Introduction to Robotics

Path Planning for Multiple Robots

Erion Plaku

Department of Electrical Engineering and Computer Science
Catholic University of America

Objective: enable robots to navigate collaboratively to achieve spatial positioning goals

Issues studied:

- Multi-robot path planning
- Traffic control
- Formation generation
- Formation keeping
- Target tracking
- Target search
- Multi-robot docking

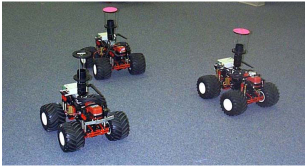


Figure: Formation (Kumar, UPenn)

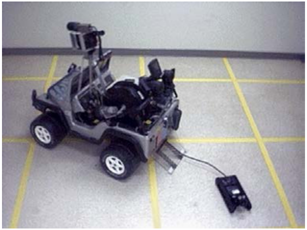


Figure: Docking (Murphy, USF)

- Given: m robots in k-dimensional workspace, each with starting and goal poses
- Determine path each robot should take to reach its goal, while avoiding collisions with other robots and obstacles
- Typical optimization criteria:
  - Minimized total path lengths
  - Minimized time to reach goals
  - Minimized energy to reach goals
- Unfortunately, problem is PSPACE-hard
  - Instead, opt for locally optimal portions of path planning problem

- Force multiplication

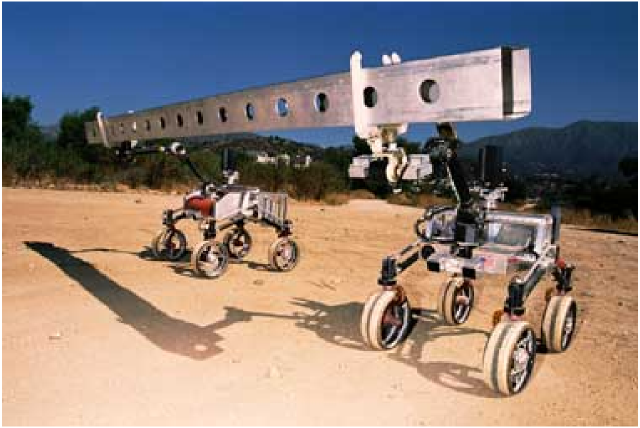


Figure: NASA Planetary Outpost - JPL

- Simultaneous Presence

Figure: Security Robot - iRobot

- Redundancy, fault tolerance

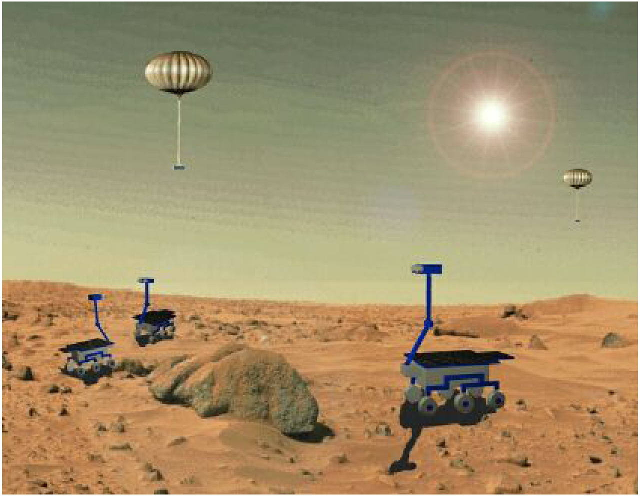


Figure: Mars explorations - Matsuoka 2002

Motivation

- Case for multiple robots
  - R robots to increase performance by a factor $\geq$ R
  - Tasks that cannot be accomplished by one robot
- Applications
  - Competitions
  - Underwater sensing
  - Unmanned aerial vehicles

- Competitions

Figure: RoboCup (Padua, Italy, 2003)

