Lecture 19: Developing Autonomy for Robots in Teams

Brett Browning

Autonomous Robots 16-200
Carnegie Mellon University in Qatar
Overview

- Logistics
- The plan
- RoboCup research environment
- My research
Logistics

- **Weekly project meetings**
  - You give a 5 minute update on your progress
  - To be held in Thursday Lab slot
  - Additional help also available on Thursday

- **Homework #4** is due Thursday at 10.30am

- **Change to the syllabus**

- Please take care of the batteries!!!
The Plan

- Today: RoboCup and Brett’s research
- Wednesday & Next week: Manipulation
- Homework #5
- Research projects
  - Weekly meetings during lab session
  - Demonstration, paper, presentation, poster
  - (Remember, start now!)
RoboCup Robot Soccer
Robot Soccer

Challenge to researchers to improve robot intelligence through friendly competition

- “By the year 2050, develop a team of fully autonomous humanoid robots that can win against the human world soccer champion team”, www.robocup.org

First competition in 1997, Nagoya Japan
Small Size Robot League

CMDragons [Browning, Bruce, Bowling, Veloso et al. 03]
Legged League
Mid-Size League
Segway Soccer

Segway Soccer [Browning et al. 04]
Humanoid League
RoboCup Details

- Annual international competitions
  - Next one in RoboCup 2006 in Bremen, Germany
  - Large and growing larger (2,000 competitors in 2006)

- Teams compete in games of soccer
  - Human referee commands translated by computer
  - Fully autonomous during game

- Research communicated via
  - Technical reports, papers, symposium
  - Code releases
Common Challenges

- Autonomous robots with real-time perception
- Operating in a dynamic environment
- Operating in a team with other robots, humans, novel teammates
- Operating with adversaries
  - Creates highly dynamic environment
  - Encourages high performance solutions
  - Encourages strategy adaptation and learning
My Interests

- Real-time vision perception
- Autonomously robot control
  - Individually, and within team
- Effective learning mechanisms
  - Adapt to environment, changes, opponents, task
Let's Focus on Control

Robot must choose actions to perform its role within the team
State Machines for Control

State machines have a long history in robotics, control and AI

- For behavioral control [Brooks 86, Balch et al. 95]
- Hybrid control [Lynch & Krogh,00]
- State estimation [Thrun et al. 05], many more

State abstraction provides a powerful mechanism for describing (and implementing) sequences with different modes of control
Definition of a State Machine

- PS – Perceptual state
- CS – Control states
- CA – Control actions
- T(PS⁺) – Transition function
- R – Termination result

Initial State

Success

Failure
Hierarchies of State Machines for Individual Control

- Really equivalent to a larger state machine
- Allows for state machine reuse (i.e. Macros)
- Allows for natural task decomposition into sub-tasks

Call a sub-skill
Soccer Example

Shoot 1
1. Goto ball
2. Approach ball
3. Aim
4. Kick

Shoot 2
1. Goto ball
2. Steal
3. Dribble
4/5. Aim/Kick
Parallel Execution of Hierarchies

- Operate state machines in parallel
  - Independent or non-conflicting tasks
  - Requires scheduling for conflicting tasks
- Event driven execution, sleeping states
Call a state machine a skill

Idea: We can equate management of skills with a multi-threaded operating system

<table>
<thead>
<tr>
<th>OS Terms</th>
<th>Skill Terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thread</td>
<td>Skill</td>
</tr>
<tr>
<td>Resource management</td>
<td>Action management</td>
</tr>
<tr>
<td>Sleeping</td>
<td>Sleeping</td>
</tr>
<tr>
<td>Signals</td>
<td>Events</td>
</tr>
<tr>
<td>Scheduling</td>
<td>?</td>
</tr>
<tr>
<td>Synchronization</td>
<td>?</td>
</tr>
</tbody>
</table>
Skill Learning

Key idea:
- Skill structure and kernel creates natural basis for applying learning

Learning applicable in three ways
- Learning control policy in a state
- Learning state transitions
- Learning hierarchy/which skills to call
Skill Selection Learning

Key idea:
- In a state a skill may call multiple sub-skills to do the same task
- Learn which sub-skill works best

Approach
- Use success/failure history as reward signal
- Apply an expert systems technique to learn which ‘expert’, or skill, is best
Skill Selection Approach

Treat a skill as an ‘expert’

Call one of N sub-skills

Tie success/failure to reward signal
Coordinating Robots: Plays

Synchronized state machines

Role 1

Role 2
Play Manager

- Manages selection, execution and synchronization for each play
- Beyond synchronization
  - Play selection
  - Dynamic role assignment
  - Monitoring
  - Learning
An Example Play

APPLICABLE offense
DONE aborted !offense

ROLE 1
pass 3
mark best_opponent

ROLE 2
block

ROLE 3
pos_for_pass R B 1000 0
receive_pass
shoot A

ROLE 4
defend_lane
Playbook Strategy

Play Manager handles multiple plays, and select as appropriate for world state

Learn which plays work better

\[ w_1 = 0.7 \quad w_2 = 6.7 \quad w_N = 0.07 \]

Play 1
APP: offense
Role 1:
• dribble
• pass
Role 2:
...

Play 2
APP: offense
Role 1:
• dribble
• fake
• pass
Role 2:
...

Play N
APP: defense
Role 1:
• block
Role 2:
...
Summary

State abstraction is a powerful technique that can be used at many levels from single robot to team coordination:

- Management of state machines equivalent to multi-threaded OS management
- Provides a natural basis to apply learning
- Provides a natural mechanism for task decomposition
End of Lecture

See you on Wednesday