Seventh Conference on Simulation of Adaptive Behavior (SAB): From animals to animats 7, Edinburgh, UK, 2002

Behavior Coordination for a Mobile Visuo-Motor System in an Augmented Real-World Environment

Dimitrij Surmeli **Horst-Michael Gross**
Dept. of Neuroinformatics, Ilmenau Technical University,
PO Box 100565, D-98684 Ilmenau, Germany
E-mail: Dima.Surmeli@informatik.tu-ilmenau.de

Abstract

We utilize HUMPHRYS' W-Learning on a real robot Khepera to coordinate three behaviors in an augmented maze: first, to drive straight and fast while avoiding obstacles, second, to find a location marked by one projected color (e.g., where food can be found), and third, to escape from another color. We describe the experimental setup and compare results of the individual agents to those of a monolithic agent solving all tasks, and of the agents coordinated by different types of W-Learning. We demonstrate the feasibility of W-Learning on a real visuo-motor system and conclude by discussing why the monolith outperforms all forms of coordination investigated.

1 Introduction

Values from some Reinforcement Learning methods may serve to coordinate some agents sharing the same 'body'.

We investigate W-Learning (Humphrys, 1997) and its applicability on a real robot for an adaptive coordination of multiple goals.

The overall task may be solved by a monolithic learner at the cost of an immense state-action space and training time. Scalability to complex tasks, acceptable adaptation times and an easier design of partial reinforcement functions are the driving forces for multi-agent systems.

2 Theoretical background

2.1 W-Learning

In W-Learning (Humphrys, 1997), fig.1 right, a collection of individual agents is coordinated to each solve a task and is assumed to be a fully trained Q-Learners. It uses the Q-values of these agents to determine just how badly they want to execute their own best action in a certain state and select one.

The resulting effect may be described as follows: agent i, insists not on its own suggested action and relinquishes ownership of a state agent j, who executes an action that to agent i holds less reward, but possibly avoids catastrophic losses for j.

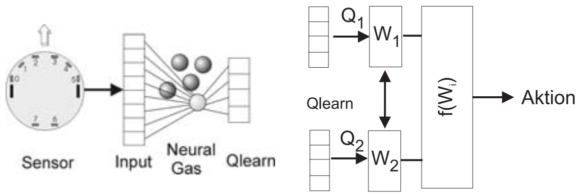

Figure 1: Structure of agents and principle of W-Learning

3 Experimental setup

We use a Khepera miniature robot equipped with 8 infrared sensors and an additional omni-directional color camera, in a special maze with wooden obstacles and projected color patches shown in fig.2 left and actions expressed as *speed* $v \in [-1, 4, 7]$; *angle* $\alpha \in [-30, 0, 30]°$.

3.1 Image preprocessing

The camera produces an omni-directional view by looking straight up at a parabolic mirror. It is transformed into a rectangular image (fig. 2 right top) and reduced to a grid of 6 by 4 regions, where averaged colors (fig. 2 right second) in HSI color space provided the inputs for the agents. The regions preserve color, some heading and distance information.

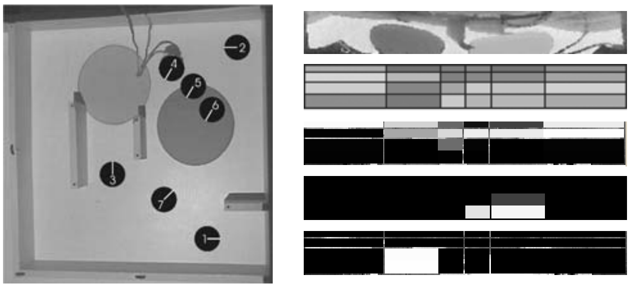

Figure 2: Top-view of the scenario with Khepera and image produced and processed as inputs to Bump (third), Thirst (fourth), and Escape (last). Robot is in Pos.4

3.2 Setup of the individual agents

All agents, except the W-Learner, share the architecture shown in figure 1 left, comprised of of a vector quantizer, such as a Neural Gas (Martinetz and Schulten, 1991) and *Sarsa* (Rummery and Niranjan, 1994) in a subsequent supervised layer trained by straight Delta-learning. *Sarsa* was used with $(\gamma = 0.8, \lambda = 0.8, \alpha = 0.5)$ with the the learning rate α held constant. Adaptation of the input clustering and the Q-values proceeded concurrently, and a Boltzmann exploration was used. Each of the expert agents Bump, Thirst and Escape, used a Neural Gas of 15 nodes, and the monolith 50. We realized the following agents:

- **Bump**, which realizes an obstacle avoidance and otherwise, tries to go as straight and fast as possible,
- **Thirst**, which will try to drive onto a blue patch
- **Escape**, who avoids driving onto the a patch,
- **W-Learner**, who consisted of and coordinated the very same agents above, complemented by a coordination by a form of W-Learning
- **Monolith**, who is charged with solving all three tasks as one overall problem and received as inputs the inputs of all the other agents in one big vector

With IR = normalized infrared readings, v =robot wheel speed, α = steering angle, lRF = lowest regions,

$$\begin{aligned} r(Bump) = &\quad \begin{cases} -8.0, \text{if } \max_{IR} > 0.9 \\ |v| * 0.25 - |(0.025 * \alpha)| & else \end{cases} \\ r(thirst) = &\quad \sum activation(lRF(blue)) \\ r(esc) = &\quad -\sum activation(lRF(red)) \quad (1) \\ r(Mono) = r(WL) = &\quad r(Bump) + r(Thirst) + r(Esc) \end{aligned}$$

4 Results

We measured the performance of the agents from 7 starting points (fig. 2 left) by calculating average total rewards (see eq.2) per step, for each trial individually, and summing those for all starting points, averaged across a number of runs.

The Khepera was set into the respective point in the orientation indicated by the white line in the dark circles in figure 2 left and was allowed to drive until collision or a maximum number of steps of 51, and the reward and trial length were averaged for 10 experiments.

The results are presented numerically for all agents in table 1 and discussed in the following paragraphs.

Negotiated WL (Humphrys, 1997) turned out to perform best among the WL-variants, while the monolith outperformed all variants of WL. It was able to use all correlated inputs as landmarks, and so coordinate the different tasks more easily by finding optimal compromises and exploiting mutual support.

Agent	cum avg reward (σ)	avg length (σ)
Mono	6.6 (4.3)	238.7 (45.9)
WL(neg)	2.3 (3.4)	193.0 (60.6)
WL(mCH)	1.9 (3.1)	254.8 (40.1)
WL(wl)	1.3 (2.5)	211.4 (27.4)
WL(wlF)	-0.7 (1.9)	134.0 (44.4)
Bump	-3.9 (4.8)	292.3 28.5)
Escape	-7.2 (6.0)	137.2 (29.6)
Thirst	-8.4 (8.0)	183.5 (28.4)
Random	-23.9 (9.1)	100.0 (32.2)

Table 1: Overview of all results. The numbers represent the mean of total reward (as in eq. 2) per step in a trial, and length of trials in steps, summed over all trials and the numbers in parentheses standard deviations. Mono = Monolith, WL = W-Learning, neg = negotiated, wl = learned WL, mCH = maximal collective happiness, wlF = learned WL with full state space

It's superior since it solves entire task optimally at the cost of solving subtasks suboptimally. In contrast, in WL the individual agents solve their tasks optimally, but optimality at the global task cannot be achieved.

The monolith uses a total reward signal, which WL never uses, to flexibly weigh the subtasks for an optimal overall task, two disconnected steps for WL.

One reason for WL's suboptimality was the fixed weighting of the agents for the entire state space. Different weight configurations supported better behavior of WL in different states, which is currently unachievable.

The monolith considers costs of transitions between subgoals. With implicit planning through $\gamma > 0$, it optimizes full sequences of coordinated actions, when WL looks at the current situation, plans only for one agent.

References

Humphrys, M. (1997). *Action Selection methods using Reinforcement Learning.* PhD thesis, University of Cambridge, Computer Laboratory.

Martinetz, T. and Schulten, K. (1991). A "Neural Gas" network learns topologies. In Kohonen, T., Mkisara, K., Simula, O., and Kangas, J., (Eds.), *Artificial Neural Networks*, pages 397–402. Elsevier.

Rummery, G. and Niranjan, M. (1994). On-line Q-learning using connectionist systems. Technical Report CUED/F-INFENG/TR 166, Engineering Department, Cambridge University, UK.