Proceedings of the 2007 IEEE/RSJ International
Conference on Intelligent Robots and Systems
San Diego, CA, USA, Oct 29 - Nov 2, 2007

Emulation and Behavior Understanding through Shared Values

Yasutake Takahashi, Teruyasu Kawamata, Minoru Asada*
Dept. of Adaptive Machine Systems,
Graduate School of Engineering, Osaka University,
*JST ERATO Asada Synergistic Intelligence Project
Yamadaoka 2-1, Suita, Osaka, 565-0871, Japan
Email: {yasutake,kawamata,asada}@er,ams.eng.osaka-u.ac.jp

Mario Negrello
Fraunhofer IAIS,
Schloss Birlinghoven, 53754 Sankt Augustin, Germany
Email: mario.negrello@ais.fraunhofer.de

Abstract— **Neurophysiology has revealed the existence of mirror neurons in brain of macaque monkeys and they shows similar activities during executing an observation of goal directed movements performed by self and other. The concept of the mirror neurons/systems[1] is very interesting and suggests that behavior acquisition and the inferring intention of other are related to each other. That is, the behavior learning modules might be used not only for behavior acquisition/execution but also for the understanding of the behavior/intention of other.**

We propose a novel method not only to learn and execute a variety of behaviors but also to understand behavior of others supposing that the observer has already acquired the utilities (state values in reinforcement learning scheme) of all kinds of behaviors the observed agent can do. The method does not need a precise world model or coordination transformation system to deal with view difference caused by different viewpoints. This paper shows that an observer can understand/recognize a behavior of other not by precise object trajectory in allocentric/egocentric coordinate space but by estimated utility transition during the observed behavior.

I. Introduction

Recent robots in real world are required to perform multiple tasks, adapt their behaviors in an encountered multi-agent environment, and learn new cooperative/competitive behaviors through the interaction with others. Reinforcement learning has been studied well for motor skill learning and robot behavior acquisition in single/multi agent environments. However, it is unrealistic to acquire various behaviors from scratch without any instruction from others in real environment because of huge exploration space and enormous learning time. Therefore, importance of instructions from others has been increasing, and in order to understand the instructions, it is necessary to infer their intentions to learn purposive behaviors.

Understanding other agent behavior is also a very important issue to realize social activities, for example, imitation learning, cooperative/competitive behavior acquisition, and so on. Recently, many researchers have studied on methods of other agent's behavior recognition/imitation system (e.g. [2], [3], [4], [5], [6], [7]). These typical approaches assume detailed knowledge of a given task, an environment, their body structure and sensor/actuator configuration, and so on based on which they can transform the observed sensory data of the others' behaviors into the global coordinate system of the environment, or an egocentric parameter space like the joint space of the others to infer their intentions. However, such an assumption seems unrealistic in the real world and brittle to the sensor/actuator noise(s) or any possible changes in the parameters. Furthermore, there are a variety of motion trajectories for a behavior achieving a certain task. The variety will be caused by constraints of body and/or environments, or experiences received so far. It is almost impossible to cover all variation of motion trajectories even for one behavior achieving one certain tasks. Additionally, almost of them focus on mimicry of the observed motions. Mimicry is to reenact someone else's action, without that action leading to reaching an immediate goal; it is to copy the behavior as in pantomime. It requires no understanding of the action beyond the motor mappings. Robotic and computational models dealing with mimicry set to understand what are motor programs, motor parsing and storage for sequences and the correspondence problem[8]. It usually does not touch the mechanisms of empathy, and goal selection. Conversely, emulation is when after observing an action, the observer jumps to conclusions and performs only those actions that will lead it to the goal, without caring about the exact methods of the demonstrator (although observed methods biases future actions). It requires sharing of values and reading of rewarded behavior. It is in effect a degenerate subset of imitation, and the one most often employed by non-human primates. Imitation is the crowning of copying, the sophisticated capability of reenacting sequence of actions to detailed levels, with the agent clearly aiming for the same objective as the demonstrator's [9]. This paper focuses on "emulation" of observed behavior.

