Cooperative Behavior Acquisition for Mobile Robots in Dynamically Changing Real Worlds via Vision-Based Reinforcement Learning and Development ⋆

Minoru Asada, Eiji Uchibe, and Koh Hosoda [1]

Graduate School of Eng., Dept. of Adaptive Machine Systems
Osaka University, Suita, Osaka 565-0871, Japan

Abstract

In this paper, we first discuss the meaning of physical embodiment and the complexity of the environment in the context of multiagent learning. We then propose a vision-based reinforcement learning method that acquires cooperative behaviors in a dynamic environment. We use the robot soccer game initiated by RoboCup [6] to illustrate the effectiveness of our method. Each agent works with other team members to achieve a common goal against opponents. Our method estimates the relationships between a learner's behaviors and those of other agents in the environment through interactions (observations and actions) using a technique from system identification. In order to identify the model of each agent, Akaike's Information Criterion is applied to the results of Canonical Variate Analysis to clarify the relationship between the observed data in terms of actions and future observations. Next, reinforcement learning based on the estimated state vectors is performed to obtain the optimal behavior policy. The proposed method is applied to a soccer playing situation. The method successfully models a rolling ball and other moving agents and acquires the learner's behaviors. Computer simulations and real experiments are shown and a discussion is given.

⋆ Partially supported by Japanese Society for Promotion of Science Project "Cooperative Distributed Vision for Dynamic Three Dimensional Scene Understanding." Project ID: JSPS-RFTF96P00501

[1] E-mail:asada@ams.eng.osaka-u.ac.jp

1 Introduction

Building a robot with the capability of learning and carrying out a task using visual information has been acknowledged as one of the major challenges facing vision, robotics, and AI. Here, vision and action are tightly coupled and inseparable [6]. Human beings cannot see without eye movements, which suggests that actions significantly affect visual processes and vice versa. There have been several attempts to build an autonomous agent based on a tight coupling between vision (and/or other modalities) and actions [6,6,6]. The authors of these experiments contend that vision is not an isolated process but a component of a complicated system (physical agent) which interacts with its environment [6,6,6,6]. This is a view quite different from the conventional computer vision approaches which have paid little attention to physical bodies. A typical example is so-called "segmentation" which has been one of the most difficult problems in computer vision because of its historical lack of answers to questions about the significance and usefulness of the segmentation results. These issues would be difficult to evaluate without a clear purpose. That is, the issues are *task oriented*. However, they are not straightforward design issues as determined by some special purpose application. Rather, they concern the nature of the physical agents that are capable of sensing and acting. That is, segmentation may correspond to a process of building the agent's internal representation based on its interactions with its environment.

From the standpoint of control theory, the internal representation can be regarded as a set of state vectors because it includes the necessary and sufficient information to accomplish a given task. It can also be viewed as a state space representation in robot learning for the same reason. This is especially true in reinforcement learning which has recently been receiving increased attention as a method that requires little or no a priori knowledge and that has a higher capability of reactive and adaptive behaviors [6].

There have been few works published on reinforcement learning with vision and action. Whitehead and Ballard proposed an active vision system [6] involving a computer simulation. Asada et al. [6] applied vision-based reinforcement learning to a real robot task. In these methods, the environment does not include independently moving agents; therefore, the complexity of the environment is not as great as one including other agents. In the case of a multi-robot environment, the internal representation would be more complex in order to accomplish the given tasks [6]. The main reason for this is that the learning robot cannot share another robot's perception completely; thus, it cannot discriminate among situations which other robots can, and vice versa. Therefore, the learner cannot predict the other robot's behavior correctly, even if its policy is fixed, unless explicit communication is available. It is important for the learner to understand the strategies of the other robots and to predict

their movements in advance in order to learn successful behaviors.

Littman [6] proposed the framework of Markov Games in which learning robots try to learn a mixed strategy optimal against the worst possible opponent in a zero-sum 2-player game in a grid world. He assumed that the opponent's goal is given to the learner. Lin [6] compared window-Q based on both the current sensation and the N most recent sensations and actions with recurrent-Q based on a recurrent network. He showed the latter to be superior to the former because a recurrent network can cope with historical features appropriately. However, it is still difficult to determine the number of neurons and the structures of network in advance. Furthermore, these methods utilize global information.

Robotic soccer is a good domain for studying multi-agent problems [6]. Stone and Veloso proposed a layered learning method consisting of two levels of learned behaviors [6]. The lower is for basic skills (e.g., interception of a moving ball) and the higher is for making decisions (e.g., whether or not to make a pass) based on a decision tree. Uchibe et al. proposed a method of modular reinforcement learning which coordinates multiple behaviors and takes account of tradeoffs between learning time and performance [6]. Since these methods use the current sensor outputs as states, they cannot cope with temporal changes in an object.

As described above, these existing learning methods in multiagent environments need a well-defined state space (well-defined state vectors) for the learning to converge. Therefore, a modeling architecture is required to make reinforcement learning applicable.

In this paper, we first discuss the meaning of physical embodiment and the complexity of the environment in the context of multiagent learning. We then propose a vision-based reinforcement learning method for acquiring cooperative behaviors in dynamic environments. This method finds the relationships between the behaviors of the learner and the other agents through interactions (observations and actions) using the method of system identification. In order to construct the local predictive model of other agents, we apply Akaike's Information Criterion (AIC) [6] to the results of Canonical Variate Analysis (CVA) [6], which is widely used in the field of system identification. The local predictive model is based on the observation and action of the learner (observer). We apply the proposed method to a simple soccer-like game. The task of the robot is to shoot a ball that is passed back from the other robot (passer). Also, the passer learns to pass a ball towards the shooter. Because the environment consists of a stationary agent (goal), a passive agent (ball) and an active agent (the opponent), the learner needs to construct the appropriate models for all of these agents. After the learning robot identifies the model, reinforcement learning is applied in order to acquire purposive behaviors. The

proposed method can cope with a moving ball because the state vector is estimated in a way that allows the learning system to predict its motion in the image. Simulation results and real experiments are shown and a discussion is presented.

