Robot Soccer

How to build a scoring machine

Team: Yue Wang; Hayden Bader; Branch Vincent; Bill Sun; Gao Tang

Contents

Background

Why Robot Soccer?

RoboCup provide a common task for evaluation of various theories, algorithms and agent architectures.

In order for robots to play a soccer game well, wide range of technologies need to be integrated and numbers of technical breakthrough must be accomplished.

	Chess	RoboCup
Environment	Static	Dynamic
State Change	Turn taking	Real time
Info. accessibility	Complete	Incomplete
Sensor Readings	Symbolic	Non-symbolic
Control	Central	Distributed

Industrial Demand: The feature of RoboCup is to fulfill the demand to foster technologies for the next generation industries.

Background

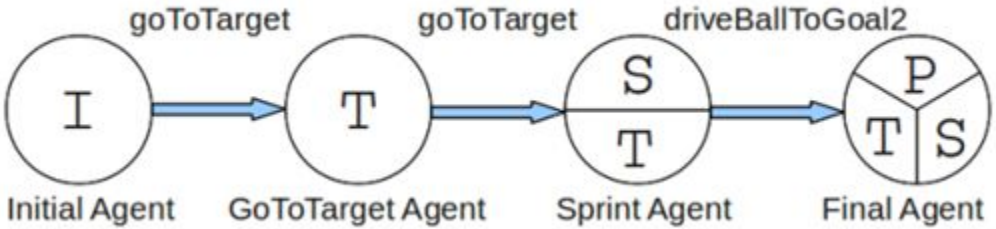
Main Architecture For robot soccer:

- ***Learning Behaviors:*** It is not feasible the program the robots against all situations, so robot learning Method seems promising.
- ***Multiagent Collaboration:*** A soccer game is a specific and very attractive real-time multiagent research from the viewpoint of distributed agent research: Each team has a team-wide common goal and goals of two teams are incompatible.
- ***real-time sensor fusion***
- ***motor control***

System of UT Austin

2011 Basic System:

learning-based omnidirectional walk engine



As optimizing 40 real-valued parameters can be impractical, a carefully chosen subset of 14 parameters was selected for optimization while fixing all other parameters.

System of UT Austin

2012:

Behavior and Strategy

Select suitable speed and prepare for a kick action concerning the relative position between opponent or ball.

1. Opponent far away: turn around and stop to make a long-range kick.
2. Opponent within a threshold: do not turn around, use a stronger and slower kick to keep ball out of opponent.
3. Opponent within a closer radius: using walk engine kicks.
4. Opponent directly at the ball: make a quick side kick to pass to ball to area with more teammates.

System of UT Austin

2014: Kicking and Passing

Kicking:

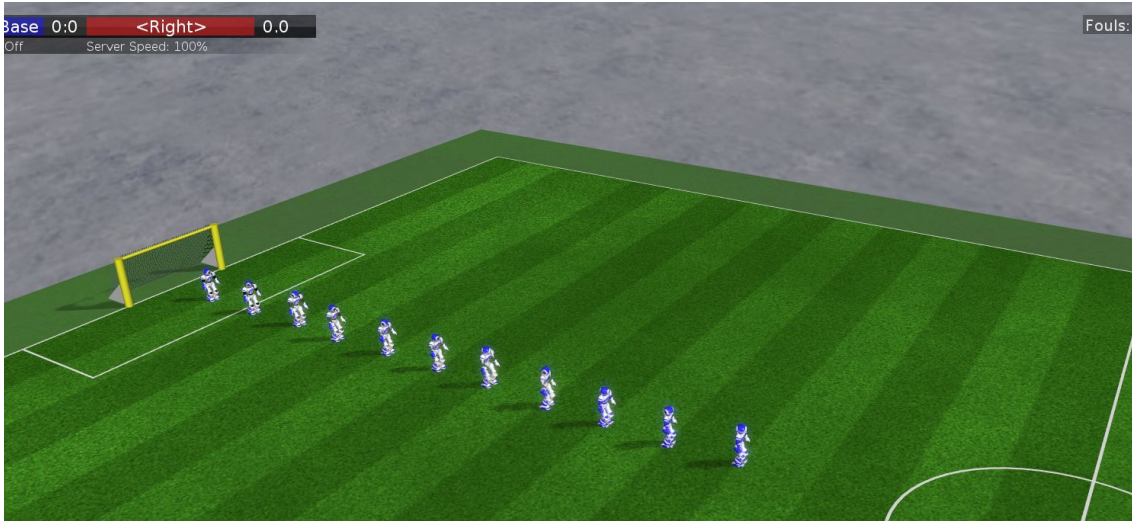
- Introduce new parameter sets in walk engine to make robot stop within a bounding box around the ball so it will not overshoot or crash on the ball.