Reinforcement learning generates not only an appropriate behavior (a map from states to actions) to achieve a given task but also an utility of the behavior, an estimated discounted sum of reward value that will be received in future while the robot is taking an optimal policy. This estimated discounted sum of reward is called "state value." This value roughly indicates closeness to a goal state of the given task, that is, if the agent is getting closer to the goal, the value becomes higher. This suggests that the observer may understand which goal the observed agent likes to achieve if the value of the corresponding task is going higher. The relationship between an agent and objects such that the agent gets close to the object or the agent

1-4244-0912-8/07/$25.00 ©2007 IEEE.

faces to a direction is much easier to understand from the observation, and therefore such qualitative information should be utilized to infer what the observed agent likes to do. The information might be far from precise ones, however, it keeps qualitative information and we can estimate well the temporal difference of the value during achieving the given task. If the observer can estimate the value of each behaviors of the other, it might be possible to recognize the other's intention, therefore the observer not only imitate the observed behavior but also cooperative/competitive behaviors according to the recognized intention.

We propose a novel method not to only learn/execute a variety of behaviors but also to understand/emulate behaviors of others. The method does not need a precise world model or an accurate coordination transformation system to cope with the problem of view dependency. We apply the method to a simple multi-agent situation where the agent has kinds of tasks such as chasing a ball, pushing a ball into a box, passing a ball to another, and so on, and the observer judges which goal the agent is now achieving from the observation with estimated values.

II. Experimental Setup

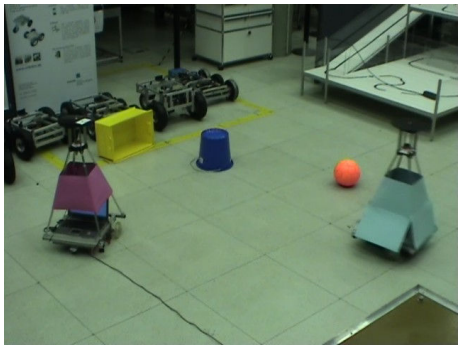


Fig. 1. Two robots and color coded toys objects

Fig.1 shows two robots and color-coded objects, e.g., an orange ball, a blue bucket, and an yellow box. The players are VolksBots [10] mobile robots endowed with omni-directional cameras on top. A simple color image processing is applied in order to detect the color-coded objects and players in real-time. The mobile platform is based on a differential wheeled vehicle and has simple basic actions, e.g. approaching an object, turning around it in clock-wise and counter-clock-wise, which were designed in advance. The two robots play by displacing objects, for example, dribbling a ball, kicking a bucket, taking a ball to a box, bringing the bucket to the other robot, and so on. While playing with objects, they watch each other and try to understand observed behaviors and emulate them, in case they see fit.

III. Outline of the Mechanisms

The reinforcement learning scheme, the state value function, and the modular learning system for various behavior acquisition/emulation are explained, here.

A. Behavior Learning Based on Reinforcement Learning

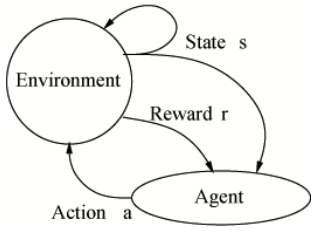


Fig. 2. Agent-environment interaction

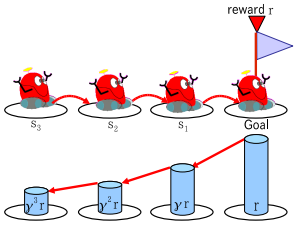


Fig. 3. Sketch of state value propagation

Fig.2 shows a basic model of reinforcement learning. An agent can discriminate a set S of distinct world states. The world is modeled as a Markov process, making stochastic transitions based on its current state and the action taken by the agent based on a policy π. The agent receives reward r_t at each step t. State value V^π, the discounted sum of the reward received over time under execution of policy π, will be calculated as follows:

$$V^\pi = \sum_{t=0}^{\infty} \gamma^t r_t \quad . \tag{1}$$

Fig.3 shows a sketch of a state value function where a robot receives a positive reward when it stays at a specified goal while zero reward else. The state value will be highest at the state where the agent receives a reward and discounted value is propagated backward to the most recent states.

