Robots.Crossover;
               Robots.Mutate;
        end
end
```

Example Robot GA

- GA-Robot (Ram, 1994): schema-based behavioral controller evolved using GAs

- Encoding: represents the individual gains of the component behaviors:
  - goal attraction
  - obstacle avoidance
  - noise

  as well as additional parameters internal to certain behaviors:
  - obstacle sphere of influence
  - noise persistence

- Instead of using a more slowly converging bit string, an encoding using floating-point values for the gains and parameters is used

Example Robot GA (con't.)

- Fitness for an individual is defined as a function of weighted penalties:

```
raw_fitness = collision_weight  x  number_of_collisions
                 +  time_weight  x  number_of_steps
                 +  distance_weight  x  distance_traveled
```

- By altering penalty weights for each component of the fitness functions, 3 different classes of robots are evolved, each specialized for a particular ecological niche:
  - Safe: optimized to avoid hitting obstacles while still attaining the goal
  - Fast: optimized to take the least amount of time to attain the goal
  - Direct: optimized to take the shortest path (which may be slower because of reduced speeds in cluttered areas)

Other Capabilities Learned with GAs

- Learning to approach both stationary and moving light sources
- Learning of primitive behaviors, such as approaching, chasing, and escaping
- Learning location of energy sources and not getting stuck in obstacle traps while seeking them out

Hybrid Genetic/Neural Learning and Control

- Several researchers have combined neural nets and GAs for robot learning

- Typically, approach is to use GAs to learn synaptic weights for a neural controller

- Example by Floreano et al:
  - Implementation of Braitenberg-style neural controller
  - Use of robot (Khepera) with 3 ambient light sensors and 8 IR proximity sensors
  - Fitness functions defined for behaviors, including:
    - Navigation and obstacle avoidance: Fitness maximizes motion and distance from obstacles
    - Homing: Fitness ensures that power is kept at adequate levels by adding a light-seeking behavior to guide it to its black recharging area when power becomes low
    - Grasping of balls using an added gripper: Fitness maximizes the number of objects gripped in an obstacle-free environment
  - Most successful individual: learned to back up until it encountered something, then turned around an attempted to grip it

Summary of Learning/Adaptation

- Neural networks, a form of reinforcement learning, use specialized, multi-node architectures.
- Learning in neural nets occurs through the adjustment of synaptic weights by an error minimization procedure, such as:
  - Hebb's rule
  - Back Propagation
- Classical conditioning in which a conditioned stimulus is eventually associated with an unconditioned response can be manifested in robotic systems
- Genetic algorithms operate over sets of individuals over multiple generations using operators such as selection, crossover, and mutation
- Effective fitness functions must be defined for a particular task and environment for successive evolutionary learning. By suitable selection, particular ecological niches can be defined for various behavioral classes of robots.

Multi-Robot Systems: Research Growing Rapidly

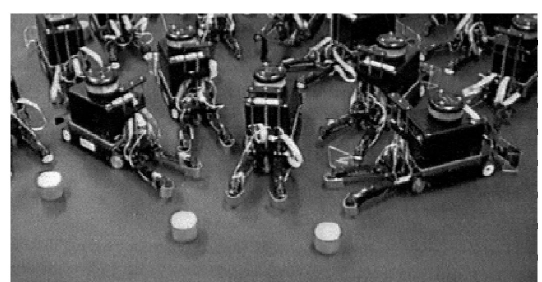
- Previous summaries/overviews of the field:
  - L. E. Parker, "Current State of the Art in Distributed Autonomous Mobile Robotics", *Proc. of Distributed Autonomous Robotic Systems 2000.*
  - Cao, *et al., "*Cooperative mobile robotics: Antecedents and directions"*, Autonomous Robots*, 4 (1), 1997: 7-28.
  - G. Dudek, *et al.*, "A taxonomy for swarm robots"*, Proc. of IEEE International Conference on Intelligent Robots and Systems (IROS),* 1993: 441-447.