- Underwater sensing

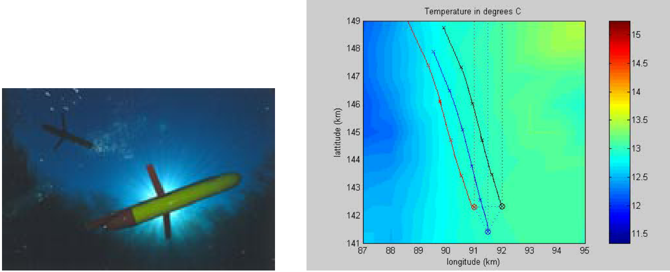


Figure: Gliders from Autonomous Ocean Sampling Network (Naomi Leonard, 2003)

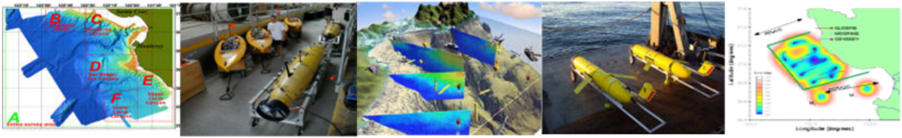
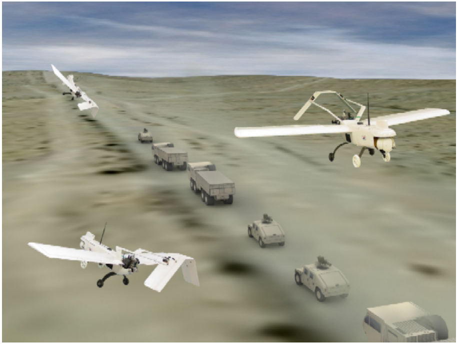


Figure: Adaptive sampling and prediction (Naomi Leonard)

- Unnmanned aerial vehicles



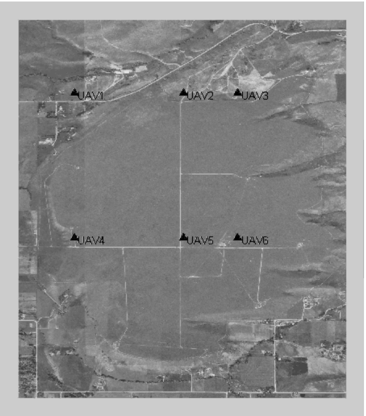


Figure: Eric Frew, MLB

Taxonomies

Planning for multiple robots is a broad field with application-specific methods

- Taxonomies are needed to:
  - allow comparing different methods
  - identify key issues
  - identify trade-offs

Useful taxonomies (proposed by Dudek et al. 1993):

- Communication
- Control distribution
- Group architecture
- Benevolence vs. competitiveness
- Coordination vs. cooperation
- Size
- Composition

Objective of communication: Enable robots to exchange state and environmental information with a minimum bandwidth requirement

Issues of particular importance:

- Information content
- Explicit vs. Implicit
- Local vs. Global
- Impact of bandwidth restrictions
- Awareness
- Medium: radio, IR, chemical scents, breadcrumbs, etc.
- Symbol grounding

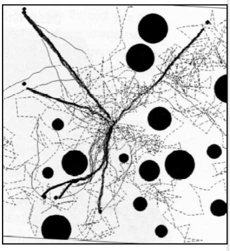


Figure: Balch, Arkin

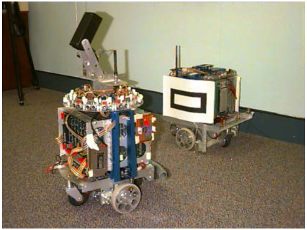


Figure: Jung, Zelinsky

Nature of Communication

Communication: An interaction whereby a signal is generated by an emitter and interpreted by a receiver

- Emission and reception may be separated in time and space
- Signaling and interpretation may be innate or learned (or both)

Cooperative communication examples:

- Pheromones laid by ants foraging food
  - time delayed, innate
- Posturing by animals during conflicts/mating
  - separated in space
  - learned with innate biases
- Writing
  - possibly separated in time and space
  - mostly learned with innate support

Multi-robot Communication

Topology:

- broadcast
- addressed
- tree
- graph

Range:

- none
- near
- infinite

Bandwidth:

- high (communication is essentially "free")
- motion-related (motion and communication costs are about the same)
- low (communication costs are very high)
- zero (no communication is available)

Explicit Communication

- Defined as those actions that have the express goal of transferring information from one robot to another
- Usually involves:
  - Intermittent requests
  - Status information
  - Updates of sensory or model information
- Need to determine:
  - What to communicate
  - When to communicate
  - How to communicate
  - To whom to communicate
- Communications medium has significant impact
  - Range
  - Bandwidth
  - Rate of failure

Implicit Communication

- Defined as communication through the world
- Two primary types:
  - Robot senses aspect of world that is a side-effect of another's actions
  - Robot senses another's actions

Key Considerations in Multi-Robot Communication

- Is communication necessary?
- Over what range should communication be permitted?
- What should the information content be?

Is Communication Needed At All?

- Keep in mind:
  - Communication is not free, and can be unreliable
  - In hostile environments, electronic countermeasures may be in effect
- Major roles of communication:
  - Synchronization of action: ensuring coordination in task ordering
  - Information exchange: sharing different information gained from different perspectives
  - Negotiations: who does what?
- Studies have shown:
  - Significantly higher group performance using communication
  - Communication does not always need to be explicit

Proper approach to communication dependent upon applications

- Communication availability
- Range of communication
- Bandwidth limitations
- Robot language
- ...

Range Communication Should be Permitted

- Tacit assumption: wider range is better
- But, not necessarily the case
- Studies have shown: higher communication range can lead to decreased societal performance

- One approach for balancing communication range and cost (Yoshida '95):
- Probabilistic approach that minimizes communication delay time between robots
- Balance out communication flow (input, processing capacity, and output) to obtain optimal range

Information Content

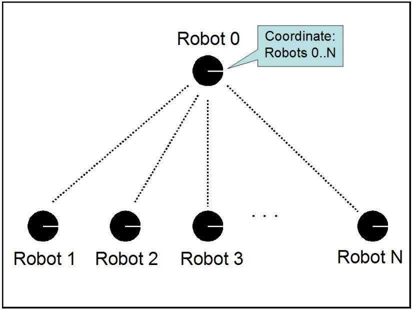
Studies have shown:

- Explicit communication improves performance significantly in tasks involving little implicit communication
- Communication is not essential in tasks that include implicit communication
- More complex communication strategies (e.g., goals) often offer little benefit over basic (state) information (display behavior is a rich communication method)

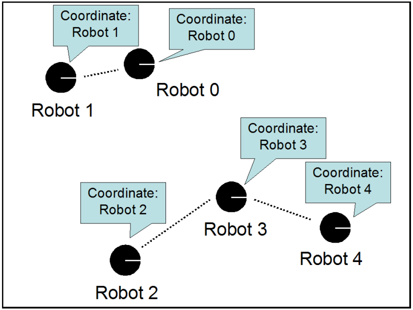
Control Distribution

- Centralized
  - All control processing occurs in a single agent
- Decentralized
  - Control processing is distributed among agents
- Hierarchical
  - Use groups of centralized systems

- Group Architectures are defined by the combination of control distribution and communication topology.
- Simply a different method of classification



Centralized



Decentralized

Benevolence vs. Competitiveness (Stone & Veloso)

- Benevolence
  - Robots work together
- Competitiveness
  - Robots compete for resources
  - Possibly wish to harm one another

Coordination

- When many robots share common resources (e.g. workspace, materials), they must coordinate their actions to resolve conflicts (e.g. collision).

Cooperation

- Many systems strive to incorporate cooperation where robots are working together towards common goals.
- Cooperation requires coordination.

Size

Define size of the multi-robot system:

- a single robot
- a pair of robots
- a limited number of robots
- an infinite number of robots

Scalability

- Describes how amenable the system is to adding more robots.
- Can result in a continuous degradation in performance as opposed to discrete.

Performance

- We can characterize the performance of a system based on the number of robots
- E.g., the number of tasks that can be accomplished in 1 hour.

