STEP2D-05 Team Description

Alexey Krichun[1], Sergey Serebryakov[2], Anton Ivanov[3], Lev Stankevitch[1]

[1]"RosElectroProm Holding" Company
7, ul. Elektropultovtsev, St. Petersburg, 195030, Russia
alexy_kr@mail.ru, stankevich_lev@inbox.ru
[2] Saint Petersburg Institute for Informatics and Automation of the
Russian Academy of Science
39, 14-th liniya V.O., St. Petersburg, 199178, Russia
sergey_s@iias.spb.su
[3] Dr. Web Ltd
80, Nevsky av, St. Petersburg, 195248, Russia
ai@drweb.com

Abstract. STEP2D-05 Soccer Simulation team is an evolution of previous years team STEP. This year enhancements are focused on using cognitive approach in inter-agent cooperation at middle and high level behavior functions such as pass function and scenario implementation. In this paper we describe two cognitive functions used in our agent.

Introduction

STEP2D-05 Soccer Simulation team is an evolution of previous years team firstly named ERA-Polytech, then Zenit-New ERA, and then STEP (Soccer Team of ElectroPult) participated in RoboCup Soccer Simulation competitions since 2002. Team is being developed at RosElectroprom Holding (ElectroPult Plant Company belongs to the Holding) in cooperation with St. Petersburg Institute for Informatics and Automation of Russian Academy of Sciences.

Team code was initially based on CMUnited-99 source code [1], but now only networking module and world model modules of CMUnited are used in team code.

This year enhancements are focused on using cognitive approach in inter-agent cooperation at middle and high level behavior functions such as pass function and scenario implementation.

Cognitive approach

Soccer agents are implemented using cognitive approach [2, 3]. The soccer agent has multilevel architecture. The main structural components of agent levels are cognitive modules that are capable to mapping cognitive functions defining agent's behavior. Well-known formal neuron and neural network models can be used for implementation of the cognitive modules. However, these models have significant

limitations related to kind of the functions. Special models using adaptive functional approximation were developed and examined. Among them cluster model based on neuro-fuzzy logic can be used to build learning modules of the soccer agents.

The lower and middle levels of the soccer agent define skills and individual behavior of the agent. Interaction of the agent with others to fulfill teamwork is determined by its upper coordination level. The special cognitive functions that realize the coordination mechanisms of teamwork are used for agent's cooperation. They have to be tuned by multi-agent learning that allows the agents to learn how to cooperate and compete. Adaptive cooperation of the soccer agents allows providing high efficiency for the teamwork. Usage of the developed architecture and uniform cognitive components that can specialize and adapt by tuning, allows the soccer agents to effective perform the teamwork in the changing environment.

As examples, we describe two cognitive functions used in our agent.

Pass function implementation

Pass is the most important skill that the soccer agent can have to make teamwork. The team's craft fully depends on this skill, because it's the main part of offense. The pass skill can be divided into 2 parts:
- determining whether the pass to a teammate will be successful, i.e. the ball will be intercepted by desired player;
- determining the player who is the most desired receiver in the current situation.

One of the ways to realize pass is to pass the ball during the game only directly to a teammate. In this case the pass function can be simple but it has following disadvantages:
- Only routes that follow directly to the teammates are considered as possible pass routes.
- All opponents not situated between the ball and the teammate, aren't considered as possible interceptors during the pass calculations.

To develop an algorithm for determining the pass success, which will work fast and will suit for the moving objects (partners or opponents), we need to collect examples of successful interceptions (offline) and afterwards (online, during match) use neuro-fuzzy logic module to approximate between got examples for determining interception point. We can define function

$$n = I(\alpha, \overline{v}_{ball}, d, \overline{v}_{player}),\qquad\qquad (1)$$

(where α – angle between ball velocity and player body, $\overline{v}_{ball}$ – ball velocity, d – distance between player and ball, $\overline{v}_{player}$ – player velocity), which returns the number of simulation cycles, needed for a particular player to intercept the moving ball. Using this function, we can calculate the interception cycles for a teammate and for all opponents, what will tell us, whether the pass will be successful or not for any pass route.

Using any method, we can gather information about all successful passes. Let's mention, that any pass variant is described by the pair (α, v), where α – pass angle, v – pass speed.

Next problem we must solve is the selection of the best variant from the set of all possible passes. In generally, any pass variant can be valued by some criteria, for example:

– "the less congested interception point";
– "the closest interception point to the opponent's goal" and much others.

So, all we need is to assign priorities to all possible variants. For this, we have to define function

$$y = K(x_i), i = \overline{1, n}, \tag{2}$$

where n is number of pass variants, x_i is i-th pass variant, which assigns priority to every of pass variants due to some criterion. Afterwards, the best pass can be found as:

$$x^{opt} = \arg\max_{i=1,n} K(x_i) \tag{3}$$

In general, many parameters affect pass choosing algorithm. We may want to take into account several criteria, not one. These reasons lead us to use the aggregated criterion, for example, additive convolution:

$$x^{opt} = \arg\max_{i=1,n} \sum_{j=1}^{m} a_j K_j(x_i), \sum_{j=1}^{m} a_j = 1 \tag{4}$$

where K_j is j-th criterion, a_j is j-th weight coefficient for criterion K_j.

The criteria computations can take much time. Using neuro-fuzzy system allows to greatly reduce time needed for computations.

Cognitive function for multi-agent scenario control

The task of learning to act in the multi-agent environment is being considered in the context of the scenarios. Formally, scenario consists of two parts – conditions of initialization, and control system, or policy (the rules of acting in the scenario). In this chapter the experiment, in which the original rule-based control system was replaced with proposed system, is considered.

