Intelligent Control of Autonomous Soccer Robots Compensating Missing Information

Hikari Fujii, Nobuyuki Kurihara and Kazuo Yoshida
Keio University
3-14-1, Hiyoshi, Kohoku-ku, Yokohama, Kanagawa 223-8522, Japan
email: fujii@yoshida.sd.keio.ac.jp

***Abstract*— The acquisition of visual information of an object is often disturbed in real environment. However, it is required for a robot to take a suitable action even if the information is missing. This paper presents a method to compensate missing information for autonomous mobile robots by using short term memory. The action selector uses compensated information and determines a suitable action. The action selector consists of a neural network whose connection weights are learned by the genetic algorithm. RoboCup soccer robot is chosen as a demonstration target. The experimental and simulation results show its effectiveness.**

***Index Terms*— Autonomous Mobile Robot, Short Term Memory, Compensating, Action Selecting, RoboCup.**

I. INTRODUCTION

The acquisition of visual information of an object is often disturbed in real environment. However, it is required for a robot to take a suitable action even if the information is not acquired or enough accurate. Since robots determine action according to the information about environment, the lack of information has an influence on action selection. Therefore, it is important to compensate missing information.

There are some methods to compensate the missing information [1], [2]. One of the methods of compensating the missing information is to use memory in robot. With compensated information, robots can act as if they recognize the information without any missing information. If the compensated information is not reliable, robot should not use that information. Therefore this paper presents a method to compensate information and select an action according to the reliability index of the information.

The purpose of this study is to establish a control method for autonomous mobile robots in the dynamic environment with uncertainty on the acquisition of information. In this study, each robot has action modules and action selector. The action module is designed as the action for midterm objective. The midterm objective is achieved with selecting the action modules continuously. The action module is selected by the action selector in each frame. To realize the purpose, the compensating method of missing information and the control method using two kinds of action selectors are proposed. With this method, the knowledge of designer and the experience of robot obtained through the simulation are combined according to the condition about the acquisition of the visual information.

To compensate the information, a short-term memory is used. In this method, the short-term memory is constructed based on the Atkinson and Shiffrin model [3] and the forgetting curve by Brown Peterson Paradigm [4], [5]. The compensated information is utilized or not according to the evaluation of the information stored in short-term memory and estimated information. And the reliability index of information is calculated to use for switching the two kinds of action selectors. One of them is designed based on the knowledge of a designer. The other is constructed by a neural network whose weights are learned by the genetic algorithm.

To examine the effectiveness of this method, the goalkeeper robot for RoboCup [6], [7] middle size league is chosen as a test bed. RoboCup Middle Size League has some restrictions about robots construction. One of the most important restrictions is to forbid robots to have global sensors. There is a possibility that robots can not recognize some area because of this restriction. Robots that act in real world have the same restriction. In RoboCup Simulation League, the designing method of agents with short-term memory and long-term memory is proposed [8]. However, the definition of the short term memory and the design of the action selector are different from our method.

The usefulness of the proposed method is shown through the simulation and the experiments.

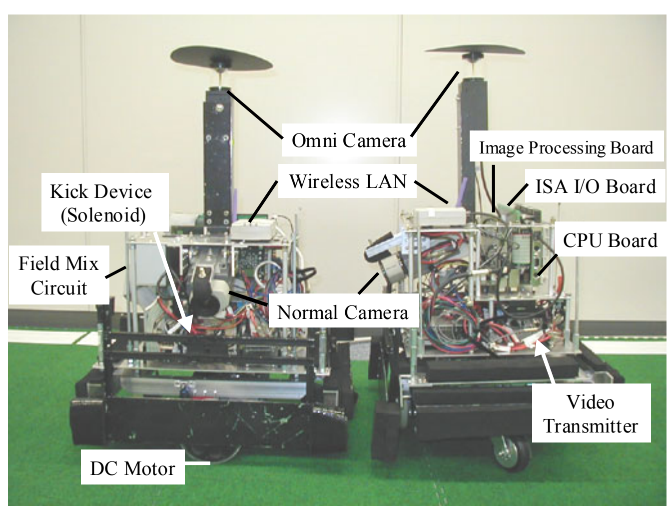


Fig.1 Goal keeper robots.