Interference

- Given limited resources, there is often a plateau or even decrease in performance once a certain threshold of robots is reached.

Composition

Homogeneous

- All robots in the system have similar functionality and hardware.

Heterogeneous

- Robots have varying functionality and hardware.
- Affects maneuverability, tasks achievable, control possibilites, K
- Can lead to robots having roles

The Robot Scout System:

- Used for sensing dangerous/hostile environments

Classifying the Robot Scout with Taxonomies

Communication:

- Wireless RF
- Broadcast with addresses
- Near range
- High bandwith

Control Distribution

- Hierarchical

Coordination and Cooperation:

- Both, but not autonomous

Benevolence vs. Competitiveness:

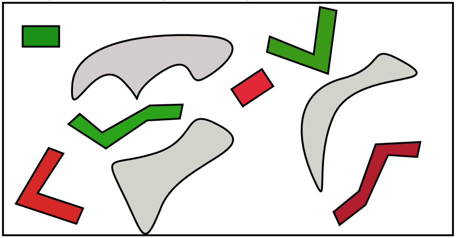
- Benevolent

Size:

- Limited (10)
- Scalable within hierarchies, but not wrt autonomy since more operators required

Composition:

- Heterogeneous



Given

- description of the environment and of the obstacles
- description of several robots $\texttt{Robot}_1, \ldots, \texttt{Robot}_N$
- initial configurations $q_{\text{init}_1}, \ldots, q_{\text{init}_N}$ for each robot
- goal configurations $q_{\text{goal}_1}, \ldots, q_{\text{goal}_N}$ for each robot

compute paths $\texttt{Path}_1, \ldots, \texttt{Path}_N$ such that

- each $\texttt{Path}_i$ starts at q_{init_i} and ends at q_{goal_i}
- each $\texttt{Path}_i$ avoids collisions with obstacles
- robots do not collide with each other, i.e., at each time t it holds that

$$\texttt{Robot}_1(\texttt{Path}_1(t)) \cap \texttt{Robot}_2(\texttt{Path}_2(t)) \cap \ldots \cap \texttt{Robot}_N(\texttt{Path}_N(t)) = \emptyset$$

where $\texttt{Robot}_i(\texttt{Path}_i(t))$ denotes the placement of $\texttt{Robot}_i$ in configuration $\texttt{Path}_i(t)$.

Taxonomy of Multi-robot Path Planning

1. Coupled, centralized approaches:
   - Plan directly in the combined configuration space of the entire robot team
   - Requires computational time exponential in the dimension of the configuration space
   - Thus, only applicable for small problems

2) Decoupled, decentralized approaches:
   - Can be centralized or distributed
   - Divide problem into parts
   - E.g., plan each robot path separately, then coordinate
   - Or, separate path planning and velocity planning

Centralized Multi-Robot Planning Approach

Treat multiple robots as just one robot

Centralized Multi-Robot Planning Approach

Treat multiple robots as just one robot

- Configuration Space $Q = Q_1 \times Q_2 \times \ldots Q_N$
- Plan path in composition configuration space Q

Centralized Multi-Robot Planning Approach

Treat multiple robots as just one robot

- Configuration Space $Q = Q_1 \times Q_2 \times \ldots Q_N$
- Plan path in composition configuration space Q

Advantages

- Off-the-shelf path-planning algorithms can be directly applied
- Guarantees completeness/probabilistic completeness

Disadvantage

- Dimensionality of configuration space increases $\Longrightarrow$ running time increases

Centralized Multi-Robot Planning Approach

Treat multiple robots as just one robot

- Configuration Space $Q = Q_1 \times Q_2 \times \ldots Q_N$
- Plan path in composition configuration space Q

Advantages

- Off-the-shelf path-planning algorithms can be directly applied
- Guarantees completeness/probabilistic completeness

Disadvantage

- Dimensionality of configuration space increases $\Longrightarrow$ running time increases

How would you apply sampling-based path-planning algorithms?