Since the most important way of using the scenarios is the using them in the opponent's penalty area, appropriate scenario was chosen. Three players are involved in this scenario – two teammates, trying to score the goal, and one opponent – goalie, defending the goal. The task of learning is in the determining such policy, allowing teammates to score the goal for minimal time:

$$\pi^* = \arg\max_{\forall \pi \in \Pi} K \tag{5}$$

where
$K = K(\pi)$ - function, determining the length of any episode, if agent follows policy π
Π - set of all possible policies;
π^* - optimal policy.

    Only one agent learns to act – agent, possessing the ball (first teammate). This agent starts the scenario. It must learn what to do in every step of the any episode – dribble to goal or pass ball direct to teammate. The second teammate, having received the ball, just shoots it right in the goal along optimal shooting direction. The episode finishes successfully, if the goal is scored, and it's considered as failing, if goalie catches the ball. All other possible terminal states are considered as errors, and aren't processed.

    With the respect to criteria (time minimization), following rewards were taken: for every episode step the reward is $r_{forEpisodeStep} = -0.001$, for successful end the reward is $r_{forSuccessEnd} = 1$, and in the case of fail the reward is $r_{forFailureEnd} = -1$. In the following figure 1 the typical snapshot of the episode is shown. Also it contains information about the state representation:
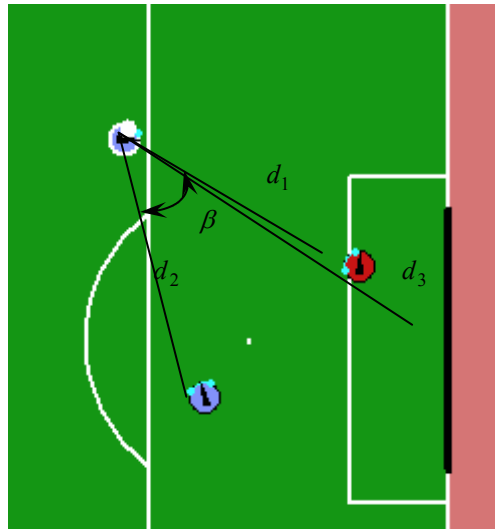

Fig. 1. State representation. d_1 - distance from teammate with ball to opponent; d_2 - distance from teammate with ball to partner; d_3 - distance from teammate with ball to the center of the goal; β - angle between d_1 and d_2; $s = (d_1, d_2, d_3, \beta)^T$; $A = \{dribble, pass\}$ -set of two possible actions.

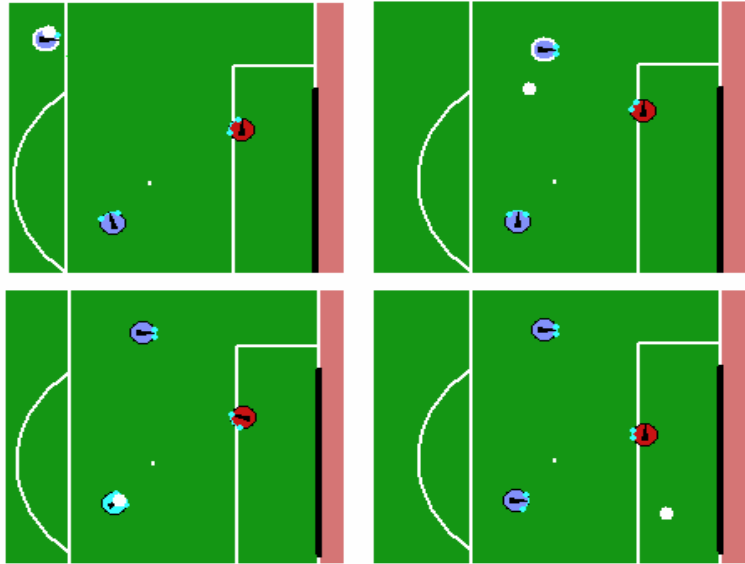   The following figure 2 shows four steps of the played episode:

Fig. 2. Episode steps

The first picture is the beginning of the episode. Agent chooses to dribble. At the picture 2 you can see, that agent has passed the ball to its partner. The following picture shows, that the second agent has received the ball, and is going to try to score the goal. As you can see, goalie is changing his defend position, trying to prevent the goal. Almost the end of the episode is shown at the picture 4. Second agent is scoring the goal, and goalie, has been outplayed, is not able to get the ball. Episode in about two simulation cycles will be ended successfully.

Conclusion

STEP2D-05 enhancements are focused on using cognitive approach pass function and scenario implementation. The cognitive approach provides more adaptive agent behavior. New pass function implementation makes pass to be more accurate. Using new scenarios enhances teamwork in attack.

In the future we intend to use STEP2D behavior and teamwork algorithms in our new 3D Soccer Simulation Team STEP3D.

References

1. Peter Stone. Layered Learning In Multi-Agent Systems. A Winner Approach To Robotic Soccer. The MIT Press, 2004

2. Stankevich L.A. Cognitive structures and agents in intellectual robot control systems. AI News #1, 2004, pp. 41-55 9 (in Russian).
3. Serebryakov S.V., Ivanov A.V., Stankevich L.A. Robotic soccer, reinforcement learning and teamwork. X International Student Olympiad on Automatic Control (Baltic Olympiad). Pages 123-128. Preprints. Saint-Petersburg, Saint-Petersburg State University of Information Technologies, Mechanics And Optics, 2004