This study is supported in part by Grant in Aid for the 21st century center of excellence for "System Design: Paradigm Shift from Intelligence to Life" from Ministry of Education, Culture, Sport, and Technology in Japan and the foundation of Keio university.

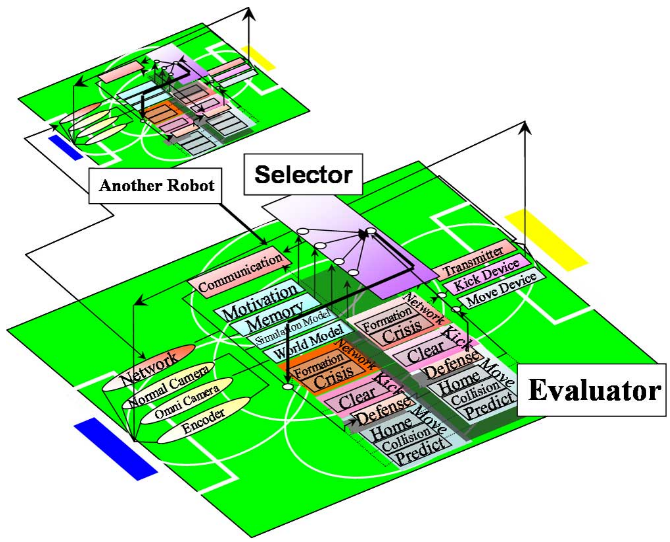


Fig.2. Concept of action control of the robot.

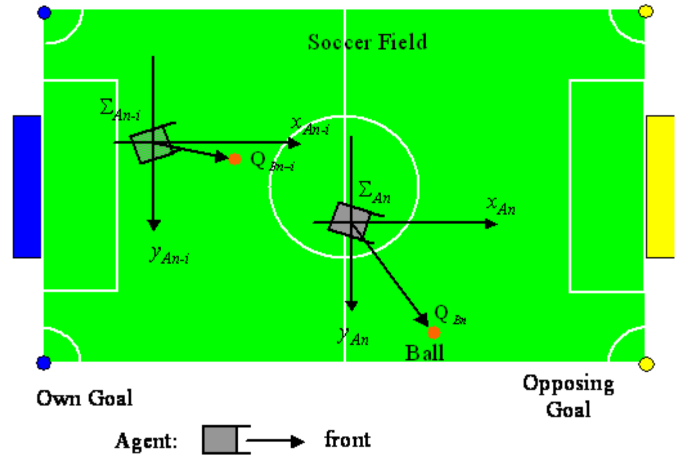


Fig.3 The coordinate for the ball position.

II. HARDWARE AND SOFTWARE SYSTEM

A. Hardware System

The goal keeper robots are shown in Fig.1. These robots have two cameras, omni-directional camera and normal camera, and two encoders as sensor. It has Pentium II 450MHz CPU and image processing board. It has a differential drive and a kicking device with solenoid. They have wireless LAN for communicating with other robots.

B. Software System

The concept of action control of our robot system is shown in Fig.2. The robot recognizes a ball and goals, and calculates its position and orientation from the information about landmarks. With acquired information, the robot selects its action. The robot has action modules. Action modules are defined as the midterm objective of the behavior for achieving a task. The action selecting method is described in the chapter 4.

III. COMPENSATING AND CALCULATE RELIABILITY METHOD

A. Concept

In this method, the robot has a short-term memory (STM). It uses the STM for compensating the position of the moving object. The STM is designed based on the forgetting curve given by Brown and Peterson. To compensate the missing information, the predicted information about current missing information is applied. The predicted information is calculated from each past memory. The weighted average of the predicted information is calculated. These weights are based on STM. The decision with regard to the usage of the compensated information depends on the continuity level about the acquisition of the information. The reliability of the information is calculated according to the continuity level about the acquisition of the information and the transition rate of estimated information of the target. The details of these criteria are described section B and C.

