Cooperation-Based Behavior Design

Hui Wang, Han Wang, Chunmiao Wang, and William Y.C. Soh

Division of Control & Instrumentation, School of EEE
Nanyang Technological University
Nanyang Avenue, Singapore 639798
{p147513815, hw, p149510969, eycsoh}@ntu.edu.sg

Abstract. Robot soccer explores such a research topic that multiple agents work together in a Real-time, Cooperative and Adversarial (RCA) environment to achieve specific objectives. It requires that each agent can not only deal with situations individually, but also present cooperation with its teammates. In this paper, we describe a robot architecture, which addresses "scaling cooperation" among robots, and meanwhile allows each robot to make decisions independently in real-time case. The architecture is based on "ideal cooperation" principle and implemented for Small Robot League in RoboCup. Experimental results prove its effectiveness and reveal several primary characteristics of behaviors in robot soccer.

1 Introduction

Our work is based on Small Robot League in RoboCup [1], which includes two five-robot physical teams that play a twenty-minute game on a field with the size of a table tennis board. It presents many challenges for physical agent [2]:

(1) For an individual agent, it should deal with the infinite motion states related to the moving objects in a real-time case.

(2) For a collection of agents, they are expected to play like a team, which means extra abilities beyond simple reactive behaviors, such as situation recognition, communication and cooperative behaviors.

In this paper, we aim to explore the cooperation mechanism in a RCA environment. Our approach is to present an agent architecture, in which cooperation is embedded. Section2 describes the agent architecture. Section3 presents an implementation of this architecture for Small Robot League. Experiment results in Section4 prove its effectiveness and reveal some main features of behaviors in robotic soccer. In the end, we will summarize important conclusions and discuss many crucial problems for further study.

2 Architecture Overview

Before we define an architecture for agents surviving in a RCA environment, we must understand what determines our design. Arkin indicates , in [3], that "If the roboticist intends to build a system that is autonomous and can successfully

A. Birk, S. Coradeschi, and S. Tadokoro (Eds.): RoboCup 2001, LNAI 2377, pp. 465–470, 2002.

compete with other environmental inhabitants, system must find a stable niche or it (as an application) will be unsuccessful.", which shows a logical way for us to decide what an agent should be, that is, environment and task require agents to have many essential functions for their survival; the agent architecture is therefore requested to fully support those functions' strong points and suppress their weak ones.

2.1 Ideal Cooperation

Although cooperation has various types, there should be some criteria to evaluate their performance since cooperation is observable for us. Four standards are therefore presented, and if certain cooperation can meet all of them, it can be called *ideal cooperation*.

We believe that for a team of agents living in a RCA environment, if they execute ideal cooperation, the cooperation should be *reliable*, *scaling*, *voluntary* and *evolutionary*. "Reliable" has double-fold meanings: first, when the same situations appear, a certain cooperation behavior will be executed repeatedly; second, if one teammate stops working due to failures, the team performance will degrade elegantly and if one more teammate is added, the team performance will improve. "Scaling" means cooperation can be local (*micro-cooperation*) or global (*macro-cooperation*). Micro-cooperation is often achieved by a few agents within a relatively small area, where they can affect the same situation directly. Macro-cooperation is always concerned with all the team members scattered around the space. "Voluntary" means agents can independently reason the opportunities of cooperation, that is, an agent can be influenced by its teammates, but never dominated by any of them. It prevents serious damages to the whole team caused by partial failures of individual agents. "Evolutionary" means the performance of certain cooperation can be improved by training.

2.2 Agent Architecture

This section presents an architecture for agents surviving in RCA environment with which an agent can make its own decision, meanwhile display cooperation with others. A high level block diagram of the architecture is shown in Fig. 1 (Shadowed area represents one agent, which is "connected" with its teammates and the environment by dashed lines).

The "Receiver" is divided into two parts: "Communication Unit" and "Perception Unit", which monitor the outside environment and extract useful information for decision-making. The difference between them is that perception unit is responsible for sensing the external states, but the communication unit receives the message from other teammates. A point is the output of receiver is the information interpreted rather than raw data. The function of *opponent and teammate modeling* is often realized in interpreter.

Decision machine is the kernel of an agent responsible for deciding what should be done in the next step(s). It has two functional units: "Information

Fusion Unit" and "Behavior Selection Unit". The former collects all the information related to the agent and performs situation recognition. The latter selects suitable behaviors according to the output of Information Emergence Unit and sends out the desired action(s). "Behavior Library Unit" and "Internal State

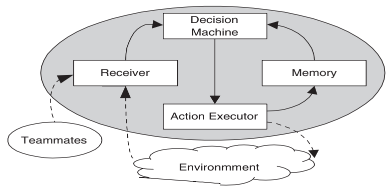

Fig. 1. Agent general architecture

Unit" consist of Memory of a robot. Internal State Unit saves the agent's personal data temporarily (it is always updated in each step). Behavior Library is a long-term memory whose contents are preset in design. **Cooperation in a team is just expressed as specific behaviors for each agent**.