2 Physical Embodiment

2.1 Meaning of Physical Embodiment

The ultimate goal of our research is to design physical agents (robots) which support the emergence of complex behaviors through their interactions. In order for intelligent behavior to occur, physical bodies must help to bring the system into *meaningful* interaction with the physical environment. That interaction is complex and uncertain, and has an automatically consistent set of natural constraints. This facilitates the correct agent design, learning from the environment, and rich meaningful agent interaction [6]. The meaning of "having a physical body" can be summarized as follows:

(i) Sensing and acting capabilities are not separable, but tightly coupled.
(ii) In order to accomplish the given tasks, the sensor and actuator spaces should be abstracted under resource-bounded conditions (memory, processing power, controller, etc.).
(iii) The abstraction depends on both how the agent is embodied including its internal workings and its experiences (interactions with its environment).
(iv) The consequences of the abstraction are the agent-based subjective representation of the environment. Its evaluation can be conducted using the consequences of behaviors.
(v) In the real world, both inter-agent and agent-environment interactions are asynchronous, parallel, and arbitrarily complex. The agent should cope with the increasing complexity of the environment in order to accomplish the given task at hand.

Even though we should emphasize the importance of *physical embodiment*, it is necessary to show that the system performs well coping with new issues in a concrete task domain. In other words, we need a standard problem that exposes various aspects of intelligent behavior in real environments.

As a task example, we adopt the domain of soccer playing robots, RoboCup, which is an attempt to foster AI and robotics research by providing a standard problem where a wide range of technologies can be integrated and examined [6].

We have proposed two methods related to 2 and 3 for the state and action space construction for reinforcement learning. One is based on an off-line learning method [6] and the other an on-line one [6]. Related to 4 and 5, we try to explain the environmental complexity based on the relationships between observations and self motions in the next subsection.

2.2 Environmental Complexity

Since each animal species can be regarded as having its own kind of intelligence, a difference of intelligence seems to depend on the kind of agent (capabilities in sensing, acting, and cognition), the kind of environment and the relationship between them. If agents have the same bodies, differences in intelligence can occur in the complexity of interactions with their environments. In the case of our soccer playing robot with vision, the complexity of interactions may change because of the presence of other agents in the field such as teammates, opponents, judges, and so on. In the following, we present our view regarding the levels of complexity of interactions, especially from a viewpoint that takes into accounts the existence of other agents.

(i) **Body of its own and static environment:** The body of its own or static environment can be defined in a way that notes the changes in the image plane that can be directly correlated with the self-induced motor commands (e.g., looking at your hand showing voluntary motion, as does changing your gaze to observe the environment). Theoretically, discrimination between "body of its own" and "static environment" is a difficult problem because the definition of "static" is relative and depends on the selection of the reference (the base coordinate system) which also depends on the context of the given task. Usually, we suppose the orientation of gravity can provide the ground coordinate system.

(ii) **Passive agents:** As a result of actions of the self or other agents, passive agents can be moving or still. A ball is a typical example. As long as they are stationary, they can be categorized into the static environment. But no simple correlation of motor commands with its body or the static environment can be expected when they are in motion.

(iii) **Other active agents:** Active agents do not have a simple and straightforward relationship with self motions. In the early stage, they are treated as noise or disturbance because they lack direct visual correlation with the self-induced motor commands. Later, they can be found from more complicated and higher order correlations (coordination, competition, and others). The complexity is drastically increased.

According to the complexity of the environment, the internal representation of the robot should be more sophisticated and complex in order to generate

various intelligent behaviors. Using real robot experiments, we show one such representation coping with the complexity of agent-environment interactions.

3 Our Approach

3.1 Architecture

To make the learning successful, it is necessary for the learning agent to estimate appropriate state vectors. However, the agent cannot obtain all the information necessary for this estimation owing to its limited sensing capability. What the learning agent can do is to collect all the observed data and to find the relationship between the observed agents and the learner's behaviors. This may identify a suitable behavior, although it might not be optimal. In the following, we use a method of system identification with the previously observed data and the motor commands as the input and future observation as the output of the system.

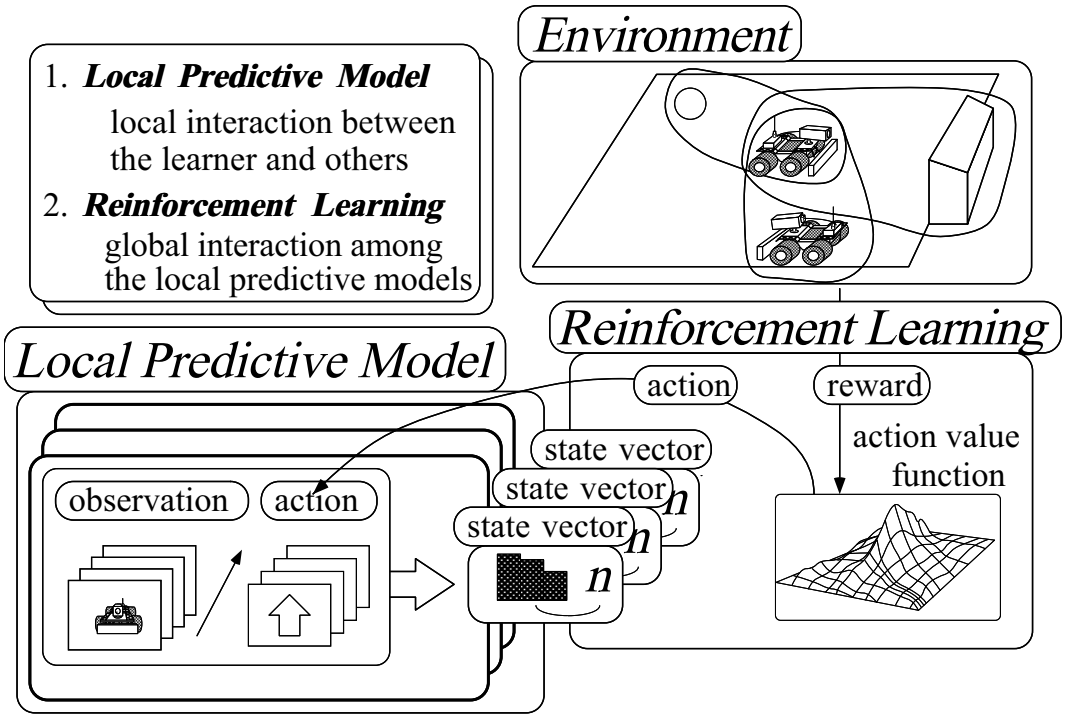


Fig. 1. Proposed architecture

Figure 1 shows the learning architecture for each robot. First, the learning robot constructs local predictive models from the sequences of not only sensor outputs but also its own action. It needs the state vectors by which it can predict future states in dynamic environments. Next, it learns cooperative behaviors based on state vectors estimated from the local predictive models. The

reason for two-phase learning is as follows: Strictly speaking, all the robots do in fact interact with each other; therefore, the learning robot should construct the local predictive model taking all these interactions into account. However, it is impossible to collect adequate input-output sequences and to estimate the proper model because the dimension of the state vector increases drastically. Therefore, the learning (observing) robot first estimates the local predictive model for each of the other (observed) robots and objects in a separate environment, and the higher interactions among robots are obtained through a post reinforcement learning process.