In this paper, the goal keeper robot for RoboCup Middle Size League is chosen as a test bed. For the goal keeper robot, the objective is to save the goal. Therefore, the ball is defined as a target object. The methods of predicting and compensating are shown in the following section.

B. Predicting method

One of the characteristics of our omni-directional vision system is that the distance between the target and the robot can not be recognized accurately, in comparison with the direction between them. With taking account of these characteristics, a method to predict ball position is proposed as follows.

The coordinate of the visual information of the target is shown in Fig.3. The origin of this coordinate is the current position of the robot. The direction for the opponent goal is x positive direction. This coordinate is defined as a rectangular coordinate system. The used information is the distance between ball and robot, and the directions for the ball and both goals. The robot can recognize its orientation through the directions for both goals.

The position vector of the ball at the current state $\boldsymbol{S}_n$ is $\boldsymbol{Q}_{Bn} = \left(x_{Bn} \quad y_{Bn}\right)^T$. The velocity of the ball $\boldsymbol{V}_{Bn} = \left(\dot{x}_{Bn} \quad \dot{y}_{Bn}\right)^T$ is calculated according to Eq.(1) from the current position vector and the past position vector $\boldsymbol{Q}_{Bn-i}$ at state $\boldsymbol{S}_{n-i}$. The $\boldsymbol{t}_{n-i}$ means the elapsed time from i frame past. The i is the number of the latest high reliable information on the memory.

$$\boldsymbol{V}_{Bn} = \left(\bar{\boldsymbol{Q}}_{Bn} - \bar{\boldsymbol{Q}}_{Bn-i}\right) / t_n + k\boldsymbol{V}_{An} \tag{1}$$

In order to cope with the noise on the image, the information of the ball position $\bar{\boldsymbol{Q}}_{Bn}$:(N=3) which is the average among the past 3 steps information is used. The $\boldsymbol{V}_{An} = \left(\dot{x}_{An} \quad \dot{y}_{An}\right)^T$ is the velocity vector of the agent calculated from encoders. The k is the weight for agent's velocity. The $\bar{\boldsymbol{Q}}_{Bn-i}$ is the ball position in the current coordinate Σ_{An}. The coordinate transformation from the past coordinate Σ_{An-i} into the current coordinate Σ_{An} is realized with using the encoder information. The predicted ball position $\boldsymbol{Q}_{Bn+j}$ is calculated with Eq.(1) and Euler's method.

In this study, the reflection of the ball is considered for predicting a ball path. In this paper, the reflection is detected according to the trend of the ball velocity $\boldsymbol{T}_{Bn}$ shown in the fol-

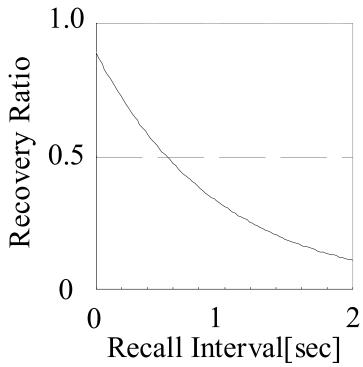


Fig. 4. Forgetting curve from Brown Peterson Paradigm

lowing equation.

$$\boldsymbol{T}_{Bn} = \sum_{m=0}^{M} \operatorname{sgn}\left(\boldsymbol{V}_{Bn-m}\right) \tag{2}$$

where,

$$\boldsymbol{T}_{Bn} = \begin{pmatrix} T_{V_{Bx}} & T_{V_{Bx}} \end{pmatrix}^{T} \text{ , M=10}$$

If the one of the elements of the value of $\boldsymbol{T}_{Bn}$ is changed into M-2 from M, the ball is reflected. The information of the prediction is deleted when the ball reflection is detected.