Action Executor translates the selected behaviors into concrete control commands and performs them.

3 Implementation in Robot Soccer

Robotic soccer is a good example of RCA environment. We implement the architecture presented in Section 3 for RoboCup Small Robot League. The key in implementation is to construct a behavior library based on cooperation and design a decision process that can estimate the cooperation opportunities independently.

3.1 Behaviors Based on Cooperation

The behaviors of a robot are organized in a layered form, and each layer may have *behavior group* with different priorities. The higher level behaviors are supported by those in the lower levels. Table 1 is a summary.

3.2 Decision-Making

Although a robot will consider other teammates when it takes actions, the robot will never be dominated by others, that is, every robot may influenced by its teammates, but such influence is achieved by message or prediction rather than orders. As for our robot team, we adopt such a belief to design the process of decision that the behaviors of dealing with the ball have higher priorities than the others. The entire decision process is broken into two phases as follows:

Phase 1 - For all of the team members

Step 1.1: Determine which robot is to deal with the ball immediately (it is called Robot A).

Step 1.2: Robot A selects the suitable behavior.

Step 1.3: Robot A broadcast its intention to its teammates.

Phase 2 - For each of the left robots,

Step 2.1: Check the message from Robot A.

Step 2.2: Determine whether to help Robot A by modeling the opponents and teammates. If YES, go to Step 2.3 or go to Step 2.4.

Step 2.3: Select behaviors from micro-cooperation behavior group.

Step 2.4: Select behavior from macro-cooperation behavior group.

Step 2.5 Return to Step 1.1.

Level 0 – Basic Motion			
Name	**Function**	**Related behaviors**	**Feature**
Direct Move	Move to a position without consideration of orientation	(None)	Pure reactive without intentions
Quick Stop	Stop as soon as possible at any speed	(None)	
Turn	Tune direction in an exact angle	(None)	
Spin	Turn around as fast as possible	(None)	
Level 1 - Advanced Motion			
Name	**Function**	**Related behaviors**	**Feature**
Approach	Include the ball within the controllable area as soon as possible	Direct Move; Quick Stop; Turn	Pure reactive without intentions
Smooth Move	Move with desired speed and direction	Direct Move; Turn	
Level 2 –Individual Behavior			
Name	**Function**	**Related behaviors**	**Feature**
Kick	Push the ball to in a desired direction	Direct Move; Spin	Intentional but no consideration about others
Block	Hinder the motion of opponents	Smooth Move	
Intercept	Get the ball when it's moving	Smooth Move	
Dribble	Control the ball in a controllable area	Smooth Move	
Unstuck	Leave corners, borders of the play ing ground or opponents when necessary	Smooth Move	
Level 3.1 – Micro-cooperation Behavior Group			
Name	**Function**	**Related behaviors**	**Feature**
Pass	Push the ball to an open area where another teammate stays.	Kick; Smooth move	Intentional and with consideration about other teammates in a local area
Block-to-Kick	Help others finish kicking by blocking opponents	Kick; Smooth move; Block	
Block-to-Dribble	Help others finish dribbling by block opponents	Dribble; Smooth move; Block	
Maximize ball control	Look for the chance to get the ball by wait at a desired place	Smooth move;	
Level 3.2 – Macro-cooperation Behavior group			
Name	**Function**	**Related behaviors**	**Feature**
Scatter	Return to the default area	Smooth move	Intentional and with consideration about other teammates in a local area
Formation change	Rearrange the robots' default positions	(None)	

Table 1. Layered behaviors based on cooperation

4 Experiments

4.1 Experiment Design

In order to check the effectiveness of our cooperation-based architecture, we present six teams with different abilities to compete each other. They are defined as in Table 2. During the development, we find out that there are several inherent weak points in a pure physical platform [4]. Therefore, a simulation platform is implemented to test the influence of certain behavior on overall team performance. The same Basic Motion listed in Table 1 is used by each team, however the performance is different. A match has two halves, each of which lasts 10 minutes. "Long" response time means 20% more time added in decision; "Common" motion accuracy means 20 % random noises added in robot's motion (for both position and orientation). Micro- and Macro-cooperation are the same for the teams that have them. Each team has the same default initial position for each team member. However, for each game, the initial positions of robots are slightly changed by adding 5% white noises. It intends to make various beginning situations.