3.2 Learning schema

In order to acquire cooperative behaviors in multi-robot environments, we make a schedule for reinforcement learning. The actual learning methods can be categorized into three approaches:

(i) **Learning a policy in a real environment:** except for easy tasks in simple environments, it is difficult to implement.
(ii) **Learning a policy in computer simulation and transferring it into a real environment:** since there is still a gap between the simulation environment and the real one, we need some modifications in the real experiment.
(iii) **Combination of computer simulation and real experiments:** based on the simulation results, learning in a real environment is scheduled.

We adopt the third approach and make a plan for learning (see Figure 2). The robot constructs the local predictive models, and then it learns the behaviors in a real environment based on the simulation results to improve performance. This also accelerates the whole learning process.

4 Local predictive models in multi agent environment

4.1 An overview of local predictive models

Figure 3 shows an overview of the local predictive model. The local predictive model estimates the state vector $\boldsymbol{\mu}$ from the sequences of input $\boldsymbol{u}$ and output $\boldsymbol{y}$. If the model cannot obtain adequate precision, it increases the historical length l to improve the model. Next, it reduces the order of the estimated state vector n based on the information criterion to make the size of the state space tractable. Reinforcement learning receives the state vectors from the

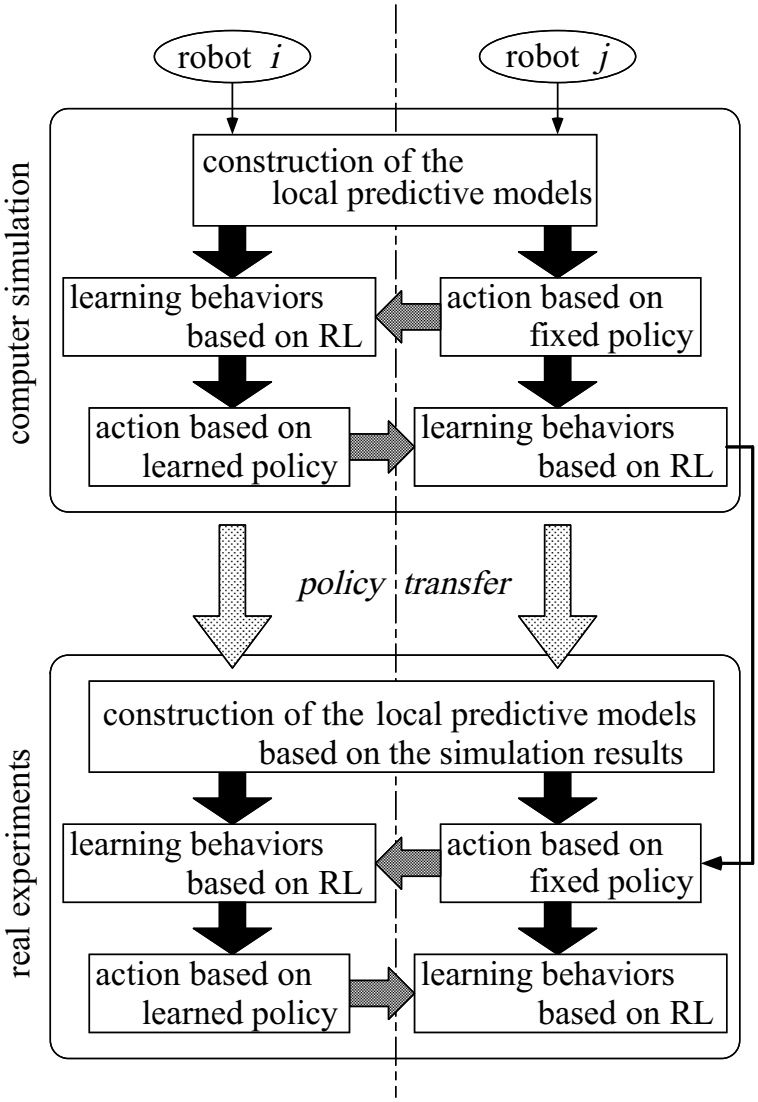


Fig. 2. Learning schedule for multi-robot environments

local predictive models, and learns the relationships among them.

4.2 Canonical Variate Analysis (CVA)

A number of algorithms to identify multi-input multi-output (MIMO) combined deterministic-stochastic systems have been proposed. Among them, Larimore's Canonical Variate Analysis (CVA) [6] is a typical one; it uses canonical correlation analysis to construct a state estimator.