C. Compensating method

A method to compensate the missing information is to use the short-term memory based on Atkinson and Shiffrin model [3] and the forgetting curve by Brown and Peterson [4], [5]. In this system, the information of the static objects such as goals and corner poles are defined as the information stored in long-term memory. The information of the dynamic objects such as the ball is defined as the information stored in short-term memory. The method to compensate missing information is described as follows.

The variables are defines as follows:

$R(t)$	:	Recovery rate function calculated from the forgetting curve.
S_n	:	State at the n term.
$\boldsymbol{Q}_n = \left(x_n, y_n\right)^T$	:	Object position in S_n.
$\boldsymbol{Q}_{n-m} = \left(x_{n-m}, y_{n-m}\right)^T$	:	Estimated object position in S_n calculated from S_{n-m}.
$\boldsymbol{Q}_{datum_n} = \left(x_{datum_n}, y_{datum_n}\right)^T$	:	Estimated object position from all memory in S_n.
P_{n-m}	:	Reliability index of predicted data.
T_{n-m}	:	Reliability index of estimated data with the proposed method.
w_{n-m}	:	Weight for calculating $\boldsymbol{Q}_{datum_n}$
VC	:	Continuity of the visible flag.
TR	:	Transition rate of the estimated information.
$Q_{ErrorMax}$	:	Estimated maximum error

The recovery ratio of the short-term memory $R(t)$ is designed based on the forgetting curve. Humans forgetting curve is shown in Fig.4. In this study, the coefficient of the function is changed to apply it to the robot in the following equation:

$$R(t) = A\exp\left(-Bt\right) \qquad A = 0.89 \qquad B = 1.0514 \tag{3}$$

The compensating information is calculated using the information stored in short-term memory. To calculate the compensating information, the predicting method above-mentioned is used. Q_{m} means the predicted current position of the object from the information of m terms before. The estimated current ball position from all memory Q_{datum} is calculated from the following equations:

$$\boldsymbol{Q}_{datum_n} = \frac{\sum_{m=1}^{Memory} w_{n-m}\boldsymbol{Q}_{n-m}}{\sum_{m=1}^{Memory} w_{n-m}} \tag{4}$$

where

$$w_{n-m} = T_{n-m}P_{n-m}R\left(term_{n-m}\right) \tag{5}$$

The $term_m$ means the elapsed time from m terms before. The T_m is the reliability index of Q_m.

In this study, the P is the visible rate of the past information. The reliability index T is calculated from continuity (VC) about the acquisition of the information and the transition rate (TR).

$$T_n = VC_n \cdot (1 - TR_n) \tag{6}$$

The VC means the continuity about the acquisition of the information with the recovery ratio. The VC is calculated from the visibility of object, and the elapsed time from the latest observed time of object. It is defined by the following equation:

$$VC_n = \frac{\sum_{m=0}^{Memory} R\left(term_{n-m}\right)V_{n-m}}{\sum_{m=0}^{Memory} R\left(term_{n-m}\right)} \tag{7}$$

where

$$V_i = \begin{cases} 0 & \text{(if the object is invisible)} \\ 1 & \text{(if the object is visible)} \end{cases} \tag{8}$$

The TR means the transition rate of the estimated information. If the TR is too big, there is a big difference between the past information and predicted information. This means that the predicted information is not reliable. The TR is defined by the following equation:

$$TR_n = \frac{\left|Error\right|}{Q_{Error\,Max}} \tag{9}$$

where

$$Error = \sqrt{\left(x_n - x_{datum_n}\right)^2 + \left(y_n - y_{datum_n}\right)^2} \tag{10}$$

The missing information is compensated with the predicted information according to the following equation:

$\boldsymbol{Q}_{SLI_n}$	:	Compensating data.
TR_{thre}	:	Threshold of the transition rate for compensating.
VC_{thre}	:	Threshold of the continuity of the visible flag for compensating.