Treat multiple robots as just one robot

- Configuration Space $Q = Q_1 \times Q_2 \times \ldots Q_N$
- Plan path in composition configuration space Q

Advantages

- Off-the-shelf path-planning algorithms can be directly applied
- Guarantees completeness/probabilistic completeness

Disadvantage

- Dimensionality of configuration space increases $\Longrightarrow$ running time increases

How would you apply sampling-based path-planning algorithms?

- $\textsc{GenerateSample}$:
  return$[\textsc{GenerateSample}_1(), \ldots, \textsc{GenerateSample}_N()]$

Treat multiple robots as just one robot

- Configuration Space $Q = Q_1 \times Q_2 \times \ldots Q_N$
- Plan path in composition configuration space Q

Advantages

- Off-the-shelf path-planning algorithms can be directly applied
- Guarantees completeness/probabilistic completeness

Disadvantage

- Dimensionality of configuration space increases $\Longrightarrow$ running time increases

How would you apply sampling-based path-planning algorithms?

- $\textsc{GenerateSample}$:
  return$[\textsc{GenerateSample}_1(), \ldots, \textsc{GenerateSample}_N()]$
  Improve likelihood of generating collision-free samples:

```
for several times do
    generate random samples for all robots
    for several times do
        check which robots are in collision
        generate random samples only for robots in collision
        if no robots are in collision then
            return collision-free sample for all robots
```

Centralized Multi-Robot Planning Approach

Treat multiple robots as just one robot

- Configuration Space $Q = Q_1 \times Q_2 \times \ldots Q_N$
- Plan path in composition configuration space Q

Advantages

- Off-the-shelf path-planning algorithms can be directly applied
- Guarantees completeness/probabilistic completeness

Disadvantage

- Dimensionality of configuration space increases $\Longrightarrow$ running time increases

How would you apply sampling-based path-planning algorithms?

- $\textsc{GenerateSample}$:
  return$[\textsc{GenerateSample}_1(), \ldots, \textsc{GenerateSample}_N()]$
  Improve likelihood of generating collision-free samples:

```
for several times do
    generate random samples for all robots
    for several times do
        check which robots are in collision
        generate random samples only for robots in collision
        if no robots are in collision then
            return collision-free sample for all robots
```

- $\textsc{GeneratePath}(q_A, q_B)$:
  return$[\textsc{GeneratePath}_1(q_{A_1}, q_{B_1}), \ldots, \textsc{GeneratePath}_N(q_{A_N}, q_{B_N})]$

Decentralized Multi-Robot Planning Approach

[proposed by O'Donnell and Lozano-Perez 1989]

- Plan paths for each robot independently of other robots
- Coordinate robot paths so that collisions among robots are avoided

Decentralized Multi-Robot Planning Approach

[proposed by O'Donnell and Lozano-Perez 1989]

- Plan paths for each robot independently of other robots
- Coordinate robot paths so that collisions among robots are avoided

Advantage

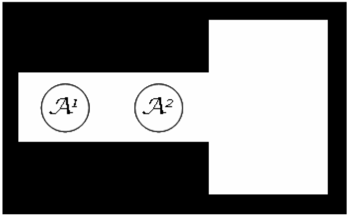
- Dimensionality of configuration space does not increase

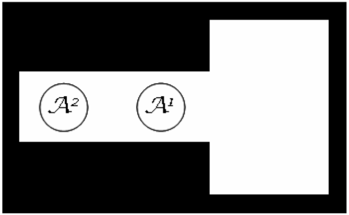
Disadvantage

- Coordination not always possible $\Longrightarrow$ decoupled planning is incomplete

Types of Decoupled approaches

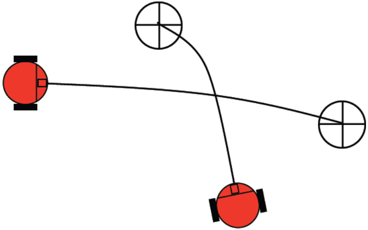
- Path coordination
  - Plan independent paths for each robot
  - Plan velocities to avoid collisions (velocity tuning)
- Prioritized planning
  - Consider robots one at a time, in priority order
  - Plan for robot i by considering previous i-1 robots as moving obstacles





