# Multiagent Systems

Lecture 5:

Multi-Robot Coordination

Research Topics

Kagan Tumer, kagan.tumer@oregonstate.edu

---

# Motivations

- Exploration in hazardous environments
  - Under water
  - On distant planets
  - Inside damaged buildings
- Tasks beyond the limits of single robots
  - Cooperative lifting
  - Assembling large complex structures
- Tasks that can be completed more rapidly with multiple robots
  - Collecting trash in a large office building
  - Searching for mines
- Distributed sensing (thousands of sensors)
  - Studying complex ecosystems (tree tops)
  - Detecting temperature changes in the ocean

Kagan Tumer, kagan.tumer@oregonstate.edu

# Historical Notes

- Electromechanical tortoises, 1950, W. G. Walter
  - Vacuum tube technology
  - Moved toward a light when there was a light
- Multiple manipulators, 1980s
  - Two arms grasp the same object
  - Actions of one robot arm constrain the actions of the other
- Multiple manipulators part 2, 90s and beyond
  - Mobile robots grasp an object without grasping (Stanford)
  - Moving a sofa
  - Box pushing
- Behavior based robots, Mataric
- Robocup, 1998 onward

Kagan Tumer, kagan.tumer@oregonstate.edu

# Control issues

- Centralized and hierarchical
  - Army
  - Factory

  - Advantage: well defined
  - Disadvantage: no redundancy

- Decentralized and local control
  - Ants

  - Advantage: fault tolerant, role redistribution
  - Disadvantage: difficult to control

Kagan Tumer, kagan.tumer@oregonstate.edu

# Centralized Control

- Applicable when the controllers can be placed in a position to observe and communicate with all robots
- Useful when:
  - Individual robots would have to be larger than practical
  - Overall positional sensing is limited
  - Manufacturing costs are high

- Example: Warehouse applications
- USC prototype: electronic assembly using robots

Kagan Tumer, kagan.tumer@oregonstate.edu

---

# Distributed Control

- Applicable when the robots will need to take independent actions
- Useful when:
  - Separation in space
  - Time lag
  - Redundancy is relevant
  - Cost of single point of failure outweights cost of robots

- Example: exploration robots
- CMU prototype: planetary exploration rovers

Kagan Tumer, kagan.tumer@oregonstate.edu

# Multi robot architectures

- Maja Mataric: robots interact based on primitives:
  - Avoid collisions
  - Follow other robots
  - Disperse to achieve desired separation
  - Aggregate to achieve desired proximity
  - Home in on signal
  - Flock (move coherently without a leader)
- Displays foraging behavior with minimal intelligence and communication

Kagan Tumer, kagan.tumer@oregonstate.edu

---

# Multirobot architectures

- Ranger-Scout architecture
  - Two or more drastically different sized robots
  - Larger robot carries smaller robots
  - Marsupial style (Murphy 2002)
- Cellular robotics
  - Simple robots (cells)
  - Cellular robotic systems (Beni 88, Beni, Hackwood 92)
  - Similar to swarms

Kagan Tumer, kagan.tumer@oregonstate.edu

## Communication among Robots

- Point to point communication
  - Individual robots communicate with one another
  - Expensive (power, computation)
  - Information overload

- Broadcast
  - Robots broadcast information
  - Only robots within a range receive broadcast
  - Broadcaster may not know who received information

- Communication via the environment
  - Messages implicit
  - Turn of a light after reaching it
  - Leave trail on the path

---

## Cooperation without Communication

- Foraging task can be achieved with no communication (Balch and Arkin 92, 94)

- Now add grazing and consuming to the robot actions and study three forms of communication
  - No communication
    - Each robot uses only own sensor to detect other robots, goals and obstacles
  - State communication
    - Robots communicate their internal states to other robots
  - Goal communication
    - Robots communicate location of targets to one another

# Formation Control

Organized movement of a team of robots (formation flying):

- Local information
  - Unit center referenced: Robots determine their position with respect to the center of the robots
  - Leader referenced: Robots determine their position with respect to the leader’s known position
  - Neighbor referenced: Robots determine their position with respect to their immediate neighbors (Kroo, 2000)
  - Friend referenced: Robots determine their position with respect to the position of a “friend” (Mataric 2002)
- Global approach

Kagan Tumer, kagan.tumer@oregonstate.edu

---

# Formation Control

Organized movement of a team of robots (formation flying):

- Local information
- Global approach
  - Leader following: Robots receive motion of a leader
  - Virtual structure: Entire formation is treated as a single structure that needs control