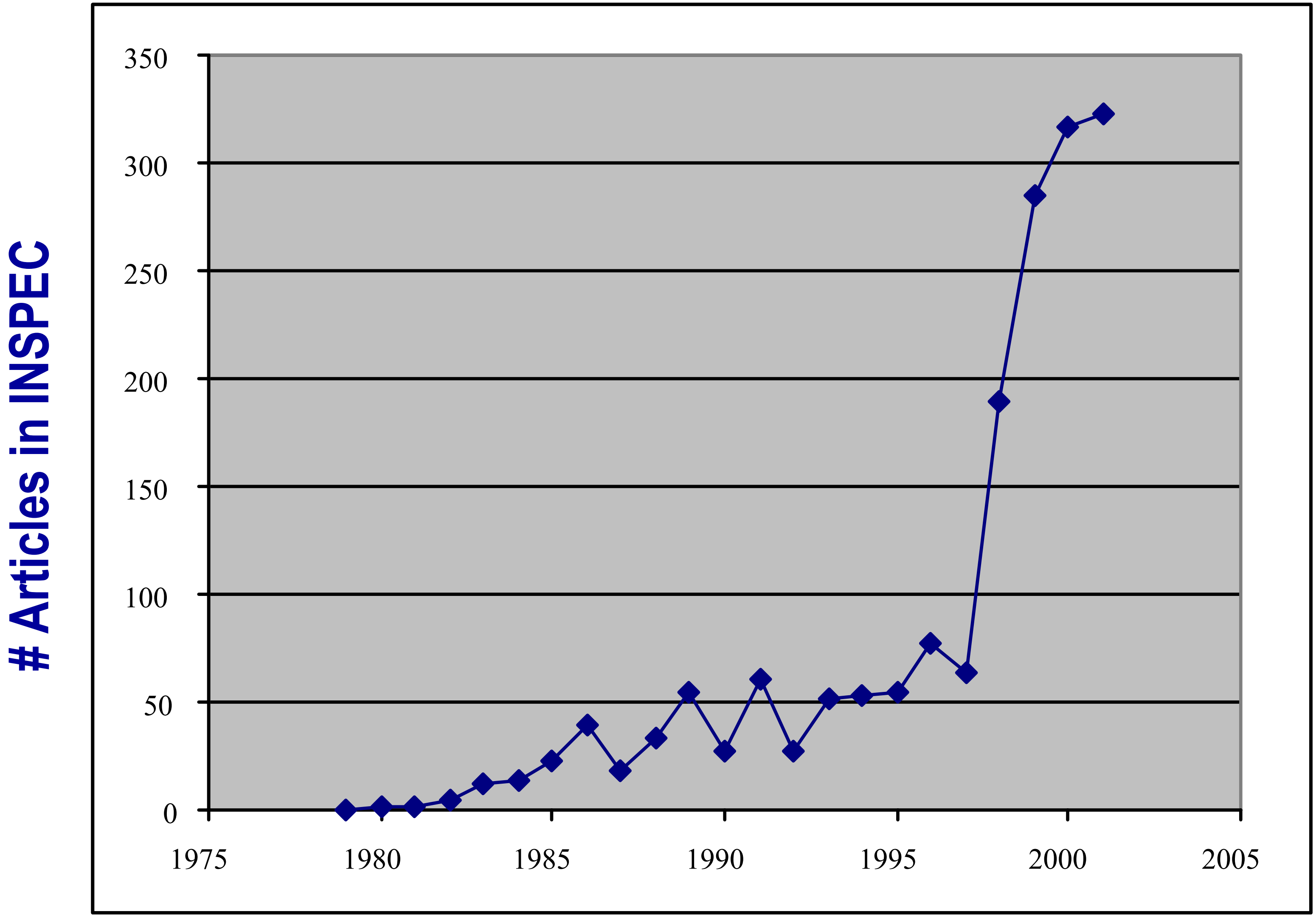



How rapidly is this research growing?

- To investigate, conducted an INSPEC* Search:
  - Yearly query, 1979 - 2001
  - Searched for articles including at least one of the following terms:
    - Multi-robot
    - Multirobot
    - Cooperative robot
    - Collaborative robot
    - Distributed robot

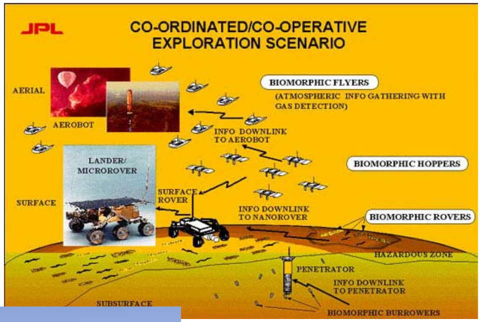
*** Citation index for physics, electronics, and computing**

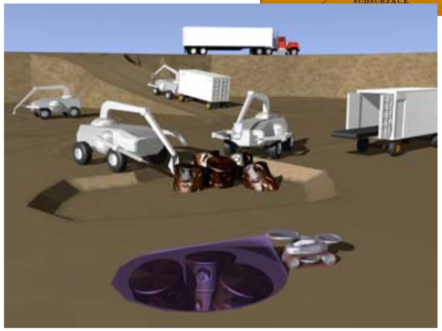
Results of INSPEC Search Show Enormous Growth in Multi-Robot Systems Research



What are reasons for enormous growth?

- Many potential application domains
- RoboCup influence
- Increased computational capabilities
- Advances in individual autonomous robotics
- Advances in understanding of complex systems
- Etc...