- Velocity tunning can be considered a path coordination strategy
- Goal is to construct independent robot paths that are collision free of obstacles by modifying velocities of robots following their paths so robots will not collide

- Example: Despite intersecting, the following pair of paths are velocity tunable
- Implementation: through time parameterization



Path Coordination in Decoupled Planning

Presented by O'Donell and Lozano-Perez in "Deadlock-Free & Collision-Free Coordination of Two Robot Manipulators"

Task:

- Coordinate trajectories of 2 robots

Method:

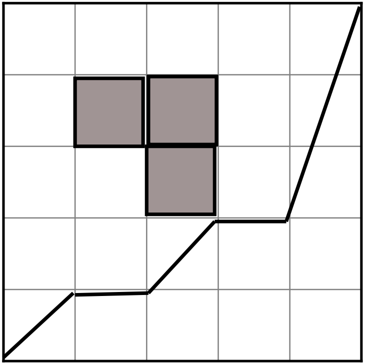
- Plan a path for each robot independently
- Let the path be comprised of many path segments
- Coordinate asynchronous execution of the path segments

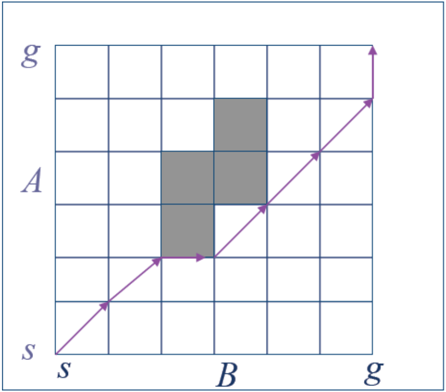
Problems with Coordination:

- Avoid collisions and deadlock
- Gets harder for $n > 2$ robots

Coordination diagram for $\texttt{Path}_1$, $\texttt{Path}_2$

- 2D grid with horizontal (vertical) axis corresponding to time for $\texttt{Path}_1$ ($\texttt{Path}_2$)
- cell (i, j) is marked as "forbidden" iff the i-th segment of $\texttt{Path}_1$ collides with the j-th segment of $\texttt{Path}_2$
- coordination is achieved by selecting any non-decreasing curve that avoids the "forbidden" cells and connects the lower-left corner to the upper-right corner





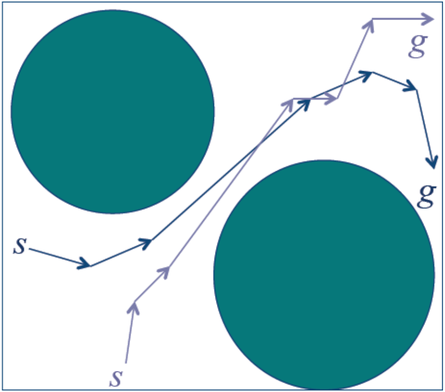


Figure: Task completion diagram and sample path

Prioritized Multi-Robot Planning Approach

in-between centralized and decentralized planning

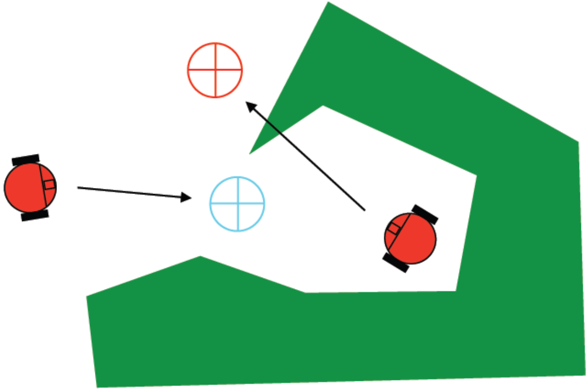
- Robots sequentially construct trajectories.
- As each robot constructs its trajectory, it will use previously constructed trajectories as obstacles to avoid.

for $i = 1, \ldots, N$ **do**
    plan path for robot i to avoid collisions with obstacles
    and avoid collisions with paths planned for robots $1, \ldots, i-1$

Example: Three robots where robot 0 has highest priority and robot 2 has the lowest.