Let $\boldsymbol{u}(t) \in \Re^m$ and $\boldsymbol{y}(t) \in \Re^q$ be the input and output generated by the unknown system

$$\boldsymbol{x}(t+1) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{w}(t),$$

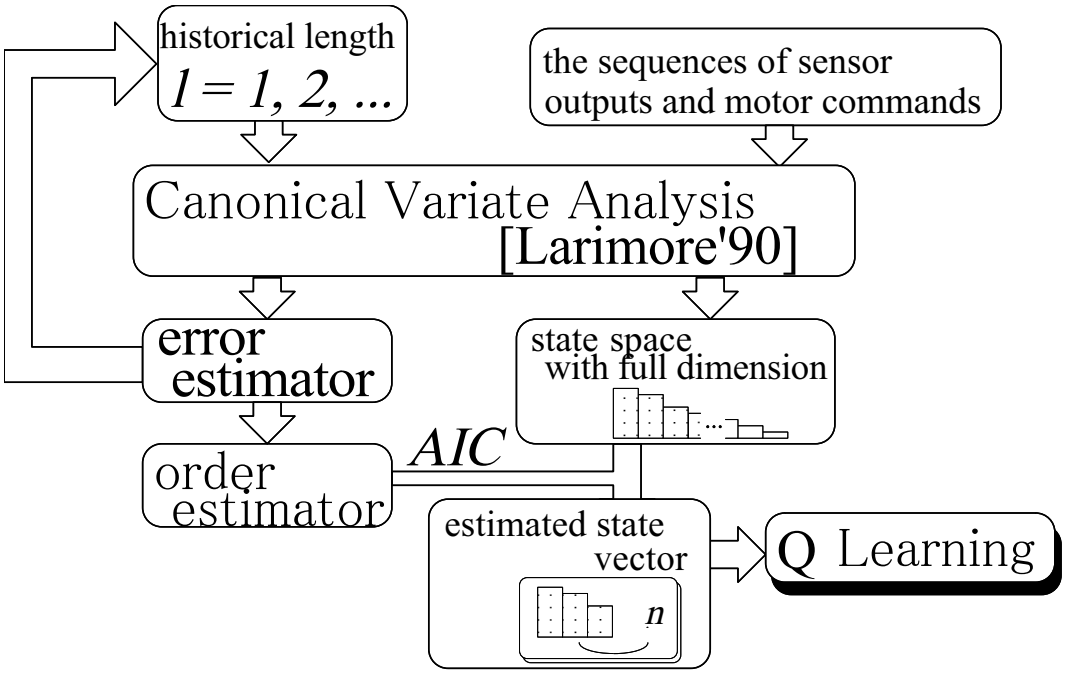


Fig. 3. Local predictive model

$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t) + \boldsymbol{v}(t), \tag{1}$$

with

$$E\left\{\begin{bmatrix}\boldsymbol{w}(t)\\ \boldsymbol{v}(t)\end{bmatrix}\begin{bmatrix}\boldsymbol{w}^T(\tau)\ \boldsymbol{v}^T(\tau)\end{bmatrix}\right\} = \begin{bmatrix}\boldsymbol{Q} & \boldsymbol{S}\\ \boldsymbol{S}^T & \boldsymbol{R}\end{bmatrix}\delta_{t\tau},$$

and $\boldsymbol{A}$, $\boldsymbol{Q} \in \Re^{n\times n}$, $\boldsymbol{B} \in \Re^{n\times m}$, $\boldsymbol{C} \in \Re^{q\times n}$, $\boldsymbol{D} \in \Re^{q\times m}$, $\boldsymbol{S} \in \Re^{n\times q}$, $\boldsymbol{R} \in \Re^{q\times q}$. $E\{\cdot\}$ denotes the expected value operator and $\delta_{t\tau}$ the Kronecker delta. $\boldsymbol{v}(t) \in \Re^q$ and $\boldsymbol{w}(t) \in \Re^n$ are unobserved, Gaussian-distributed, zero-mean, white noise vector sequences. CVA uses a new vector $\boldsymbol{\mu}$ which is a linear combination of the previous input-output sequences since it is difficult to determine the dimension of $\boldsymbol{x}$. Eq. (1) is transformed as follows:

$$\begin{bmatrix}\boldsymbol{\mu}(t+1)\\ \boldsymbol{y}(t)\end{bmatrix} = \Theta\begin{bmatrix}\boldsymbol{\mu}(t)\\ \boldsymbol{u}(t)\end{bmatrix} + \begin{bmatrix}\boldsymbol{T}^{-1}\boldsymbol{w}(t)\\ \boldsymbol{v}(t),\end{bmatrix}, \tag{2}$$

where

$$\hat{\Theta} = \begin{bmatrix}\boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T} & \boldsymbol{T}^{-1}\boldsymbol{B}\\ \boldsymbol{C}\boldsymbol{T} & \boldsymbol{D}\end{bmatrix}, \tag{3}$$

and $\boldsymbol{x}(t) = \boldsymbol{T}\boldsymbol{\mu}(t)$. We follow the simple explanation of the CVA method.

(i) For $\{\boldsymbol{u}(t), \boldsymbol{y}(t)\}$, $t = 1, \cdots N$, construct new vectors

$$\boldsymbol{p}(t) = \begin{bmatrix} \boldsymbol{u}(t-1) \\ \vdots \\ \boldsymbol{u}(t-l) \\ \boldsymbol{y}(t-1) \\ \vdots \\ \boldsymbol{y}(t-l) \end{bmatrix}, \quad \boldsymbol{f}(t) = \begin{bmatrix} \boldsymbol{y}(t) \\ \boldsymbol{y}(t+1) \\ \vdots \\ \boldsymbol{y}(t+k-1) \end{bmatrix},$$

(ii) Compute estimated covariance matrices $\hat{\Sigma}_{pp}$, $\hat{\Sigma}_{pf}$ and $\hat{\Sigma}_{ff}$, where $\hat{\Sigma}_{pp}$ and $\hat{\Sigma}_{ff}$ are regular matrices.

(iii) Compute singular value decomposition

$$\begin{aligned} \hat{\Sigma}_{pp}^{-1/2} \hat{\Sigma}_{pf} \hat{\Sigma}_{ff}^{-1/2} &= \boldsymbol{U}_{aux} \boldsymbol{S}_{aux} \boldsymbol{V}_{aux}^T, \\ \boldsymbol{U}_{aux} \boldsymbol{U}_{aux}^T = \boldsymbol{I}_{l(m+q)}&, \ \boldsymbol{V}_{aux} \boldsymbol{V}_{aux}^T = \boldsymbol{I}_{kq}, \end{aligned} \tag{4}$$

and $\boldsymbol{U}$ is defined as:

$$\boldsymbol{U} := \boldsymbol{U}_{aux}^T \hat{\Sigma}_{pp}^{-1/2}.$$

(iv) The n dimensional new vector $\boldsymbol{\mu}(t)$ is defined as:

$$\boldsymbol{\mu}(t) = [\boldsymbol{I}_n \ \ 0] \boldsymbol{U} \boldsymbol{p}(t), \tag{5}$$

(v) Estimate the parameter matrix Θ applying least- squares method to Eq. (2).