The state value increases if the agent follows a good policy π. The agent updates its policy through the interaction with the environment in order to receive higher positive rewards in future. Analogously, as animals get closer to former action sequences that led to goals, they are more likely to retry it. For further details, please refer to the textbook of Sutton and Barto[11] or a survey of robot learning[12].

B. Modular Learning System

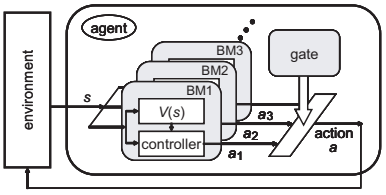


Fig. 4. Modular learning system

In order to observe/learn/execute a number of behaviors simultaneously, we adopt a modular learning system. Jacobs

and Jordan [13] proposed a mixture of experts, in which a set of the expert modules learn and are weighted by the gating system to produce the output. Fig.4 shows a sketch of such a modular learning system. We prepare a number of behavior modules (BM in the figure) each of which adopts the behavior learning method described in III-A. The module is assigned to one goal-oriented behavior and estimates one state value V^{π}. A module receives a positive reward when it accomplishes the assigned behavior and zero reward else. The behavior module has a controller that generates predictions of next state values, selecting the action with the maximum value. The gating module will then select one output from the inputs of the different behavior modules according to the player's intention.

C. Behavior Categorization based on Estimated Values

Each behavior module can estimate a state value of observed behavior at an arbitrary time t to accomplish the specified task. An observer watches a demonstrator's behavior and maps the sensory information from an observer viewpoint to a demonstrator's one with a simple mapping of state variables. Fig.5 shows a simple example of this transformation. It detects color-coded objects on the omni-directional image, finds the demonstrator, and shifts the axes so that the position of the demonstrator comes to center of the image. Then it roughly estimates the sensory information in the egocentric coordinate and the state of the demonstrator. Every behavior module estimates a sequence of its state value from the estimated state of the observed demonstrator and the system selects modules which values are increasing.

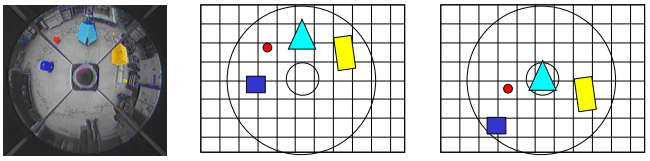


Fig. 5. Simple view transformation from self's to other's. left : a captured image the of observer, Center : object detection (center is self), Right : moving the position of demonstrator to center

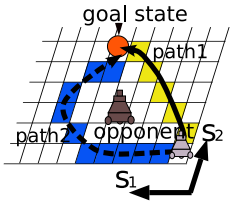


Fig. 6. Sketch of different behaviors in a grid world

Fig.6 shows an example task of navigation in a grid world. There is a goal state at the top center of the world. An agent can move one of the neighboring square in the grids every step. It receives a positive reward only when it stays at the goal state while zero else. There are various optimal/suboptimal policies for this task as shown in Fig.6. If one tries to match the action that the agent took and the one based on a certain policy in order to infer the agent's intention, he or she has to maintain various optimal policies and evaluate all of them in the worst case.

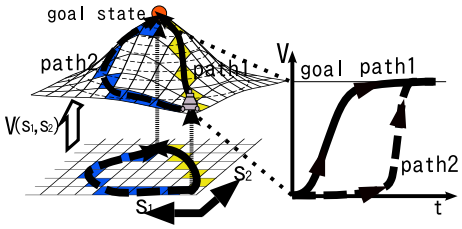


Fig. 7. Inferring intention by the change of state value

On the other hand, if the agent follows an appropriate policy, the value is going up even if it is not exactly the optimal one. Likewise, in emulation one is not committed with the optimal policy, as the behaviors are the ones available in the portfolio of the agent, which are not necessarily the optimal ones, but the ones that the agent knows to lead to the goal (Fig.7).

This indicates a possibility of robust intention recognition even if several appropriate policies can exist for the current task. An agent tends to acquire various policies depending on the experience during learning. The observer cannot practically estimate the performer's experience beforehand, therefore, it needs a robust intention recognition method, which is provided by the estimation of state values.

