Cooperative Robotics – Robot Soccer System

The overall robot soccer system

Chess versus robot soccer

<table>
<thead>
<tr>
<th></th>
<th>Chess</th>
<th>Robot soccer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environment</td>
<td>Static</td>
<td>Dynamic</td>
</tr>
<tr>
<td>State change</td>
<td>Turn taking</td>
<td>Real time</td>
</tr>
<tr>
<td>Information accessibility</td>
<td>Complete</td>
<td>Incomplete</td>
</tr>
<tr>
<td>Sensor readings</td>
<td>Symbolic</td>
<td>Non symbolic</td>
</tr>
<tr>
<td>Control</td>
<td>Central</td>
<td>Distributed</td>
</tr>
</tbody>
</table>
Control Structure

- Role level: Determines the roles of each robot – defender, attacker and goal keeper
- Action level: Selects actions of each robot – shooting, blocking, dribbling, etc.
- Behavior level: Move and obstacle avoidance
- Execution level: Motor control
Vision based soccer robot system

- Uses a hybrid control structure
  - Hierarchical structure
    - easy to design the control algorithm (modular)
    - higher level: for effective cooperation and coordination
    - lower level: for faster and stable control algorithm
  - Reactive behavior structure
    - fast adaptability to changing environments

Control structure (Soty – Korea)

- Highest level
  - For efficient cooperation among (equivalent) agents
  - Easy to model other agents’ actions
  - can dispense with communication among robots

Role Level

- Assigns roles & areas
- Selected action (action level)

Action Level

- Basic actions
  - Pr/Cr (role and behavior levels)

Ensures reliability
- Degree of achievement: proper action assignment among agents
- Pr – reference posture, Cr – desired circle
Basic Action (#17)

- **Primitive actions**
  - Stop
  - Wandering
  - Sweep_Ball
  - Locate_Ball

- **Attacking actions**
  - Shoot
  - Cannon_Shoot
  - Turning_Shoot
  - Spin_Shoot
  - Position_To_Shoot

- **Defending actions**
  - Push_Ball
  - Position_To_Push_Ball
  - Screen_Out_Ball

- **Goalkeeper actions**
  - Block_Ball
  - K_Defense_Goal,
  - K_Default_Position
  - Keeper_Attack
  - K_Need_Escape_Goal

---

Basic Actions – Example 1

- **Wandering**
  - to wander around playground when no specific action is allotted to

- **Screen_Out_Ball**
  - to block opponent robots or ball

- **K_Defense_Goal and K_Default_Position**
  - to hold goal keeper’s position against attacking robots

---

Basic Actions – Example 2

- **Shoot action**
  - highest priority (most important)
  - selects this action when a robot is in a good position to shoot

---

Basic Actions – Example 3

- **Cannon_Shoot action**
  - conditions to be satisfied:
    - robot’s orientation vector directed towards goal area
    - its projection falls within the goal width, and
    - the ball is located within the robot path along its trajectory.
Behavior level

- Obstacle avoidance (reactive), circular_motion and move behaviors
- Uses reactive behavior to adapt to varying environment

Move behavior

- Basic behavior
  - A kind of position controller
  - Input
    - desired posture \((x, y, q)\) from action level
  - Output
    - right and left motor velocities (cm/sec)

Circular_motion behavior

- Useful for Spin_Shoot and Turning_Shoot actions
- Generates circular trajectory
  - center at \((Cx, Cy)\) with radius \(Cr\) and direction \(Cd\)

Obstacle avoidance behavior

- Entirely different from move behavior
- Action level command enables this behavior
- Obstacles
  - Ball, home robots and opponent robots
- Boundary areas for obstacle avoidance depends on
  - the size of obstacle and
  - the degree of its danger

Pr: reference posture
Pc: Current posture
Execution level

- Lowest level
- Velocity control of robot actuators (2 DC motors)
- Fastest sampling time for velocity control

Action Level Strategy

- Zone defense
  - each robot has its own area of action and role
  - a robot operates within its own area.
- Zone defense - drawbacks
  - a robot may get blocked by an opponent robot
    - solution: role level
  - difficult to reach the ball when it is located in one of the boundary areas
    - solution: variable zone defense (introduce common area)

Variable Zone Defense

- Common area
  - two robots allowed to operate within this area
  - obstacle avoidance behavior takes care of conflicts, if any