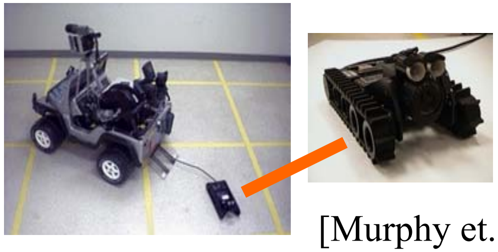


One Categorization of Multi-Robot Systems

- Cooperative robotics field is often divided according to a number of criteria:
  - Collective (swarm) cooperation
    - Many robots; sub-symbolic communication (possibly implicit)
    - Typically uses insect society cooperation model

[Mataric et. al. USC/MIT]

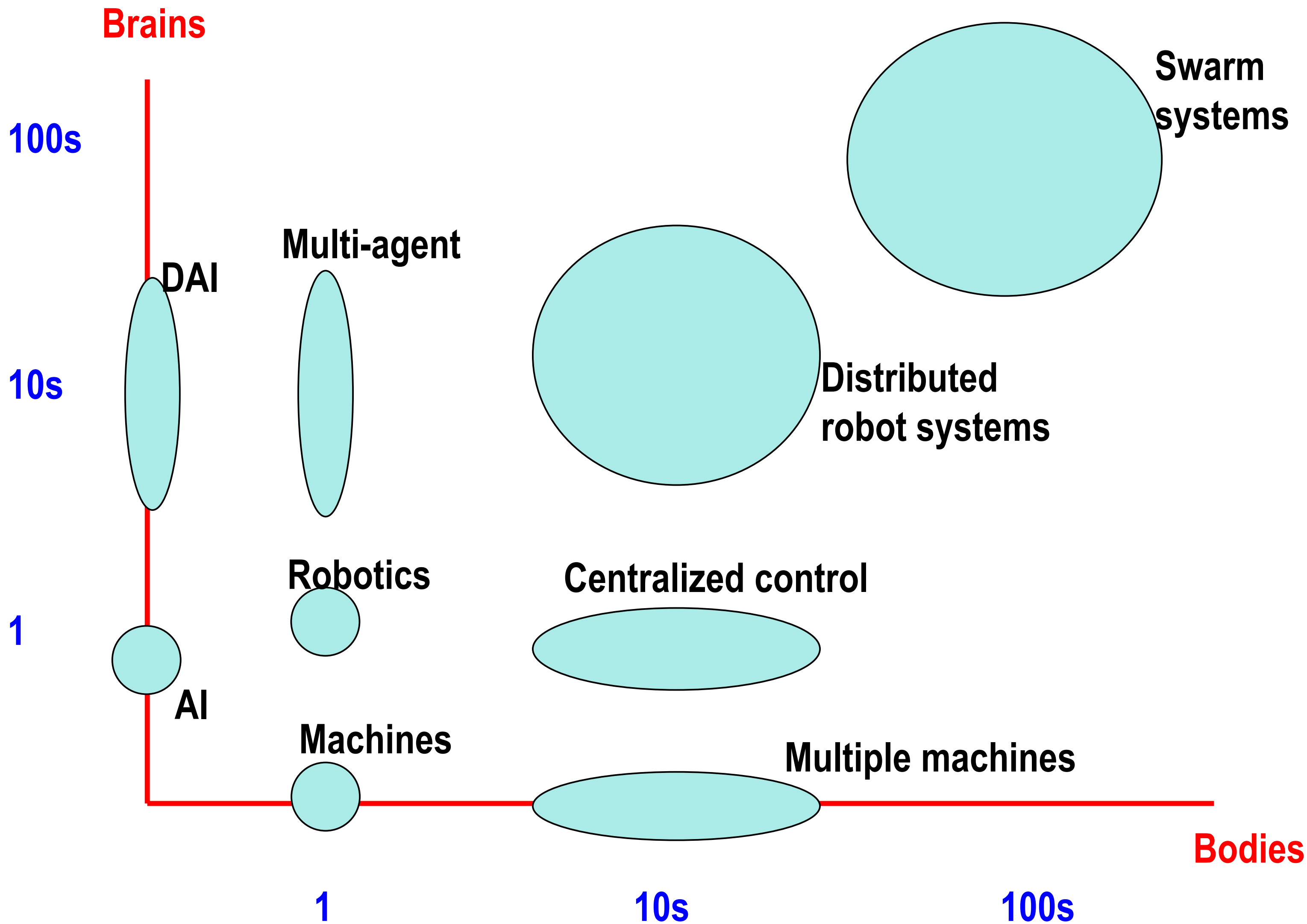
  - Homogeneous vs. Heterogeneous systems
    - Sensors, actuators and behavior
    - Affects communication possibilities



[Murphy et. al. USF]

  - Centralized vs. Distributed
    - Centralized systems typically use classical-AI planning, rather than being behavior-based (new AI)

Multi-Robot Systems -- Brains + Bodies



Are N Robots Better than One?

- Positive aspects of teaming:
  - Improved system performance
  - Task enablement
  - Distributed sensing
  - Distributed action at a distance
  - Fault tolerance
- Negative aspects of teaming:
  - Interference
  - Communication cost and robustness
  - Uncertainty concerning other robots' intentions
  - Overall system cost