The method has also a possibility of robustness against calibration error of view transformation self's to other's. The relationship between an demonstrator and objects such that the demonstrator gets close to the object or the agent faces to a direction is much easier to understand from the observation, and therefore such qualitative information should be utilized to infer what the observed agent intends. The information might be far from precise, however, it keeps qualitative information so it can estimate well the temporal difference of the value.

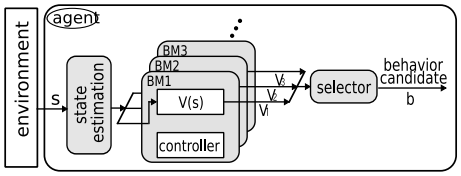


Fig. 8. Behavior inference diagram

While an observer watches a demonstrator's behavior, it uses the same behavior modules for categorization of observed behavior as shown in Fig.8. Each behavior module estimates the state value based on the estimated state of the observed demonstrator and sends it to the selector. The selector watches the sequence of the state values and selects a set of possible behavior modules of which state values are going up as a set of behaviors the demonstrator is currently taking. As mentioned before, if the state value goes up during a behavior, it means that the module is valid for explaining

the behavior. The observed behavior is categorized into a set of behavior whose modules' values are increasing.

Here we define reliability g that indicates how much the observed behavior would be reasonable to be categorized into a behavior

$$g = \begin{cases} g + \beta & \text{if } V_t - V_{t-1} > 0 \text{ and } g < 1 \\ g & \text{if } V_t - V_{t-1} = 0 \\ g - \beta & \text{if } V_t - V_{t-1} < 0 \text{ and } g > 0 \end{cases},$$

where β is an update parameter, and 0.1 in this paper. This equation indicates that the reliability g will become large if the estimated utility rises up and it will become low when the estimated utility goes down. We put another condition in order to keep g value from 0 to 1.

IV. Experimental Results

In this section, we describe experimental results of behavior generation based on my value, categorization of observed behavior, and emulation of observed behavior, one by one.

A. My Action, My Value, My Behavior

We let one player learn a number of behaviors shown in Table I at the beginning. In the environment, there are two players, one with a magenta marker and the other with a cyan marker, along with a yellow box, and a red ball. There is no blue bucket at this moment. The player has learned each behavior with a little human support and acquired experiences enough to cover all of the explorable state space. After the learning phase, the player can take an appropriate action in every state based on value of the action, then it produces a behavior. As mentioned, if it

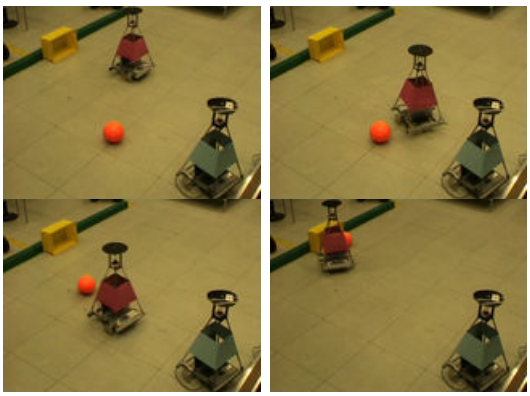


Fig. 9. A behavior of pushing a ball into an yellow box

takes an optimal policy, the value of the behavior keeps increasing until it reaches the goal state of the behavior while the other values pace up and down. Fig.9 shows one scene that a magenta player shows a behavior of pushing a ball into a yellow box. Fig.10 shows a sequence of values during the scene. The orange line indicates the value of the behavior. It shows increasing tendency during the behavior. The behavior is composed of behaviors of approaching a ball and approaching a yellow box so that the red line goes up in the earlier stage and the yellow line goes up in the later stage.

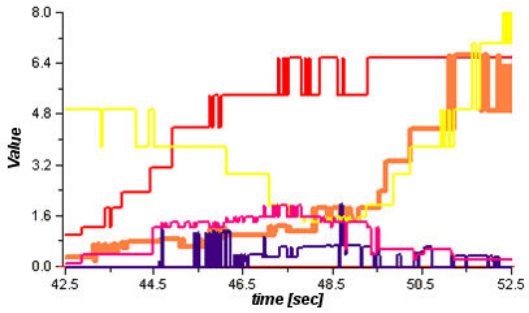


Fig. 10. Sequence of values during a behavior of pushing a ball into an yellow box, red line : approaching a ball, yellow line : approaching an yellow box, orange line : pushing a ball to yellow box, light magenta : approaching another player, dark magenta : pushing a ball to another player