As mentioned above, the learning (observing) agent applies the CVA method to each (observed) agent separately because of an excessively high dimension of the whole state space. Hereafter, we denote the estimated state vector as $\boldsymbol{x}$ instead of $\boldsymbol{\mu}$ for the sake of the reader's understanding.

4.3 Determine the dimension of other agent

It is important to decide the dimensionality n of the state vector $\boldsymbol{x}$ and lag operator l for it provides necessary historical information for determining the size of the state vector when we apply CVA to the classification of agents. Although the estimation is improved if l becomes larger and larger, much more historical information is necessary. However, it is desirable that l be as small as possible with respect to memory size. Complex behaviors of other agents can be captured by choosing an order n that is high enough.

In order to determine n, we apply Akaike's Information Criterion (AIC) which is widely used in the field of time series analysis. AIC is a method for balancing precision and computation (the number of parameters). Let the prediction error be $\boldsymbol{\varepsilon}$ and covariance matrix of error be

$$\hat{\boldsymbol{R}} = \frac{1}{N-k-l+1} \sum_{t=l+1}^{N-k+1} \boldsymbol{\varepsilon}(t)\boldsymbol{\varepsilon}^T(t).$$

Then $AIC(n)$ is calculated by

$$AIC(n) = (N-k-l+1)\log|\hat{\boldsymbol{R}}| + 2\lambda(n), \tag{6}$$

where λ is the number of the parameters. The optimal dimension n^* is defined as

$$n^* = \arg\min AIC(n).$$

While, the parameter l is not under the influence of the $AIC(n)$. Therefore, we utilize $\log|\hat{\boldsymbol{R}}|$ to determine l.

(i) Memorize the q dimensional vector $\boldsymbol{y}(t)$ about the agent and m dimensional vector $\boldsymbol{u}(t)$ as a motor command.
(ii) From $l = 1 \cdots$, identify the obtained data.
  (a) If $\log|\hat{\boldsymbol{R}}| < 0$, stop the procedure and determine n based on $AIC(n)$,
  (b) else increment l until the condition (a) is satisfied or $AIC(n)$ does not decrease.

5 Reinforcement learning based on the local predictive models

Since the local predictive model merely represents the local interaction between the learner and one of the other objects separately, the learning robot needs to estimate the global interaction among models and decide to take actions to accomplish given tasks.

In the following, we give a brief explanation of Q learning and modular reinforcement learning to accelerate the learning time with multiple goals.

5.1 Q learning

A Q-learning method provides robots with the capability of learning to act optimally in a Markovian environment. A simple version of the Q-learning algorithm is shown as follows:

(i) Initialize $Q(x, u)$ to 0s for all combinations of $\boldsymbol{x}$ and $\boldsymbol{U}$.
(ii) Perceive current state x.
(iii) Choose an action u according to action value function.
(iv) Carry out action u in the environment. Let the next state be x' and the immediate reward be r.
(v) Update action value function from x, u, x', and r,

$$\begin{aligned} Q_{t+1}(x,u) = &(1-\alpha_t)Q_t(x,u) \\ &+\alpha_t(r+\gamma \max_{u' \in \boldsymbol{U}} Q_t(x',u')) \end{aligned} \tag{7}$$

where α_t is a learning rate parameter and γ is a fixed discounting factor between 0 and 1.
(vi) Return to 2.

5.2 Modular reinforcement learning

Since the time needed to acquire an optimal behavior mainly depends on the size of the state space, it is difficult to apply standard Q-learning to multiple tasks. Therefore, we use the modular reinforcement learning method [6].

Figure 4 shows the basic idea of the modular reinforcement learning, where the number of the tasks n is two for ease of illustration. In order to reduce the learning time, the whole state space is classified into two categories based on the maximum action values separately obtained by Q-learning: the area where one of the learned behaviors is directly applicable (*no more learning area*), and the area where learning is necessary owing to the competition of multiple behaviors (*re-learning area*). Then, all states $x \in \boldsymbol{X}$ are classified according to the Mahalanobis distance between the non-kernel state x and the kernel states x_{kernel}. Eventually composite state space $\boldsymbol{X}$ is classified into the no more learning area $\boldsymbol{X}_i$, $i = 1 \cdots n$ and the re-learning area $\boldsymbol{X}_{rl}$. These areas are exclusive.

In the case of states belonging to the no more learning area, the learning robot no longer needs to update the action value function. Therefore, the learning robot uses the action value functions which have been acquired previously. If the learning robot is in the re-learning area, the robot estimates the discounted value γ to learn the appropriate action value function. As a result, the modular reinforcement learning can take account of a tradeoff between the learning time and performance when the robot coordinates multiple behaviors.