Passing with Kick Anticipation

- The robot can evaluate where to kick to ball if it can not kick to the goal directly by the relative position of opponents and goal.
- The robot will broadcast the predicted position of the ball when it decide to kick it. Other agents will be assigned to positions to catch to ball by dynamic role distribution.

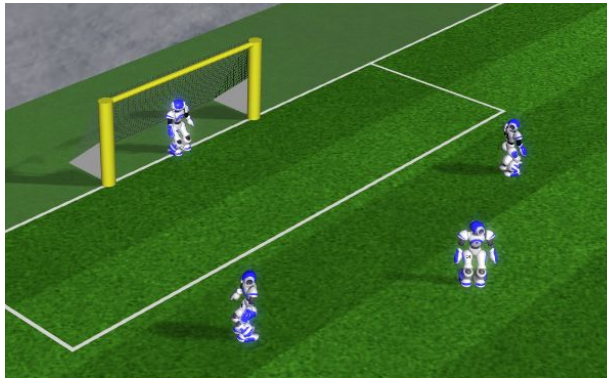
Our Implementation

- Three main components
  - Goalie (1)
  - Defense (3)
  - Offense (7)
- Offensive-based formation



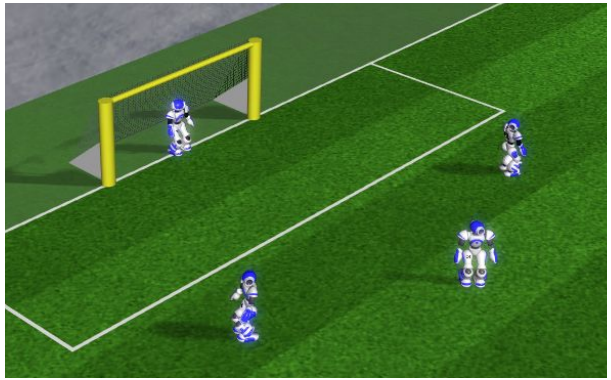
Our Implementation - Goalie

- Clear ball
  - *Condition*: Near ball, distance less than 1m.
  - *Behavior*: Go to ball and kick ball to midfield
- Defend goal
  - *Condition*: Not near ball, distance greater than 1m.
  - *Behavior*: Move along goal line to face and align with ball. Position is the intersection of the goal line with the line from the back of the goal to the ball.



Our Implementation - Defense

- Clear ball
  - *Condition*: Near ball, distance less than 5m.
  - *Behavior*: Go to ball and kick ball to midfield
- Defend goal
  - *Condition*: Not near ball, distance greater than 5m.
  - *Behavior*: Form an arc around our goal. Arc's midpoint intersects the line from the goal line's midpoint to the ball. Defenders always facing the ball.



Our Implementation - Offense

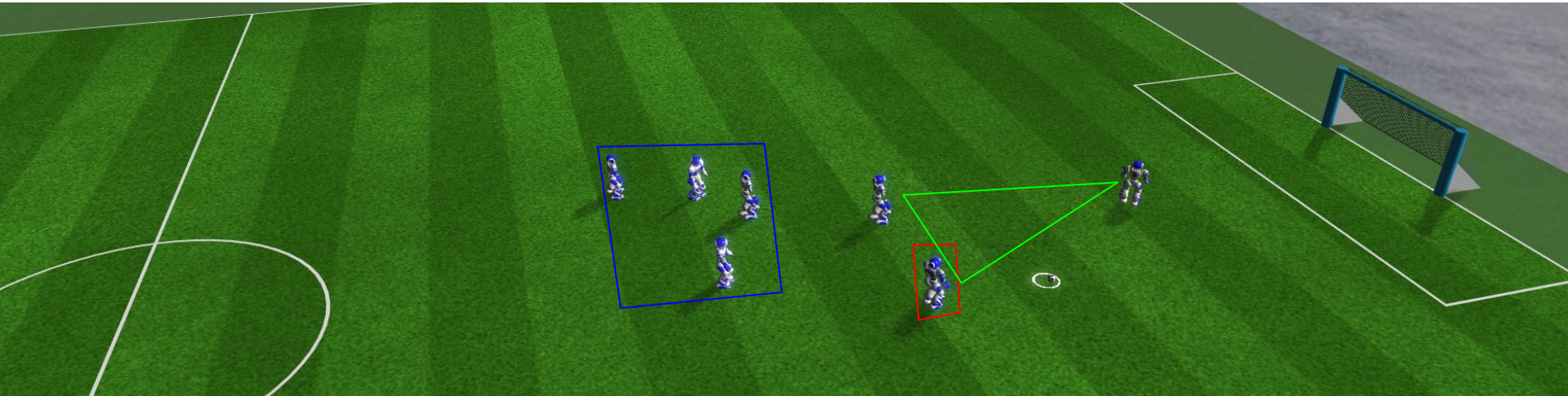
Attacker (nearest to ball) ---

- Obtain ball
  - *Condition*: Does not have ball
  - *Behavior*: Get ball
- Shoot
  - *Condition*: Has ball and near goal
  - *Behavior*: Shoot at goal's center
- Pass
  - *Condition*: Has ball and player closer to goal (within passing distance)
  - *Behavior*: Pass
- Dribble
  - *Condition*: Has ball and otherwise
  - *Behavior*: Dribble to goal