Kagan Tumer, kagan.tumer@oregonstate.edu

## Robot Soccer

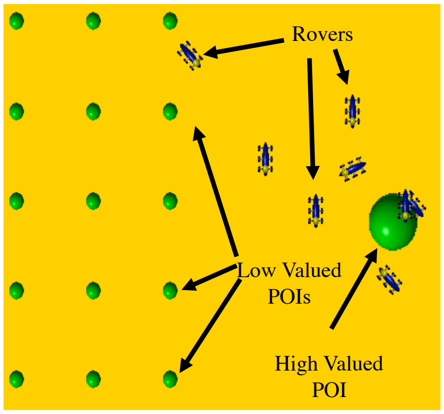
- Simulation League
  - 2 D
  - 3 D
- Small size robot league (18 cm, 5 per team)
- Middle size robot league (50 cm, 4 per team)
- Standard size robot
  - Identical platforms (Software competition)
  - Was four legged competition in previous case
- Humanoid league
  - Currently: penalty kick, 2 vs. 2

> By the year 2050, develop a team of fully autonomous humanoid robots that can win against the human world soccer champion team.

Kagan Tumer, kagan.tumer@oregonstate.edu

---

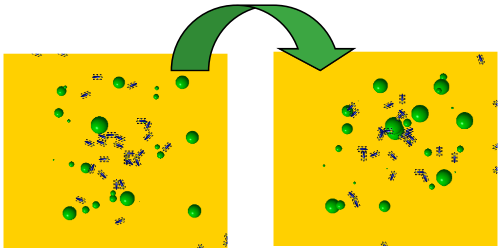
## Recall: Rover Coordination



$$G = \sum_t \sum_j \frac{V_j}{\min_i \delta(L_j, L_{i,t})}$$

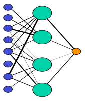
$$\delta(x, y) = min\{\|x - y\|^2, d^2\}$$

- Rovers observe points of interest (POIs)
  - POIs vary in value, time and place
  - Get more information closer to POI
  - Only primary observation counts
- Learning problem
  - Rovers learn in single trial (non-episodic)
    - Dynamic: POIs appear/disappear
  - Rovers reset at regular intervals (episodic)
    - Static: POIs the same in each episode
    - Dynamic: POIs different in each episode



# Challenges

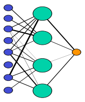
- How to design an adaptive control mechanism for problems with:
  - Continuous state spaces
    - Need to generalize

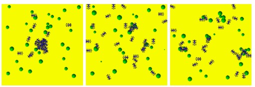


---

# Challenges

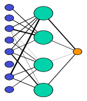
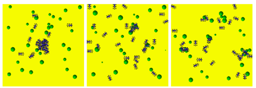
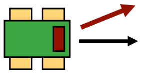
- How to design an adaptive control mechanism for problems with:
  - Continuous state spaces
    - Need to generalize
  - Dynamic environment
    - Need to learn sensor to actions mapping





# Challenges

- How to design an adaptive control mechanism for problems with:
  - Continuous state spaces
    - Need to generalize
  - Dynamic environment
    - Need to learn sensor to actions mapping
  - Noisy sensors/actuators
    - Need to be robust

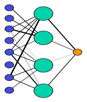
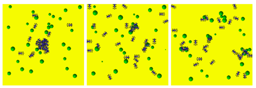
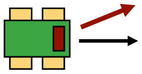
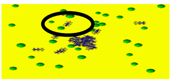




Kagan Tumer, ktumer@mail.arc.nasa.gov

---

# Challenges

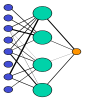
- How to design an adaptive control mechanism for problems with:
  - Continuous state spaces
    - Need to generalize
  - Dynamic environment
    - Need to learn sensor to actions mapping
  - Noisy sensors/actuators
    - Need to be robust
  - Limited communication/observation
    - Need to use local information effectively

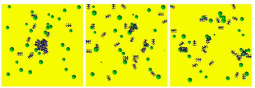
Kagan Tumer, ktumer@mail.arc.nasa.gov

# Challenges

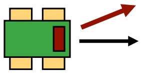
- How to design an adaptive control mechanism for problems with:
  - Continuous state spaces
    - Need to generalize



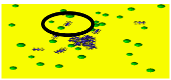
  - Dynamic environment
    - Need to learn sensor to actions mapping



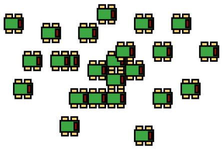
  - Noisy sensors/actuators
    - Need to be robust



  - Limited communication/observation
    - Need to use local information effectively