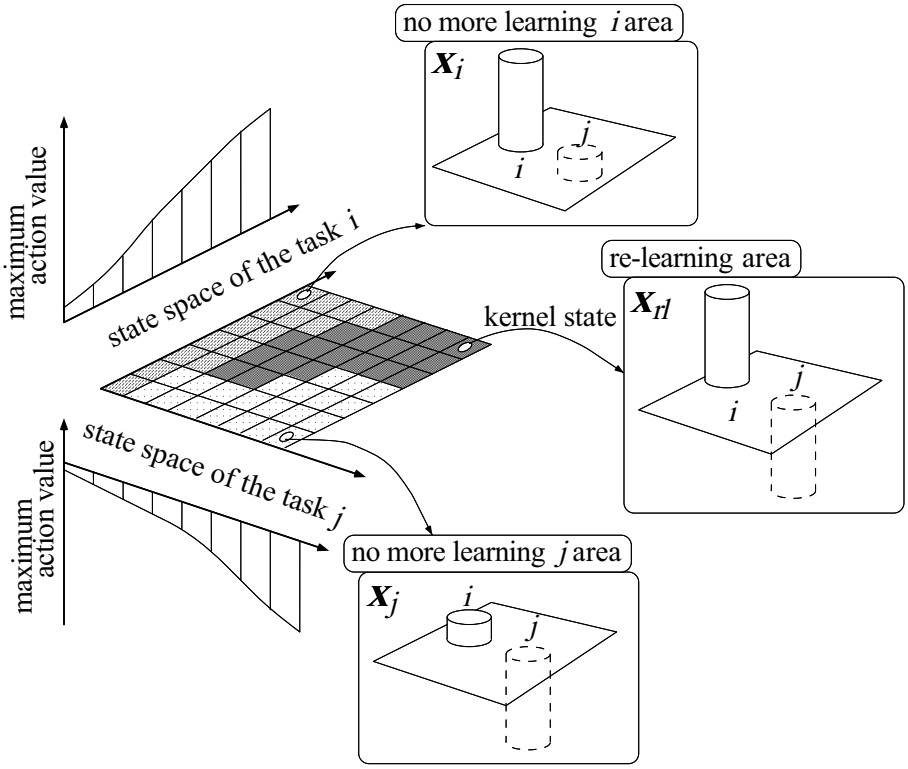


Fig. 4. Basic idea of the modular reinforcement learning

6 Experiments

6.1 Task and assumptions

We apply the proposed method to a simple soccer-like game that includes two mobile robots (Figure 5). Each robot has a single color TV camera and

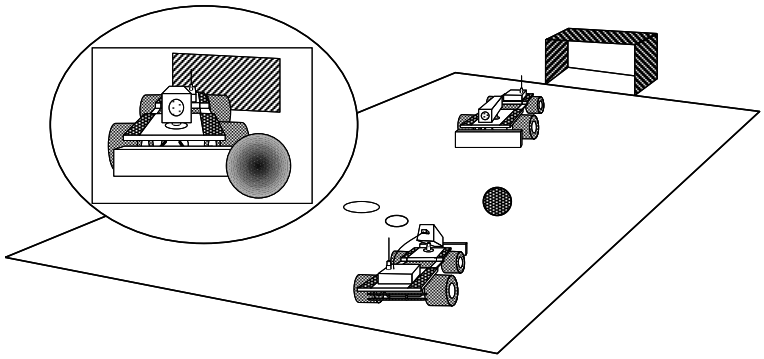


Fig. 5. The environment and our mobile robot

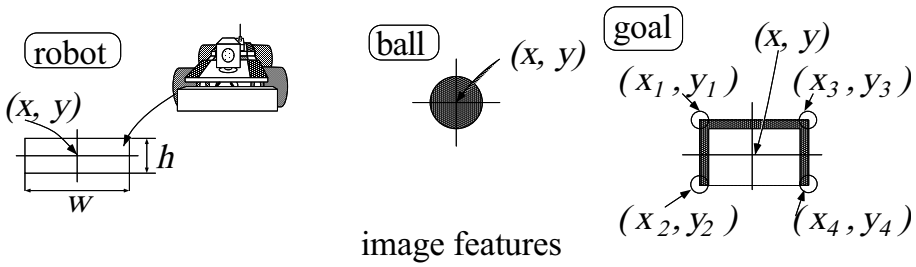
does not know the locations, the sizes, and the weights of the ball and the other agent. Nor does it know any camera parameters such as focal length and tilt angle, or kinematics/dynamics of itself. They move around using a 4-

wheel steering system. The effects of an action against the environment can be conveyed to the agent only through visual information. For motor commands, each agent has 7 actions such as go straight, turn right, turn left, stop, and go backward. The input $\boldsymbol{u}$ is defined as the 2 dimensional vector

$$\boldsymbol{u}^T = [v \;\; \phi], \quad v, \phi \in \{-1, 0, 1\},$$

where v and ϕ are the velocity of the motor and the angle of steering respectively and both of which are quantized.

The output (observed) vectors are shown in Figure 6. As a result, the dimen-



robot	ball	goal
area	area	area
center position	center position	center position
height	radius	4 corners
width		

Fig. 6. Image features of the ball, goal, and agent

sions of the observed vector about the other robot, the ball, and the goal are 5, 4, and 11, respectively.

6.2 Simulated and robotic experiments

First, the shooter and the passer construct the local predictive models for the ball, the goal, and the other robot in computer simulation. Next, the passer begins to learn the behaviors under conditions that assume that the shooter is stationary. After the passer has finished learning, we fix the policy of the passer. Then, the shooter starts to learn the shooting behaviors. We assign a reward value 1 when the shooter shoots a ball into the goal and the passer passes the ball to the shooter. A negative reward value -0.3 is given to the robots when a collision between two robots occurs. In these processes, modular reinforcement learning is applied for the shooter (passer) to learn certain shooting (passing) behaviors and avoiding others.

Next, we transfer the results of computer simulation to real environments. In order to construct the local predictive models in a real environment, the

robot selects actions using probabilities based on semi-uniform undirected exploration. In other words, the robot executes a random action with a fixed probability (20 %) and the optimal action learned in computer simulation (80 %). We perform 100 trials in robotic experiments. After the local predictive models are updated, the robots improve the action value function again based on the obtained data. If the local predictive model in the real environment increases the estimated order of the state vector, the action value functions are initialized based on the action value functions in computer simulation in order to accelerate learning. Finally, we perform 50 trials to check the result of learning in the real environment.

Table 1 shows the result of the estimated state vectors in computer simulation and real experiments, where $\log |\boldsymbol{R}|$ and AIC denote the logarithm of covariance matrix of error of the local predictive model and Akaike's information criterion, respectively. In order to predict the situation that follows, $l = 1$ is sufficient for the goal, while the ball needs 2 steps. Two reasons may explain why the estimated orders of state vectors are different between computer simulation and real experiments:

– Because of noise, the prediction error of real experiments is much larger than that of computer simulation.
– In order to collect the sequences of observation and action, the robots do not select the random action but instead move according to the result of computer simulation. Therefore, the experiences are quite different from each other.

As a result, the historical length l in the real experiments is larger than that of the computer simulation. On the other hand, the estimated order of state vector n for the other robot of the real experiments is smaller than that of the computer simulation, since the components for higher and more complicated interactions cannot be discriminated from noise in the real environments.

Table 2 shows the comparison of performances for the computer simulation and real experiments. We observed the result of replacing the local predictive models between the passer and the shooter. Eventually, large prediction errors on both sides were noted. Therefore, the local predictive models cannot be replaced between physical agents. Figure 7 shows a sequence of images where the shooter shoots a ball which is kicked by the passer.