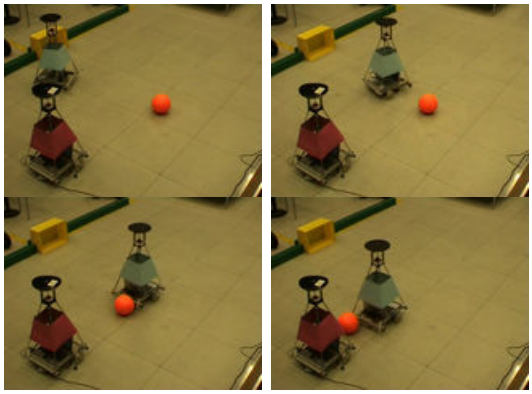


Fig. 11. Magenta player observes an demonstrator's behavior of pushing a ball to the magenta player

1) Categorization of Observed Behavior: When a player watches a behavior of the other, it categorizes the observed behavior based on repertoire of its own behaviors. Fig.11 shows one scene in which the magenta player observes an demonstrator's behavior of pushing a ball to the magenta player. Figs.12(a) and (b) show sequences of estimated values and reliabilities of the behaviors, respectively, as the demonstrator pushes a ball to the player. The dark magenta line indicates the behavior and keeps tendency of increasing value during the behavior in this figures. This behavior is composed of behaviors of approaching a ball and approaching to another player again, then, the red line goes up at the earlier stage and the light purple line goes up at the later stage in Fig.12(a). All reliabilities start from 0.5 and increase if the value goes up and decrease else. Even when the value stays low, if it is increasing with small value, the reliability of the behavior increases rapidly. The reliability of the behavior of pushing a ball into another player, dark magenta line, reaches 1.0 at middle stage of the observed behavior.

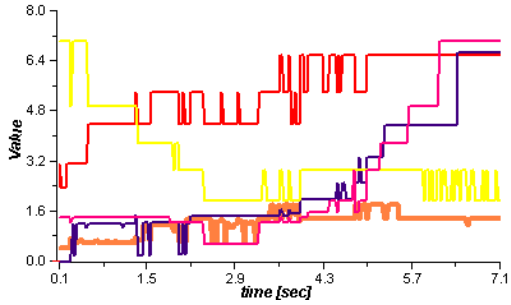
B. Emulation of Observed Behavior

Here, we introduce a new object in the environment, a blue bucket. Because a player does not have any experience with a blue bucket, there is no associated behavior with the object

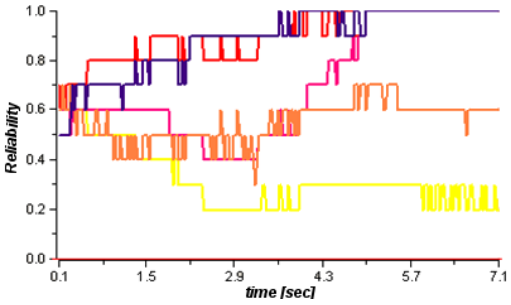
TABLE I

LIST OF BEHAVIORS LEARNED BY SELF AND STATE VARIABLES FOR EACH BEHAVIOR

Behavior	State variables
Approaching a ball	distance to the ball
Approaching an yellow box	distance to the box position
Approaching another player	distances to the ball, the player, and angle between them
Pushing a ball to an yellow box	distances to the ball, the box, and angle between them
Pushing a ball to another player	distances to the ball, the player, and angle between them

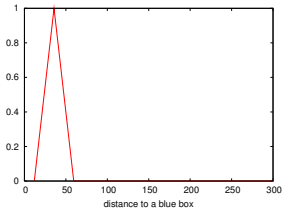


(a) Estimated Values

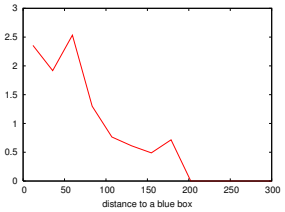


(b) Reliabilities

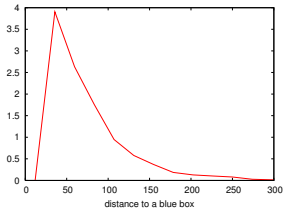
Fig. 12. Sequence of estimated values and reliabilities during a behavior of pushing a ball to the magenta player, red line : approaching a ball, yellow line : approaching an yellow box, orange line : pushing a ball to yellow box, light magenta : approaching another player, dark magenta : pushing a ball to another player



