SBC++ Simulator Team

Amin Abbaspour, Mahmood Rahmani, Bahman Radjabalipour, and Eslam Nazemi

{a-abbaspour, m-rahmani, b-radjabalipour, nazemi}@ce.sbu.ac.ir
Electrical and Computer Engineering Faculty
Shahid Beheshti University

1 Introduction

In this article the SBC++ simulator team and its overall design and scientific goals are described. Our main goal in this project is to achieve new methods of machine learning and deciding; also multi-agent behaviors such as strategy setting and cooperative learning with insistence on behavior networks. To achieve this, SBC++ uses MASM[1] model, designed by Maes, as its fundamental base.

Our team uses the CMUnited-99[4] low level code with little modification. Major modification and improvements are in the high level code.

2 Team Development

Team Leader:

Eslam Nazemi
- Iran
- Faculty member
- Attended the competition

Team Members:

Amin Abbaspour, Mahmood Rahmani, Bahman Radjabalipour
- Iran
- Undergraduate student
- Attended the competition

Web page

http://www.sbu.ac.ir/sbc/

3 REASM behavior network model

MASM(Maes Action Select Mechanism) was first introduced in 1989 by Maes[1]. in 1999, Dorer[2] introduced a revised and expanded version of it called REASM. Ten years of research on MASM resulted in REASM, which showed its abilities in RoboCup 99[3].

Here we have successfully added a new learning method to REASM. To observe this, a brief of what REASM is and how it works might be helpful. More detailed information about REASM and its principles can be found in [2].

P. Stone, T. Balch, and G. Kraetzschmar (Eds.): RoboCup 2000, LNAI 2019, pp. 397-400, 2001.

3.1 REASM in brief

There are three main layers in this model:

- Goal layer
- Competence layer
- Perception layer

Each layer contains several modules. Functions related to these modules are executed in each step. Perception layer gives information about the world to the competence layer through variable $'e'$, which shows the executability of choosing each competence module. All competence modules have a related behavior for themselves. Each time a competence module is chosen, its behavior is executed. Competence modules also have a set of parameters as follow:

γ	Activation of modules
δ	Inhibitions of modules
β	Inertia of modules
θ	Activation threshold, $\theta \in (0, a]$, where a is threshold upper limit
$\Delta\theta$	Threshold decay

Competence modules are activated and inhibited by goal modules or other competence modules. The following formulas calculate this activation and inhibition for competence module k:

$$a_{k_{g_i}}^{t}{}' = \gamma . f(i_{g_i}, r_{g_i}^{t}).ex_j \tag{1}$$

$$a_{k_{g_i}}^{t}{}'' = -\delta . f(i_{g_i}, r_{g_i}^{t}).ex_j \tag{2}$$

$$a_{k_{g_i}}^{t}{}''' = \gamma . \sigma(a_{succg_i}^{t-1}).ex_j.(1 - \tau(p_{succ}, s)) \tag{3}$$

$$a_{k_{g_i}}^{t}{}'''' = -\delta . \sigma(a_{confg_i}^{t-1}).ex_j.\tau(p_{succ}, s) \tag{4}$$

$$a_{k_{g_i}}^{t} = abs \max(a_{k_{g_i}}^{t}{}', a_{k_{g_i}}^{t}{}'', a_{k_{g_i}}^{t}{}''', a_{k_{g_i}}^{t}{}'''') \tag{5}$$

$$a_k^t = \beta a_k^{t-1} + \sum_{i=1}^{n} a_{kg_i}^t \tag{6}$$

where:

n	Number of goal modules linked to this competence module
ex_j	Expectation of goal module linked to the competence module
$\tau(p, s)$	A triangular function of perceptron p and world-state s
$f(i, r)$	A triangular function of importance i and relevance r
$\sigma(x)$	$=1/(1 + \exp(k(\mu - x)))$ Goetz[5] formula

After calculating all competence modules' activation, it is combined with executability of them using a non-decreasing function $h : R \times [0, 1] \rightarrow R, h(x, 0) = 0$. If the highest value of $h(a_k^t, e)$ lies above θ then its related competence module is chosen and its action is executed; otherwise θ is reduced as much as $\Delta\theta$. Any time the highest activation reaches this reduced threshold, θ is set to its base value again.

3.2 Toward Dynamic Activation Setting in REASM

To add learning property, we tried to use the same way cerebellum uses for its learning processes. Cerebellum compares what is supposed to be done and what is actually done; then by changing the activation of the neurons, tries to reach the ideal state.

In REASM there are several goal modules in goal layer; this gives you the opportunity of goal-oriented learning. We had to have a function considering all goals but tending to those goals which have higher importance value i. So this simple rule saying, "Do a work which tries to satisfy the most important goals" was used. Hence changing activation γ and inhibition δ of competence modules simulates learning.

γ and δ are changed according to the following rules(competence module k):

$$\gamma_k^t = \gamma_k^{t-T_k}.\psi(\frac{\sum_{k=1}^{n} i_{g_i}.\Delta f(i_{g_i}, r_{g_i}^t)}{n}) \tag{7}$$

$$\delta_k^t = \delta_k^{t-T_k}.\psi(\frac{-\sum_{k=1}^{n} i_{g_i}.\Delta f(i_{g_i}, r_{g_i}^t)}{n}) \tag{8}$$

$$\psi(x) = \frac{2}{1+e^{-\eta x}} \tag{9}$$

Here each behavior has an estimated time for its effect T_k. Actions performed by the competence layer changes the world-state, therefore output of goal functions is different between t and $t - T_k$. We take advantage of this difference. An average is calculated by adding difference of goal modules' output at time t and $t - T_k$ times goal's importance (through which important goals obtain higher effect in final average). If this average lies above zero, function ψ increases γ and decreases δ , otherwise decreases γ and increases δ.

ψ is a simple sigmoid function. The only extra parameter in ψ is η which is the learning rate and in this model is set manually. Experiences show that $\eta = 0.10$ gives better results in RoboCup domain.(this value is strongly related to your perception layer design and its output).

Some of this world-state changing between $t - T_k$ and t is caused by other agents, so it would be much effective if we take into account only those changes which are the result of this agent's activation. Overcoming these deficiencies are our future work and research topics.

It is clear that changing the activation of competence modules leads to separate activation values for each module. (previously, γ was common among all modules.)

4 Conclusion and future goals

The SBC++ simulator team is at its early stages of development and wishes to introduce new methods of learning and strategy setting.

What is said above is a part of our attempts. In future we will try to use multi-layer-input system in REASM to increase its learning capacity and adjusting network by using on-line coach.

References

[1] Maes, P. (1989). The Dynamics of Action Selection. In Proceedings of the International Joint Conference on Artificial Intelligence-'89 Morgan Kaufmann, Detroit.

[2] Dorer, K. (1999). Behavior Network for Continuous Domains Using Situation-Dependent Motivation. To appear in : Proceedings of the 16th International Joint Conference of Artificial Intelligence Morgan Kaufmann, Stockholm.

[3] Dorer, K. (1999). Extended Behavior Networks for the magmaFreiburg Soccer Team, Center for Cognition Science, Institute of Computer Science and Social Research Albert-Ludwig-University, Freiburg.

[4] Stone, P. et al (1999). The CMUnited-99 Champion Simulator Team AT&T Labs-Research,180 Park Ave.,room A273,Florham Park,NJ 07932.

[5] Goetz, Ph (1997). Attractors in Recurrent Behavior Networks, Ph.D thesis. State University of New York, Buffalo.