7 Discussion

What kinds of image features should be used? Theoretically, any features can be considered. The necessary condition is that features should provide

Table 1
The estimated dimension

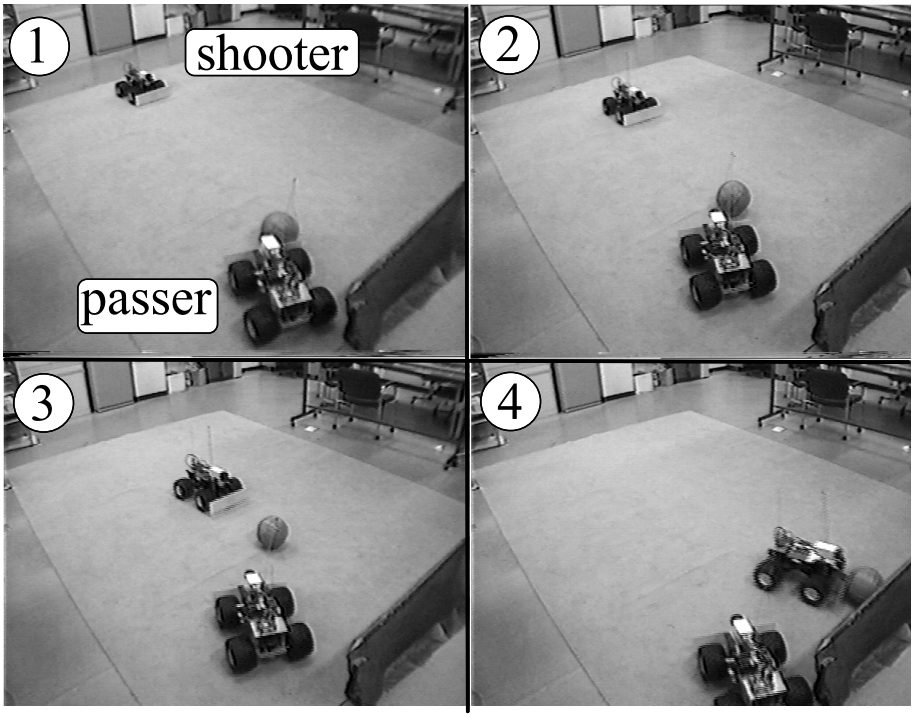
observer	target	l	n	$\log \lvert \boldsymbol{R} \rvert$	AIC
computer simulation					
shooter	ball	2	4	0.23	138
	goal	1	2	−0.01	121
	passer	3	6	1.22	210
passer	ball	2	4	0.78	142
	shooter	3	5	0.85	198
real experiments					
shooter	ball	4	4	1.88	284
	goal	1	3	−1.73	−817
	passer	5	4	3.43	329
passer	ball	4	4	1.36	173
	shooter	5	4	2.17	284

Table 2
Performance result in real experiments

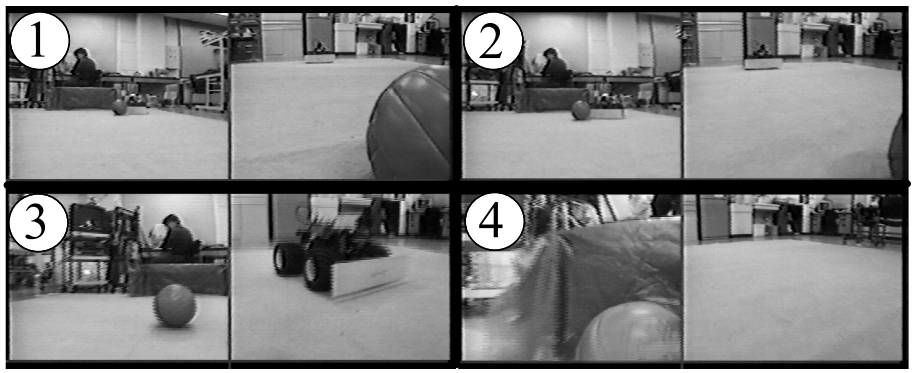
	success of shooting	success of passing
before learning	57/100	30/100
after learning	32/50	22/50

sufficient information for the agent to do the tasks at hand. The redundant information can be filtered by the CVA process, that is, the eigen values for the redundant information are lower than that of dominant components. In our experiment, we consider as many basic image features as possible such as centroid, area, size (radius, side), coordinates of boundary rectangle, etc. In the experiment, the dominant features were extracted and their linear combination constructed the state vectors.

There must be non-linearity of the relationship between objects. CVA is used for only state vector estimation, that is, linear approximation of the interactions between the learner and one of other agents, separately. We call such dynamics "lower dynamics." The role of reinforcement learning can be regarded that it might absorb the non-linearity of higher interactions among agents. Such interaction represents "higher dynamics of the system." We may conclude that as long as the number of other agents is not so large, say two or three, reinforcement learning is capable to absorb the non-linearity, but if the



(a) top view



(b) obtained images (left:shooter, right:passer)

Fig. 7. Acquired behavior

number increase, simple reinforcement application might not be sufficient to represent higher and more complicated interactions.

It seems difficult to apply the approach to dynamically changing environments, since the state vectors are determined by CVA and AIC off-line. We think that CVA could be performed on-line in the same way that PCA can be performed on-line, by neural networks. This may suggest a natural extension of the approach from off-line method to on-line one, and this is one of our

future works.

The use of AIC for vector size determination in the state vector estimation is not so convincing. Ideally, task performance is a better indicator than AIC of the trade off between too many state vectors (slow convergence) and too few (perceptual aliasing). But as a practical matter, determining task performance requires many iterations and may, therefore, be computationally infeasible. AIC, which is grounded in information theory, seems more principled than most other noniterative approaches.

In our approach, we suppose that the complexity of the environment seems to correspond to the complexity of interactions and, therefore, also the complexity of internal representation. One may claim that an ant walking across a beach may be walking on a complex environment, its behavior (trail) may be complex, but that doesn't mean its internal representation is complex.

From a viewpoint of classical AI, the geometrical complexities of beach (sand trial) and ant's kinematics might be very high, therefore the interaction between the ant and the environment measured by external observer seems very complicated. This might be wrong. The ant behavior seems purely reflexive and not so complicated internal representation is included. From the process of evolution, such a walking skill has been developed and embedded into ant genes.

