Robot Learning by Demonstration

A Project Proposal by
Adam Olenderski

Idea

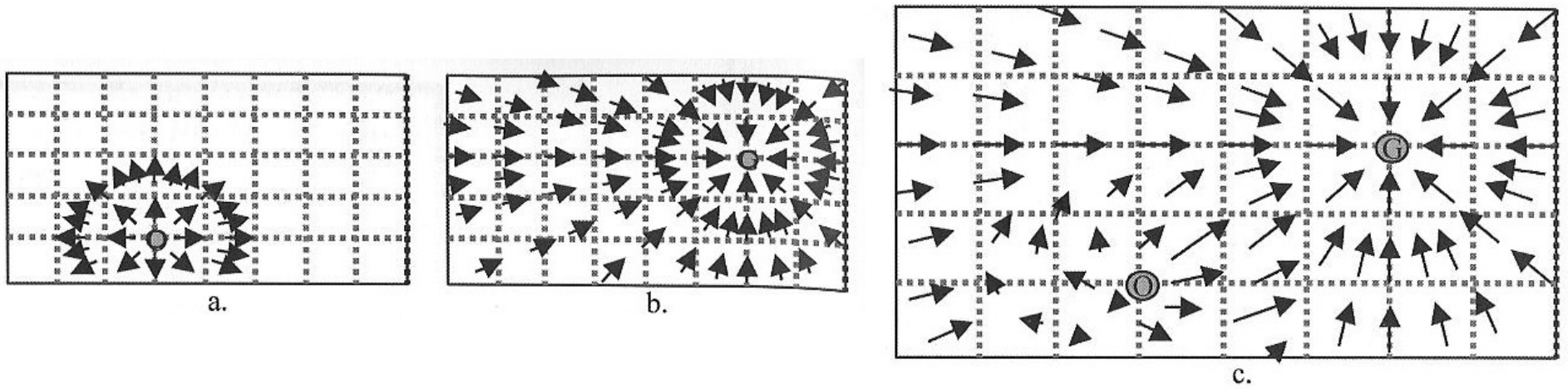
- Use concepts from the study of complex systems to learn the interactions between different behaviors in a behavior-based robot.
- Based on current research
- One paper already published on this topic
  - Olenderski, Adam, Monica Nicolescu, and Sushil Louis. Robot Learning by Demonstration Using Forward Models of Schema-Based Behaviors. Proceedings of the Second International Conference on Informatics in Control, Automation, and Robotics. Vol. III. 2005. 263-269

The Robot

- ActivMedia Pioneer 3DX
- SICK Laser rangefinder
- 16 sonar rangefinders
- Fiducial finder (for detecting goal objects)
- Player/Stage software

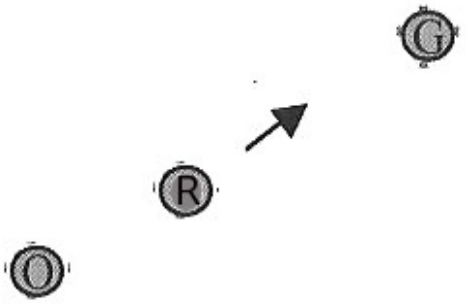
Background: Potential Fields

- Potential fields use vectors to determine the direction and speed of the robot.
- Each object in the world (obstacles, goals) can be thought of as generating a force field that affects the robot when in that object's proximity.



Potential Fields: Continued

- Measurement of the potential field at any given point is robot-centric; the robot only has to generate one vector at any time, not the entire field
  - Quick computation
  - Allows for quick reaction



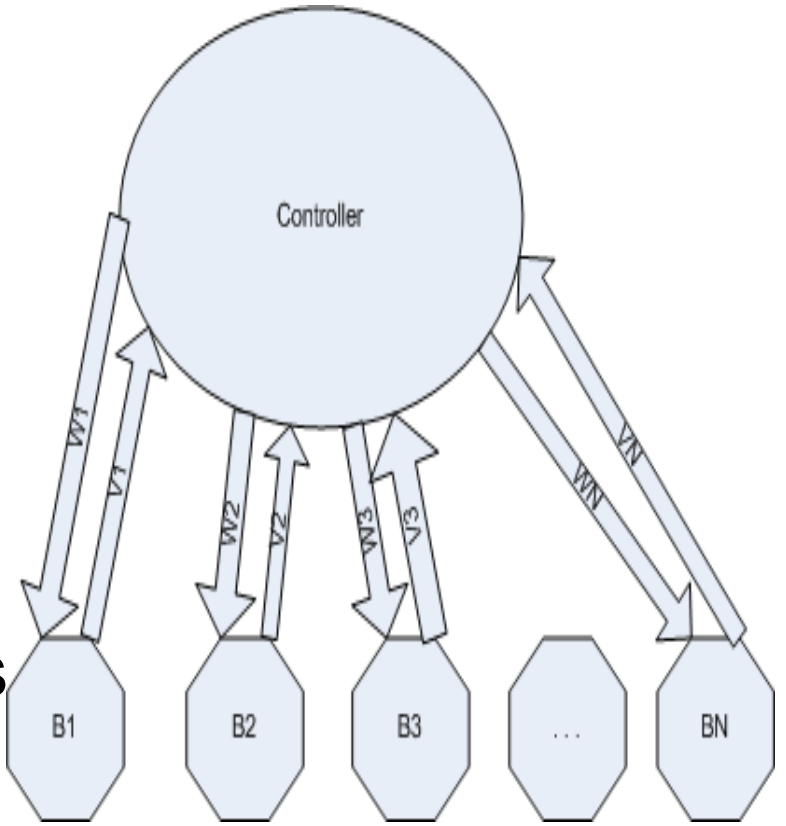
- Each concurrently-running behavior is tasked with generating one type of vector—e.g., an avoid behavior monitors obstacles it needs to avoid, etc.

Combining Behaviors

- In a potential-fields-based approach, the commands sent to the actuators (the motors that control the wheels) consist of a fusion of the directions suggested by the individual behaviors: a vector sum.
- Each behavior is weighted differently to indicate importance
  - Ex: if the user stayed far away from walls and obstacles, the avoid weight should be far higher than the wander or wall-follow weights

Complex System Breakdown

- Many behaviors in a system
- Behaviors communicate directly with Controller
  - Send vectors
  - Receive weights
  - Controller changes weights depending on which behaviors are active
  - Behaviors apply weights to vectors
- Behaviors do not communicate with each other.



The Problem: Determining Weights

- How do we find the weights that most accurately model the user's priorities during the demonstration?
- At each timestep during the demo, record the input from the user as well as a suggestion from each behavior.
- Use a learning algorithm to determine a set of weights that will make the weighted vector sum as close as possible to the user's input for the demonstration

Learning Weights

- Offline: Use the recorded information as input to a neural net, whose output is the set of weights.
- Online: Use small, incremental changes to update the weights during the demonstration in the hope that the resultant controller will exhibit complex behavior (a la cellular automata)

Questions and Discussion