(a) Reward reading



(b) Value Back-propagation



(c) Correction of Value

Fig. 13. Development of value through observation of other

in the player's repertoire. However, after one player shows some interaction with the blue bucket, the other should be able to recognize the goal, and later effectively emulate it.

This procedure will be as follows:

1) A player watches a behavior of the demonstrator,
2) transforms the sensory information in observer's coordinate to the one in demonstrator's coordinate,
3) reads demonstrator's reward,
4) back-propagates the reward as values to sequence of states estimated during the observation,
5) emulates the observed behavior and updates values by exploration through trial and error or mental rehearsal.

First of all, a behavior of approaching a blue bucket is shown to a player as a simple case. The player executes from 1 to 3 of the list above one by one. Fig.13(a) shows estimated state value function after it reads the demonstrator's reward. The x and y axes indicate distance to the bucket and state value, respectively. It has only one peek at a state where it will get a positive reward and other state's values are zero. It back-propagates the reward to the sequenced states at 4th procedure of the list above, then, estimates state values based on the state sequence of observed behavior (see Fig.13(b)).

After the player estimates the value of the observed behavior, it tries by itself, thereby achieving emulation. The estimated state value function is a good reference to imitate the observed behavior while the estimated state values might be inconsistent because of difference of their body dynamics or error of estimated sensory information during observation. In order to correct values of the behavior, a state transition model or self-experience of the behavior is necessary. A state transition model can be acquired through some exploration. If it has the model in advance through the experience of playing with the object, it is able to use it without further exploration. Fig.13(c) shows the corrected state value function after some exploration.

Next, a behavior of pushing blue bucket to the yellow box are shown to a player. It follows the same procedure of the lists above, in this manner acquiring a new behavior through the observation. Fig.14 shows a sequence of observed behavior and Fig.15 shows sequences of estimated values and reliabilities of behaviors during the observation. The blue and green lines indicate the behavior of approaching a blue bucket and the one of pushing it to the yellow box. It shows that it successfully recognizes the observed behavior.

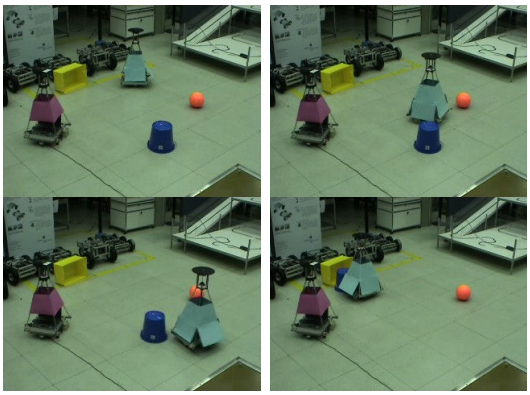


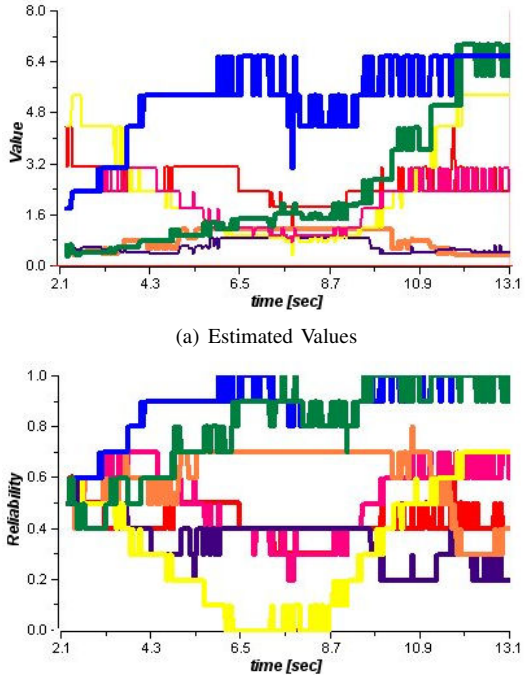
Fig. 14. The magenta player observes an demonstrator's behavior of pushing a blue bucket to an yellow box