  - Multiple agents to coordinate
    - Collective action needs to optimize system objective

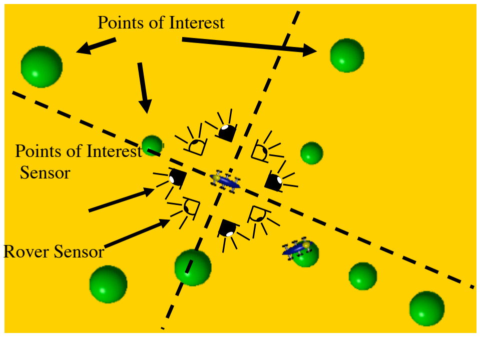


Kagan Tumer, ktumer@mail.arc.nasa.gov

Modeling, Learning and Control Group, NASA ARC

---

# Agent Decision Making

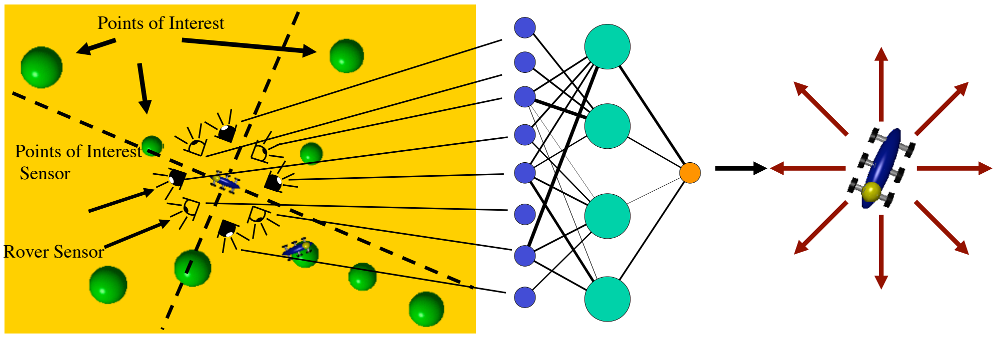


- Rovers observe environment through eight continuous sensors

Kagan Tumer, ktumer@mail.arc.nasa.gov

Modeling, Learning and Control Group, NASA ARC

# Agent Decision Making

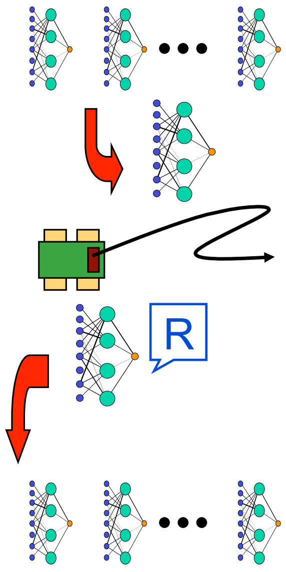


- Rovers observe environment through eight continuous sensors
- Sensors fed into local controller
- Output of controller determines direction/velocity of rover

---

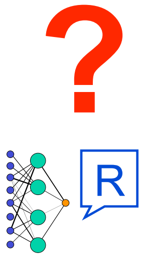
# Recall: Rover Control

1. At t=0 initialize N=10 controllers
2. Pick a controller using $\epsilon$ greedy alg ($\epsilon$=.1)
3. Randomly modify controller
4. Use controller on this agent for 15 steps
5. Evaluate controller performance
6. Re-insert controller into pool
7. Remove worst controller from pool
8. Go to step 2



## Rover Control

5. Evaluate controller performance



Kagan Tumer, ktumer@mail.arc.nasa.gov

Modeling, Learning and Control Group, NASA ARC

---

## Recall: Objective Functions

$$G = \sum_t \sum_j \frac{V_j}{\min_i \delta(L_j, L_{i,t})}$$

Global

(Fully Factored, Low Learnability)

$$P_i = \sum_t \sum_j \frac{V_j}{\delta(L_j, L_{i,t})}$$

“Perfectly Learnable”

(Low Factoredness,     Learnability)

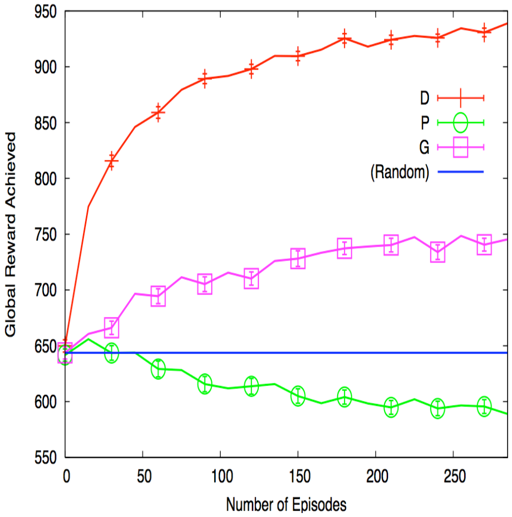
$$D_i = \sum_t \left[ \sum_j \frac{V_j}{\min_{i'} \delta(L_j, L_{i',t})} - \sum_j \frac{V_j}{\min_{i' \neq i} \delta(L_j, L_{i,t})} \right]$$