- Construct robot 0's trajectory.
- Construct robot 1's trajectory, considering robot 0 as an obstacle to avoid.
- Construct robot 2..s trajectory, considering robot 0 and robot 1 as obstacles to avoid.

- The priority is of critical importance
  - Example: inside robot needs priority

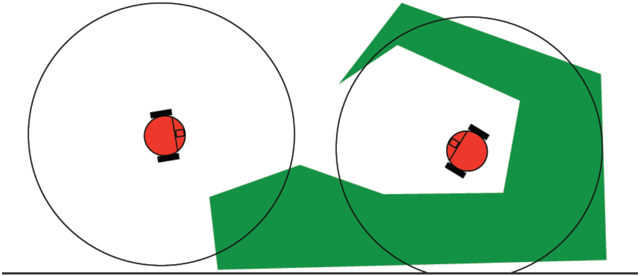


Static vs. Dynamic Priority Systems:

- Static: priorities stay constant over time.
- Dynamic: priorities change over time, either to reflect each individual robot's current value to a mission, or the degree of planning difficulty.

Determining priorities dynamically:

- Can determine each robot's degree of planning difficulty based on the amount of occupied space surrounding the robot.



Priority-based Planning: Centralized vs. Decentralized

Centralized Case: in central planner

for $i = 1, \ldots,$ nRobots **do**
    assign to robot i priority $p[i]$ where p is an integer
for $i = 1, \ldots, \mathrm{nRobots}$ **do**
    construct trajectory for robot i, using robots $i, \ldots, i-1$ as obstacles to avod

Decentralized Case: for robot i

Broadcast robot i's priority bid
Receive priority bids
Determine robot i's priority
Receive trajectories from robots of higher priority
Construct trajectory using received robots' trajectories as obstacles to avoid
Broadcast trajectory to other robots of lower priority

Lots of types of motion coordination:

- Relative to other robots:
  - E.g., formations, flocking, aggregation, dispersion
- Relative to the environment:
  - E.g., search, foraging, coverage, exploration
- Relative to external agents:
  - E.g., predator-prey, target tracking, pursuit
- Relative to other robots and the environment:
  - E.g., containment, perimeter search
- Relative to other robots, external agents, and the environment:
  - E.g., evasion, soccer

Natural flocks consist of two balanced, opposing behaviors:

- Desire to stay close to flock
- Desire to avoid collisions with flock

Why desire to stay close to flock?

- In natural systems:
- Protection from predators
- Statistically improving survival of gene pool from predator attacks
- Profit from a larger effective search pattern for food
- Advantages for social and mating activities

- Flocks, Herds, and Schools: A Distributed Behavioral Model, Craig Reynolds, Computer Graphics, 21(4), July 1987, pgs. 25-34.
- The Nerd Herd, Mataric, 1994
- Stupid Robot Tricks: A Behavior-Based Distributed Algorithm Library for Programming Swarms of Robots, James McLurkin, Master's thesis, M.I.T., 2004.

- Idea: use local controls to generate desired global behavior
- Robots are 12" long, have 4 wheels, bump sensors around body, and radio system for localization, communication, data collection, and kin recognition
- Fundamental principle: Define basis behaviors as general building blocks for synthesizing group behavior
- Set of basic behaviors:
  - Avoidance
  - Safe-wandering
  - Following
  - Aggregation
  - Dispersion
  - Homing
- Combine basic behaviors into higher-level group behaviors:
  - Flocking
  - Foraging

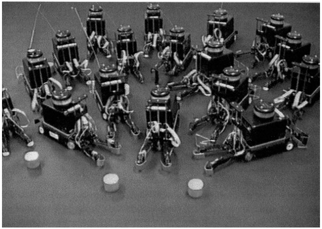


Figure: The Nerd Herd, Mataric, 1994

[movie: NerdHerd]