Name	**Response time**	**Motion Accuracy**	**Strategy**
Team1	Long	Common	1)Chase the ball and push it to the opponent goal; 2)Only one robot deals with the ball at any time; 3)Always the robot that is nearest the ball is selected to deal with it.
Team2	Short	Good	(Same as Team1)
Team3	Long	Common	Micro-cooperation only
Team4	Long	Common	Micro-cooperation and macro-cooperation
Team5	Short	Good	Micro-cooperation only
Team6	Short	Good	Micro-cooperation and macro-cooperation

Table 2. Design of the six teams in experiment

4.2 Results and Discussion

The results of totally 50 games are recorded (refer to Table 3). We use the Ratio of Win (RW), Average Goal (AVG) and Average Loss (AVL) to evaluate the team performance.

Opponent	**RW (%)**	**AVG**	**AVL**		Team2 V.S. Team5	32	1.4	4.7
Team1 V.S. Team2	0	0.2	8.3		Team2 V.S. Team6	28	1.1	4.2
Team1 V.S. Team3	14	1	8.1		Team3 V.S. Team4	48	1.2	1.3
Team1 V.S. Team4	12	0.7	8.0		Team3 V.S. Team5	2	0.3	9.3
Team1 V.S. Team5	0	0.1	10		Team3 V.S. Team6	0	0.25	9
Team1 V.S. Team6	0	0	9.5		Team4 V.S. Team5	0	0.4	11
Team2 V.S. Team3	92	5.5	1.5		Team4 V.S. Team6	0	0.1	8.7
Team2 V.S. Team4	86	4	1.3		Team5 V.S. Team6	46	3.1	3.3

Table 3. Competition Scores

Summarily, we order the teams according to their performance as follows: Team6 $\geq$ Team5 $>$ Team2 $>$Team4 $\geq$ Team3$>$Team1. Here, "$\geq$" means "slightly better than"; "$>$" means "obviously better than".

From the experiment results, it can be summarized as: (1) All the teams with better performance in Basic Motion defeat the teams with common performance. (2) All teams with cooperation behaviors defeat the teams without cooperation behaviors. (3) The teams with both micro-cooperation and macro-cooperation behaviors are just slighter better than the teams with micro-cooperation behaviors only.

(1) means **a behavior that is frequently used has great influence on the team's overall performance**. Basic Motion represents the personal ability of robot, which is the foundation of other behaviors and therefore used at any time. So, we must try our best to enhance the capacities of individual robots when developing a soccer robot team.

(2) means **cooperation is an essential mechanism for robots in RCA environment if they intend to complete goals more efficiently and robustly**.

(3) may lead to a misunderstanding that macro-cooperation is not so effective as macro-cooperation. We cannot get such a general conclusion since the scale of team is not considered in experiments, which may be a potential factor. A more reasonable explanation to (3) is **the behaviors closely related to dealing with the ball directly impose heavy affects on team performance**.

5 Conclusions and Future Work

Beginning with a cooperation-embedded architecture, we construct a team of robots able to make decisions independently to look for cooperation opportunities in both local areas and the global area. Experiment results preliminarily prove the effectiveness of our design.

We have stated that ideal cooperation should be evolutionary. How to improve the cooperation performance is our next objective. A common method is to use machine learning. Previous researches put focus on simulation [5] and middle-size physical robot with local vision [6]. There is still few report about success in small-size robot application. The possible challenges include: "How to evaluate cooperation performance quantitatively?", "How to applied learning in situation recognition?" and "How to obtain learning samples in physical environment effectively?"

References

1. Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., Osawa, E., and Matsubara, H. RoboCup: A Challenge Problem for AI. AI Magazine , 18(1): pp73-85, 1997.
2. Minoru Asada, Peter Stone, Hiroaki Kitano, Barry Werger, Yasuo Kuniyoshi, Alexis Drogoul, Dominique Duhaut, Manuela Veloso, Hajime Asama and Sho'ji Suzuki. The RoboCup Physical Agent Challenge: Phase I. Applied Artificial Intelligence (AAI), Volume 12, 1998
3. Ronald C. Arkin. Behavior-based robotics. pp51, MIT Press, 1998.
4. Hui Wang, Han Wang, Yew Tuck Chin, William Y. C. Soh, A Multi-Platform for Robot Soccer Team Development. Tech. Report of Robot Soccer of NTU, 2000
5. Peter Stone. Layered Learning in Multi-Agent Systems: A Winning Approach to Robotic Soccer. MIT Press, 2000.
6. M. Asada, S. Noda, S. Tawaratsumida, and K. Hosoda. Vision-based behavior acquisition for a shooting robot by using a reinforcement learning . In Proc. of IAPR / IEEE Workshop on Visual Behaviors-1994, pp 112-118, 1994.