Difference

(High Factoredness, High Learnability)

# Episodic Learning

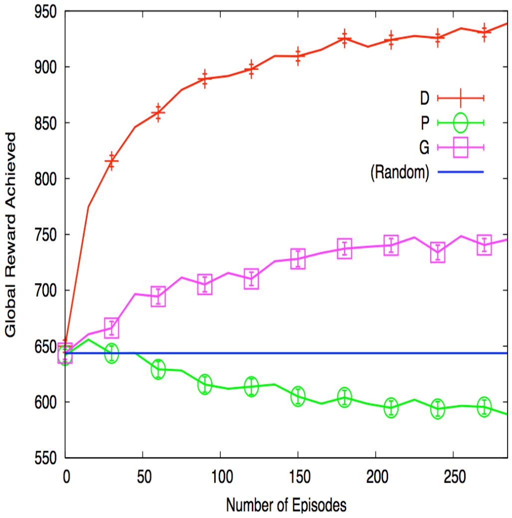
## Dynamic Environment
## 30 Rovers



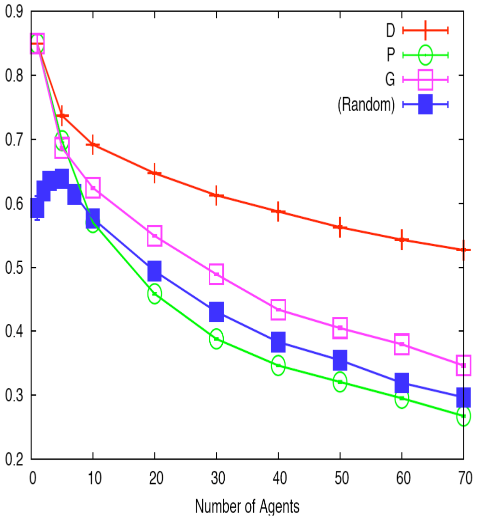
---

# Episodic Learning

## Dynamic Environment
## 30 Rovers

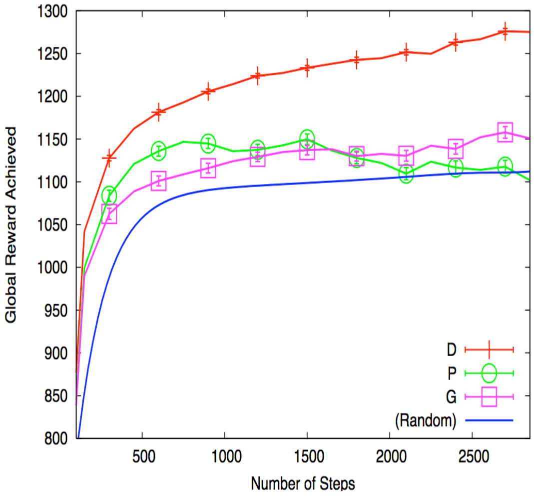


## Dynamic Environment
## Scaling

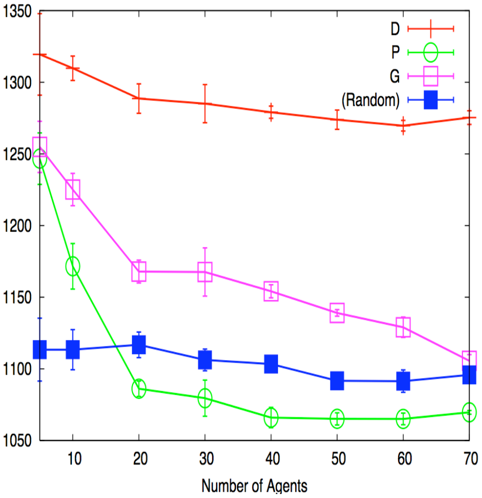


# Non-Episodic Learning

Dynamic Environment
(30 Agents)



Dynamic Environment
(Scaling)