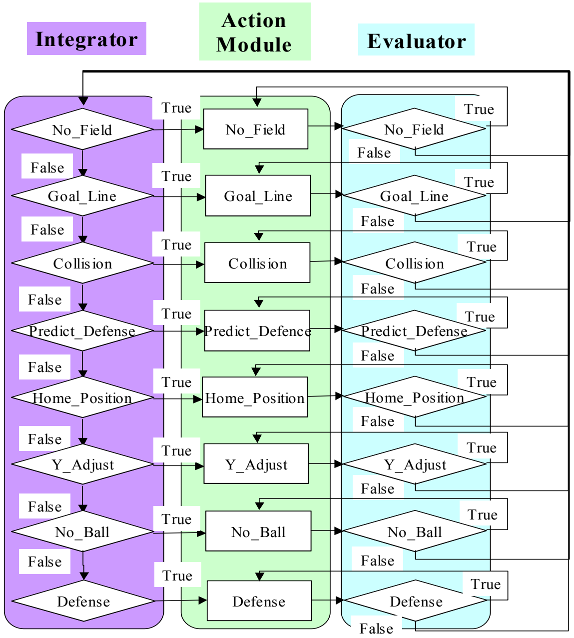


Fig. 5. Flow of the rule base action selector

case invisible

$$Q_{SLI_n} = \begin{cases} Q_{datum_n} & (VC_n > VC_{thre}) \\ invisible & (VC_n \le VC_{thre}) \end{cases} \quad (11)$$

case $TR_n > TR_{thre}$

$$Q_{SLI_n} = \begin{cases} Q_{datum_n} & (VC_n > VC_{thre}) \\ Q_n & (VC_n \le VC_{thre}) \end{cases} \quad (12)$$

According to Eqs.(11) and (12), the robot can compensate missing or unreliable information.

The thresholds of the *VC* and *TR* are determined according to the designer's experience.

IV. Action Selection Method

A. Concept

In this study, each agent has action modules and action selector. The action module is designed as the action for midterm objective, such as defense, going to home position and so on. These midterm objectives are achieved with selecting the action modules continuously. The agent can select the action modules according to the adoption of the action selector in each frame.

The action selector selects a suitable action module according to the internal and external information. In this study, two kinds of the action selectors are proposed. One is the rule base action selector, and the other is constructed by the neural network learned with the genetic algorithm.

This method is applied to a goal keeper robot for RoboCup Middle Size League. These action selectors and the combining method are described in the following section.

B. Rule base action selector

The rule base action selector is designed according to designer's knowledge. The flow of the action selector is shown in Fig.5. The robot has 8 kinds of action modules. The action selector checks the rule and determines the action. With this action selector, the robot can behave suitably in assumed situation.

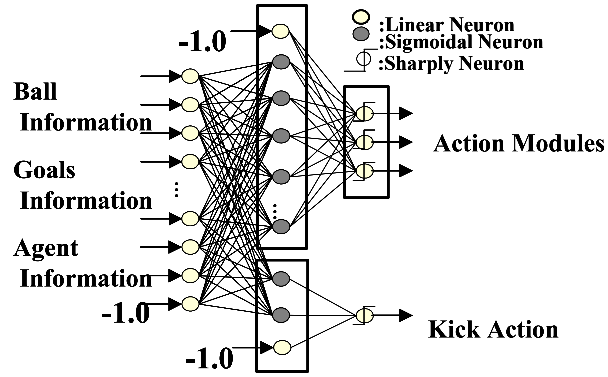
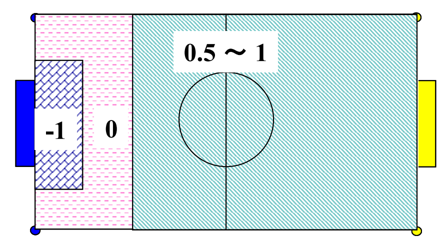