Role level strategy

- Role level allows role and area of operation to robots as per the current situation
  - fault state
    - blocked by opponent robots
    - robot malfunctioning
    - weak battery
  - when goalkeeper robot breaks down, the role level makes
    - one of the attack robots as goalkeeper, and
    - the other as striker with the complete playground as its area
Vision-based soccer robot system

The control algorithm:
  - low level: kinematics
    - simple and fast (less calculation time)
    - real time control
  - high level: strategy
    - to avoid other robots and to compete with opponents
    - a discrete event system (DES) represented in Perti-net
    - easy to add robot agents (expandable/flexible)
    - reinforcement learning

Control Structure

Supervisor

- States
  - P1: Robot1 attacks and robot2 defends
  - P2: Robot1 defends and robot2 attacks
  - P3: Robot1 defends
  - P4: Robot1 attacks
  - P5: Robot1 and robot2 attack
  - P6: Robot2 defends
  - P7: Robot2 attacks
  - P8: Robot1 and robot2 defend

- Transitions
  - T1: Robot2 in a position to attack
  - T2: Robot1 in a position to attack
  - T3: Robot1 defending and commanded to attack
  - T4: Robot1 attacking and commanded to defend
  - T5: Robot2 attacking and commanded to defend
  - T6: Robot2 defending and commanded to attack

Attack mode control

- Path planning using vector field
  - heuristically generated vector field as per ball position and target point.
  - modified vector field near boundaries.
  - Robots controlled to follow the vector field
Defense mode control

States assumed:
- d1: robot behind the ball.
- d2: robot kicks the ball.
- d3: ball behind robot and slightly away.
- d4: ball behind and close to robot.

Goalie control

Goalie states
- g1: at goal center
- g2: in target position
- g3: in intercept position
- g4: in kicking position

Robot 1 – States

Robot 2 & Ball – States

Total states = 5 (robot1: distance) x 7 (robot1: angle) x 3 (robot2: distance) x 9 (ball: location) = 945 states
Actions

- Action0: with role change
- Action1: without role change

Strategies

- Flexible zone defense
- Flexible border line modifies the zones

Team Player Roles

Conceptual strategy for 1-3 formation

The playfield is divided into five (5) different areas and each of the five (5) players is given an exclusive area in which they may operate. First in the 1-3 formation, the field is divided into two parts; one for the defensive and one for the offensive. The offense contains three players, and the offensive part of the field is divided into 3 zones; left, central, right.
1-3 Strategy

- In the 1-3 strategy, the team is composed of players who perform the roles of keeper, defender or attackers
- A 1-3 formation with 1 defender, 3 attackers, and a keeper
- This type of behavior will bring cyclic (rotational) movement for the attackers

State diagram for attacker

State diagram for the keeper

State diagram of defender

role dedicated to scoring

role dedicated to prevent scoring
Attacker 1, 1-3 formation

Attacker 2, 1-3 formation

Attacker 3, 1-3 formation
Circular targets for rotational attackers

Different time-action combinations for different attackers, when the ball is in the offense

Targets for circular movement in defense

With three attackers
Different time-action combinations for different attackers, when the ball is in the defense

<table>
<thead>
<tr>
<th>Time at 0 - 1 sec</th>
<th>Attacker 1</th>
<th>Attacker 2</th>
<th>Attacker 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Defend</td>
<td>Return</td>
<td>Panic shoot</td>
</tr>
<tr>
<td>Time at 1 - 2 sec</td>
<td>Panic shoot</td>
<td>Defend</td>
<td>Return</td>
</tr>
<tr>
<td>Time at 2 - 3 sec</td>
<td>Return</td>
<td>Panic shoot</td>
<td>Defend</td>
</tr>
</tbody>
</table>

Successful clearing

Successful saving

Low pressure defense

Defender taking offensive action
Quadrants for forward attacker


Fuzzy Behaviour-Based Decision Control of Mobile Robots

Behavior decomposition
Behavior decomposition

Robot Soccer System
Architectures and Strategies

Dr. Prahlad Vadakkepat, PhD
Associate Professor, Electrical & Computer Engineering
National University of Singapore
E4-08-19

E: prahlad@nus.edu.sg   T: 6516 2296
www.vadakkepat.com