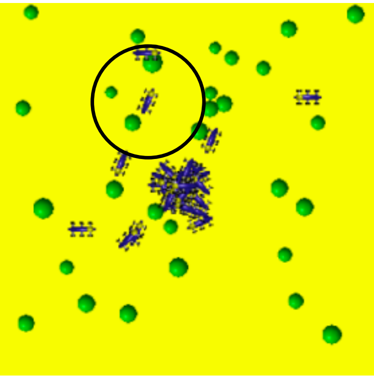


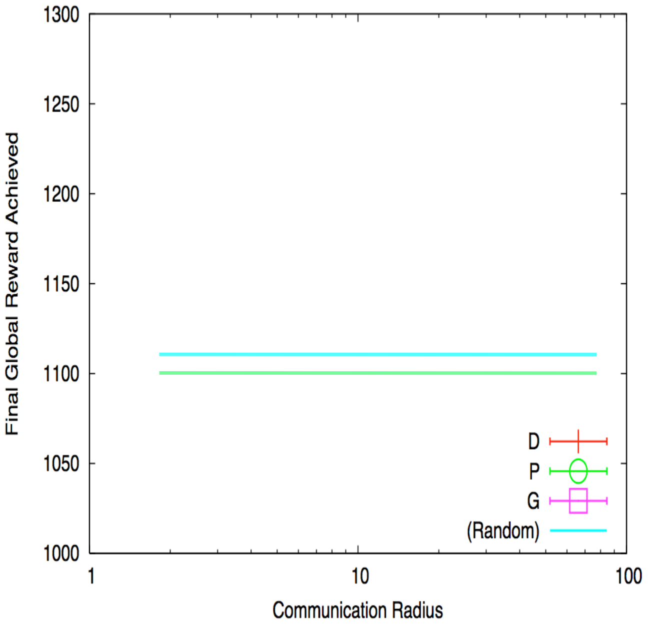
Kagan Tumer, ktumer@mail.arc.nasa.gov

Modeling, Learning and Control Group, NASA ARC

# Communication Limitations

Non-Episodic
Dynamic Environment
70x75 Unit Env
(30 Agents)



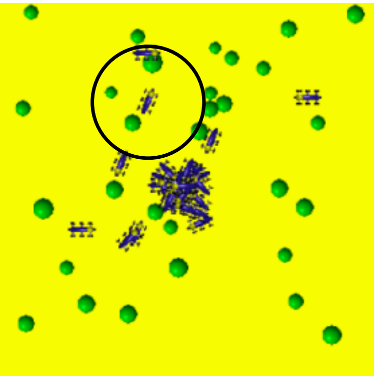


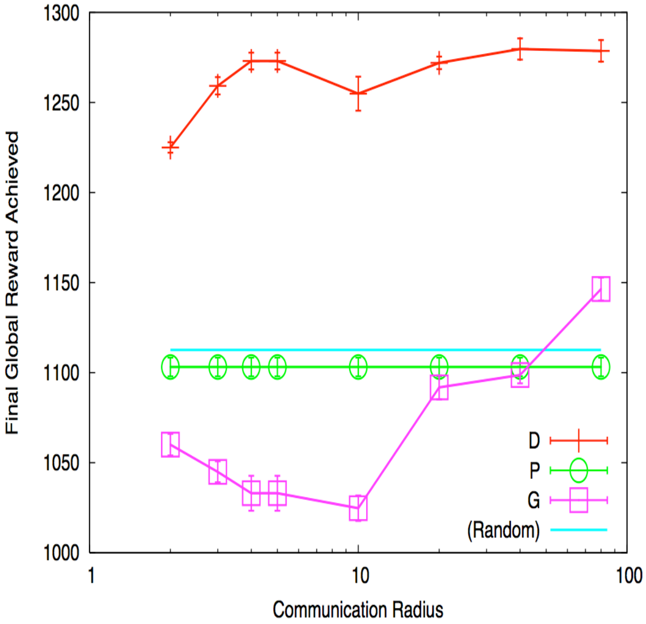
Kagan Tumer, ktumer@mail.arc.nasa.gov

Modeling, Learning and Control Group, NASA ARC

# Communication Limitations

Non-Episodic
Dynamic Environment
70x75 Unit Env
(30 Agents)





Kagan Tumer, ktumer@mail.arc.nasa.gov

Modeling, Learning and Control Group, NASA ARC

# Key Issues in Multi-Robot Systems

- Communication among robots
- Homogeneity vs. heterogeneity
- Task assignment and specialization
- Autonomy of robots
- Reliability
- Robustness
- Control architecture
- Localization
- Formation control
- scalability

Kagan Tumer, kagan.tumer@oregonstate.edu