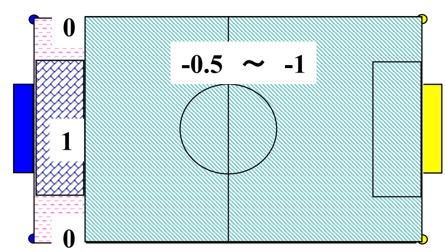
Fig. 6. Neural network action selector

TABLE I
WEIGHTS OF THE FITNESS FUNCTION

	Value			Range
α_G	1.00	$G(i)$	Get goal	0～15
α_L	-8.00	$L(i)$	Lost goal	0～15
α_B	0.001	$B(i)$	Ball position	-6000～6000
α_A	0.50	$A(i)$	Agent position	-6000～6000
α_E	0.70	$E(i)$	Self evaluation	-60000～60000
α_{Cl}	0.03	$Cl(i)$	Clear evaluation	-48000～60000
α_V	-0.20	$V(i)$	Battery	0～100



(a) Image of the evaluation of ball position: In the penalty area, the value of the ball position evaluation takes -1. Near the penalty area, its value is 0. On the other area, its value takes 0.5 to 1 according to distance between ball and own goal. Its value takes 1 when the ball is far from own goal.



(b)Image of the evaluation of agent position: In the penalty area, the value of the agent position evaluation takes 1. On the side of penalty area, it takes 0. On the other area, it takes -0.5 to -1 according to the distance from own goal.

Fig. 7. Evaluation of the ball position and agent position.

C. Neural network action selector

The neural network action selector has 15 input units, 19 hidden units and 4 output units and its weights are obtained through simulation study by the genetic algorithm. This network is divided into two parts. One is for the action selection, and the other is for the decision to kick. The action selection part has 15 input units, 16 hidden units and 3 output units. The kick decision part has 15 input units, 3 hidden units and 1 output unit. The input information is as follows:

Ball information	:	Distance, Angle, Visible flag, Velocity of x coordinate, predicted information
Opponents goal information	:	Distance, Angle
Own goal information	:	Distance, Angle, Visible flag, Both of the side information.
Relation of ball and own goal information	:	Angle between the ball and own goal
Robot information	:	Collision flag, Battery level, Elapsed time

The activation functions of hidden units are sigmoid function. The activation functions of output units are threshold.

The situation of the simulation for learning is the 2 on 2 game with 1 field player. The length of the game is 300 seconds.

The fitness function is designed by the following equation:

$$Fittness(i) = \alpha_G G(i) + \alpha_L L(i) + \alpha_B B(i) + \alpha_A A(i) + \alpha_E E(i) + \alpha_{Cl} Cl(i) + \alpha_V V(i) \quad (13)$$

The coefficients of the variables and range of the variables are shown in Table 1. The values of the variables are described in Eqs.(14), (15) and Fig. 7. Equations (14) and (15) mean that the *E(i)* takes the positive value if the determination of the neural network action selector is same as the determination\ of the rule base action selector. If the selected action by the neural network action selector is defense, the *E(i)* takes 1 or -1 according to the determination of the rule base action selector. The *P(i)* is reciprocal of the selecting rate which is defined by designer, and *k(i)* is a coefficient. The clear evaluation takes 10 when it selects clear action and the same as the rule base action selector's determination. If the clear action is selected by the neural network action selector, and it isn't the same as rule base one, the clear evaluation takes -0.01. The clear evaluator takes -8 when the neural network does not select clear action and the rule base one selects clear action. These coefficients and the value of the *E(i)* are designed according to the experience.

case Defense

$$E(i) = \begin{cases} 1 & (Evaluate(i) = Output(i)) \\ -1 & (Evaluate(i) \neq Output(i)) \end{cases} \quad (14)$$

case default

$$E(i) = \begin{cases} k(i)/P(i) & (Evaluate(i) = Output(i)) \\ -0.2 & (Evaluate(i) \neq Output(i)) \end{cases} \quad (15)$$