V. Conclusion

Above, values are defined as categories of behaviors, which are defined by the achieved goals. The observer uses its own reward functions to understand what the other will do. Preliminary investigations in a similar context have been done by Takahashi at el. [14] and they showed much better robustness of behavior recognition than a typical method. Unknown behaviors are also categorized and understood in term of one's own reward functions. Moreover, the agent chooses the next action at every time step, and that action is chosen according to experience of rewards that were back-propagated through the states with the reinforcement learning algorithm. Therefore, recognition of context leads always to selection of the action that was most likely to provide reward (adequate policy, not necessarily optimal). This shows the choice of action as a process determined by previous experience. Also in the case of novel goals, the robot performing the action, uses his own action set. This is proposed as a simple model of emulation and action understanding.

Acknowledgements

The authors would like to thank Thomas Christaller and Frank Pasemann at the Fraunhofer Institute for making our cooperation possible. This work was partially funded by the German research foundation priority program DFG-SPP 1125 "cooperating teams of mobile robots in dynamic environments"



(a) Estimated Values

(b) Reliabilities

Fig. 15. Sequence of estimated values and reliabilities during a behavior of pushing a blue bucket to the yellow box, red line : approaching a ball, yellow line : approaching an yellow box, orange line : pushing a ball to yellow box, light magenta : approaching another player, dark magenta : pushing a ball to another player, blue line : approaching a blue buckets, green line : pushing a bucket to Yellow box

References

[1] E. Oztop, M. Kawato, and M. Arbib, "Mirror neurons and imitation: a computationally guided review," *Neural Networks*, vol. 19, no. 3, pp. 254–271, Apr 2006.

[2] S. Schaal, A. Ijspeert, and A. Billard, "Computational approaches to motor learning by imitation," pp. 199–218, 2004.

[3] D. K., S. N., W. D.M., and K. M., "Selecting optimal behaviors based on contexts," in *International Symposium on Emergent Mechanisms of Communication*, 2003, pp. 19–23.

[4] B. Price and C. Boutilier, "Accelerating reinforcement learning through implicit imitatione," *Journal of Articial Intelligence Research*, 2003.

[5] T. Inamura, Y. Nakamura, and I. Toshima, "Embodied symbol emergence based on mimesis theory," *International Journal of Robotics Research*, vol. 23, no. 4, pp. 363–377, 2004.

[6] A. Billard, Y. Epars, S. Calinon, S. Schaal, and G. Cheng, "Discovering optimal imitation strategies," *Robotics and Autonomous Systems*, vol. 47, no. 2-3, pp. 68–77, 2004.

[7] M. Ito and J. Tani, "On-line imitative interaction with a humanoid robot using a dynamic neural network model of a mirror system," *Adaptive Behavior*, vol. 12, no. 2, pp. 93–115, 2004.

[8] K. Dautenhahn and C. Nehaniv, *Imitation in Animals and Artifacts*. MIT Press, 2002, ch. The Agent-Based Perspective on Imitation.

[9] S. Hurley, *Perspectives on Imitation*, 2004, ch. The Shared Circuits Model.

[10] T. Wisspeintner, "The volksbot concept - rapid prototyping for real-life applications in mobile robotics," *Information Technology*, vol. 47, no. 5, 2005.

[11] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998. [Online]. Available: citeseer.ist.psu.edu/sutton98reinforcement.html

[12] J. H. Connell and S. Mahadevan, *ROBOT LEARNING*. Kluwer Academic Publishers, 1993.

[13] R. Jacobs, M. Jordan, N. S, and G. Hinton, "Adaptive mixture of local experts," *Neural Computation*, vol. 3, pp. 79–87, 1991.

[14] Y. Takahashi, T. Kawamata, and M. Asada, "Learning utility for behavior acquisition and intention inference of other agent," in *Proceedings of the 2006 IEEE/RSJ IROS 2006 Workshop on Multi-objective Robotics*, Oct 2006, pp. 25–31.