The complexity we intend to claim here is not for geometry or structure measured by the external observer but for interaction between the agent and its environment. A good example is a case for a humanoid with many, say 30 or 40, DOFs to look at itself on the mirror. The motions of many joints generate complicated image patterns on the mirror. However, our method can identify that a single frame is sufficient to predict the change of the patterns since the image change can be simply correlated to self-induced motions.

8 Concluding remarks

This paper proposes a method of behavior acquisition that applies reinforcement learning to multi robot environments. Our method takes account of the tradeoff among the precision of prediction, the dimensionality of the state vector, and the number of steps needed.

As mentioned above, we believe that for an agent to interact with a dynamically changing world "perception" and "action" cannot be separable but must be tightly coupled in a physical body. Owing to the resource-bounded constraints, the internal representation should be abstracted (or symbolized); the

agent may have some symbols (to accomplish the given tasks) that might be shared between homogeneous physical agents. As a result, observation of other active agents and actions based on observation can be regarded as "communication." That is, observation has the role of message receiving while action the role of message sending. Unlike conventional approaches that provide a communication protocol to agents in advance our approach expects the agents to develop cooperative behaviors through the learning and development of their internal representation.

Acknowledgement

This work is partially supported by Japanese Society for Promotion of Science Project "Cooperative Distributed Vision for Dynamic Three Dimensional Scene Understanding." [6] Project ID: JSPS-RFTF96P00501. The authors thank to one of reviewers for the valuable comments to improve the paper.

References

[1] P. E. Agre. Computational research on interaction and agency. *Artificial Intelligence 72*, pp. 1–52, 1995.

[2] H. Akaike. A new look on the statistical model identification. *IEEE Trans. AC-19*, pp. 716–723, 1974.

[3] Y. Aloimonos. Introduction: Active vision revisited. In Y. Aloimonos ed., *Active Perception*, chapter 0, pp. 1–18. Lawrence Erlbaum Associate, Publishers, 1993.

[4] Y. Aloimonos. Reply: What I have learned. *CVGIP: Image Understanding*, 60:1:74–85, 1994.

[5] M. Asada. An agent and an environment: A view of "having bodies" – a case study on behavior learning for a vision-based mobile robot –. In *Proc. of 1996 IROS Workshop on Towards Real Autonomy*, pp. 19–24, 1996.

[6] M. Asada, S. Noda, S. Tawaratumida, and K. Hosoda. Purposive behavior acquisition for a real robot by vision-based reinforcement learning. *Machine Learning*, 23:279–303, 1996.

[7] Asada, M., S. Noda, and K. Hosoda (1996). Action-based sensor space categorization for robot learning. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems 1996 (IROS '96)*, pp. 1502–1509.

[8] Ballard, D. H, Hayhoe, M. M, Pook, PK, Rao, R. P. N. Deictic codes for the embodiment of cognition. *BEHAVIORAL AND BRAIN SCIENCES*, 1997 DEC, Vol.20, No.4, pp. 723–.

[9] J. H. Connel and S. Mahadevan. *Robot Learning.* Kluwer Academic Publishers, 1993.

[10] S. Edelman. Reply: Representation without reconstruction. *CVGIP: Image Understanding*, 60:1:92–94, 1994.

[11] M. Inaba. Remote-brained robotics : Interfacing AI with real world behaviors. In *Preprints of ISRR'93*, Pittsburgh, 1993.

[12] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, E. Osawa, and H. Matsubara (1997). "Robocup: A challenge problem of AI". *AI Magazine 18*, 73–85.

[13] W. E. Larimore. Canonical variate analysis in identification, filtering, and adaptive control. In *Proc. 29th IEEE Conference on Decision and Control*, pp. 596–604, Honolulu, Hawaii, December 1990.

[14] L.-J. Lin and T. M. Mitchell. Reinforcement learning with hidden states. In *Proc. of the Second International Conference on Simulation of Adaptive Behavior: From Animals to Animats*, pp. 271–280, 1992.

[15] M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proc. of the 11th International Conference on Machine Learning*, pp. 157–163, 1994.

[16] T. Matsuyama. Cooperative Distributed Vision. In *Proc. of First International Workshop on Cooperative Distributed Vision*, pp. 1–28. 1997.

[17] A. W. Moore and C. G. Atkeson. The parti-game algorithm for variable resolution reinforcement learning in multidimensional state-spaces. *Machine Learning*, 21:199–233, 1995.

[18] T. Nakamura and M. Asada. Motion sketch: Acquisition of visual motion guided behaviors. In *the 14th International Joint Conference on Artificial Intelligence*, pp. 126–132. Morgan Kaufmann, 1995.

[19] T. Nakamura and M. Asada. Stereo sketch: Stereo vision-based target reaching behavior acquisition with occlusion detection and avoidance. In *Proc. of IEEE International Conference on Robotics and Automation*, pp. 1314–1319, 1996.

[20] G. Sandini. Vision during action. In Y. Aloimonos ed., *Active Perception*, chapter 4, pp. 151–190. Lawrence Erlbaum Associate, Publishers, 1993.

[21] G. Sandini and E. Grosso. Reply: Why purposive vision. *CVGIP: Image Understanding*, 60:1:109–112, 1994.

[22] P. Stone and M. Veloso. Using machine learning in the soccer server. In *Proc. of IROS-96 Workshop on Robocup*, 1996.

[23] Takahashi, Y., M. Asada, and K. Hosoda (1996). Reasonable performance in less learning time by real robot based on incremental state space segmentation. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems 1996 (IROS96)*, pp. 1518–1524.

[24] E. Uchibe, M. Asada, and K. Hosoda. Behavior coordination for a mobile robot using modular reinforcement learning. In *Proc. of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1329–1336, 1996.

[25] E. Uchibe, M. Asada, and K. Hosoda. State space construction for behavior acquisition in multi agent environments with vision and action. In *Proc. of International Conference on Computer Vision*, 1998 (to appear).

[26] C. J. C. H. Watkins and P. Dayan. Technical note: Q-learning. *Machine Learning*, pp. 279–292, 1992.

[27] S. D. Whitehead and D. H. Ballard. Active perception and reinforcement learning. In *Proc. of Workshop on Machine Learning-1990*, pp. 179–188. Morgan Kaufmann, 1990.