D. Combining method

The rule base action selector is suitable for the assumed situation and it is easy to adjust it by designer. The neural network action selector is expected to select suitable action even if robot is in the unexpected situation. Therefore, the rule base action selector and the neural network action selector are combined according to the C_n. The C_n is calculated from T_n; trust of the ball position in the current state S_n. Combining both action selectors are realized according to the following equation. The NNS_n means the neural network action selector, and the RBS_n means the rule base action selector.

$$C_n = \alpha \cdot T_n \cdot (1 - T_n) \quad (16)$$

$$Action\ selector_n = C_n \cdot NNS_n + (1 - C_n) \cdot RBS_n \quad (17)$$

$$Action\ selector_n = \begin{cases} NNS_n & (C_n > C_{thre}) \\ RBS_n & (C_n \leq C_{thre}) \end{cases} \quad (18)$$

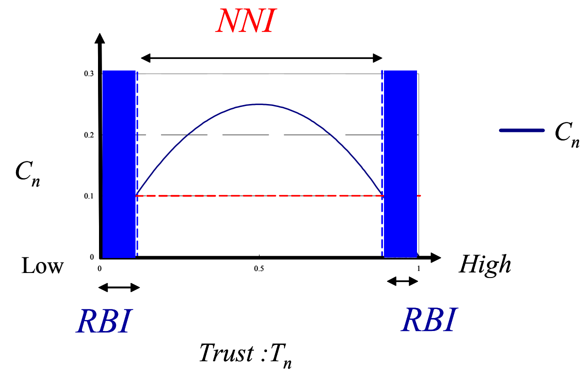


Fig. 8. Flow of the rule base action selector

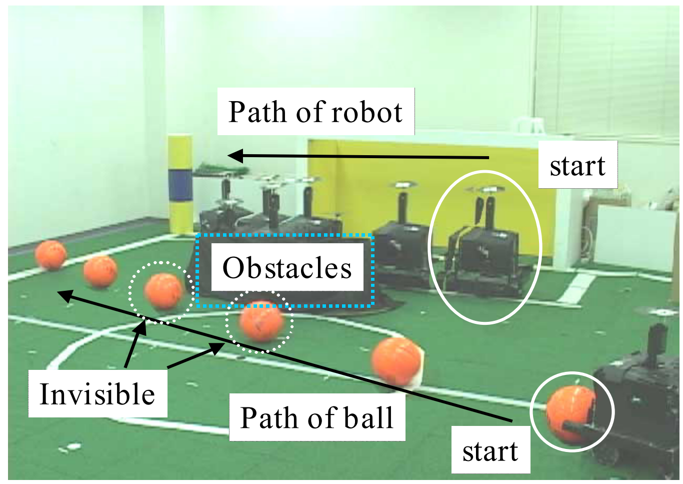


Fig. 9. Result of experiment

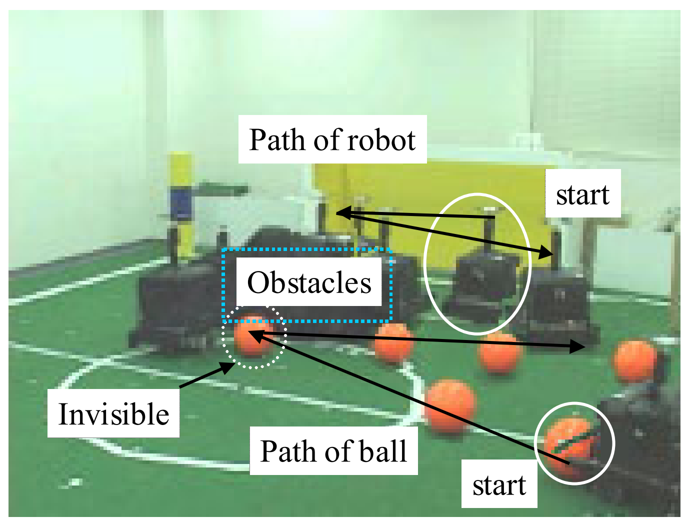


Fig. 10. Result of experiment with rebounding

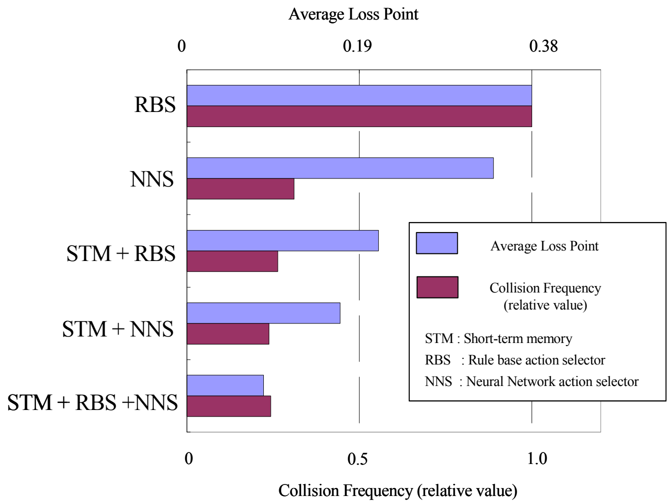


Fig. 11. Result of the simulated game with 5 kinds of action selectors.

In this study, the α is defined as 1, C_{thre} is defined as 0.01. The scheme of this combining method is shown in Fig. 8.

V. Experiment and Simulation

A. Experiment condition

To examine the effectiveness of this method, the experiment with real robots is carried out. Experimental situation is that the field players are put in front of the goal keeper as an obstacle and the kicker kicks the ball in the direction of the arrow shown in Fig. 9. The path of the ball is not visible temporarily from the goal keeper. The goal keeper compensates the missing information of the ball when the ball is not visible. The experiment is performed for 2 kinds of situations as described bellow.

Case 1 : The ball is passing behind the obstacles.
Case 2 : The ball is rebounded behind the obstacles.

B. Experimental result

The result of the case 1 is shown in Fig.9 and the result of the case 2 is shown in Fig.10. The goal keeper can follow the ball when the ball is not visible. The goal keeper is also able to follow the moving ball even if the ball path is different from the predicted path because of rebounding at invisible situation.