Our Implementation - Offense

Strong supporters (2nd/3rd nearest to ball) ---

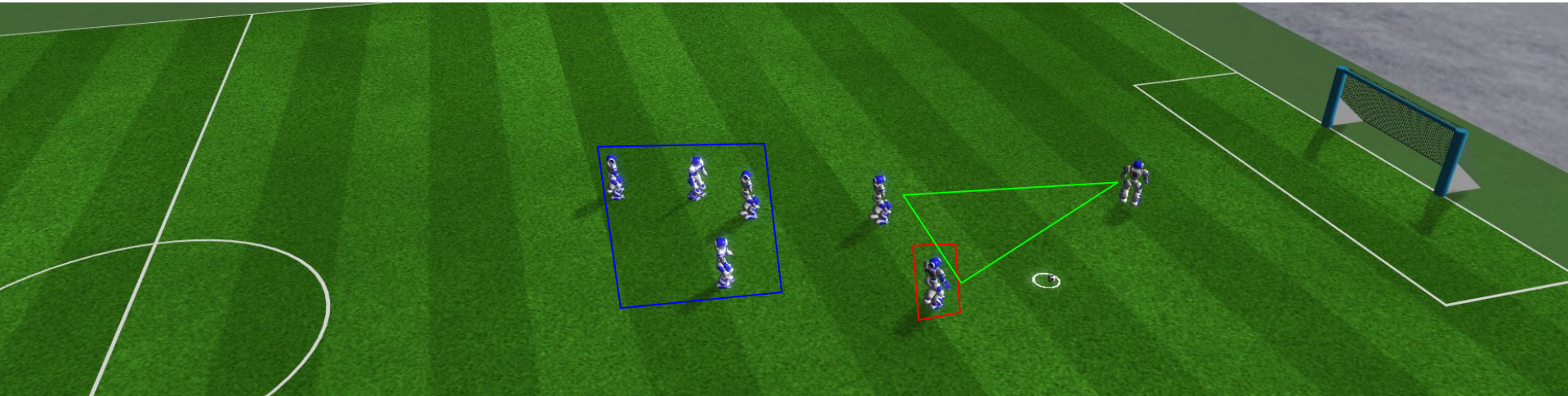
- Support attacker
  - *Condition*: Always
  - *Behavior*: Follow behind closest player (forming a triangle)



Our Implementation - Offense

Weak supporters (all others) ---

- Support rest of offensive team
  - Controlled by a potential field
  - Repulsed by teammates/opponents; Repulsed/attracted to opponent's goal
  - Attracted to ball if far



Evaluation

- Problems
  - Attacker frequently collides with opposing defense
  - Passing without accounting for defenders
  - Shooting did not account for goalie
  - Goalie did not account for shot’s trajectory
  - Do not consider (most) rules
- Solutions
  - Ensure chosen attacker is standing up
  - Add potential field to attacker to avoid opponents
  - If attacker does not possess ball, hover around ball until clear (reduces collisions)
  - Choose desired shooting target to maximize distance from opponents
  - Consider teammate’s movement during a pass
  - Only pass when ball’s trajectory is sufficiently clear of opponents
  - Goalie should track and intercept shot’s trajectory (e.g. via kalman filter)
  - Optimize parameters by simulating performance

Evaluation Part 2

- Problems
  - Teammates to be passed to were slightly out of range in real-game scenarios
  - When defending, teammates on offense stayed upfield
  - When defending, robots went to the ball instead of to a place in front of the ball
    - Result: It took a long time to regain the ball
- Solutions
  - Do more testing to evaluate a better passing range.
    - Selected range was 6m. 5m was the maximum observed practice kicking distance
  - Don’t pass only to the teammate that is closest to the opponent’s goal
    - Alternatively, attempt to pass more quickly
  - Dynamically adjust the potential field to better control positioning
    - Ex: If the ball is on our side, there should be less of a pull towards the opponent’s goal
  - Track the ball’s velocity each time step and react accordingly

Future work

- Robot Soccer
  - Learn from human
  - Adaptive strategy
  - Reinforcement learning
- Distributed robotics system
  - Make each robot intelligent
  - Adaptive strategy
  - Use learning techniques
  - Formulate optimization problems to improve performance