C. Simulation

In this simulation, the noises, the occlusions and the rules of the game in RoboCup Middle Size League; such as the replacement, are taken into consideration.

With this simulator, the 5 kinds of controllers are examined.

(a) Rule base action selector for the RoboCup 2002
(b) Neural network action selector
(c) Rule base action selector with compensating missing information
(d) Neural network action selector with compensating missing information
(e) Combination of the rule base action selector and neural network action selector with compensating missing information

The game takes 600 seconds and the opponents are the simulated EIGEN team using (a) action selector.

D. Simulation result

1) Obtained action selector

As a result of the learning the neural network action selector, the waiting action is expressed by changing the action module continually. This is suitable for Ball Replacement rule in the RoboCup Middle Size League. With this action, the number of the collision is decreased.

2) Simulated game result

To examine the effectiveness of the combining the rule base action selector and the neural network action selector, the game is simulated with each method 24 times. The average of the lost goal and the frequency of the collision with team mates are shown in Fig. 11. The frequency of the collision is calculated in comparison with result of (a).

With using NNS or STM, the frequency of the collision is decreased. The average of lost goal is decreased with the combined action selector with STM.

VI. Conclusion

In this study, the compensation method and the action selection methods are proposed. The proposed system has a short-term memory (STM) based on Atkinson and Shiffrin model and the forgetting curve given by Brown and Peterson, in order to compensate the missing information. The characteristics of the proposed action selector are the combination of the 2 kinds of action selectors; the rule base one and the neural network one. The combining method is achieved according to the reliability index of the compensated position of the object. The usefulness of this control system was shown through the numerical simulation and the experiments. The future subject of this study will be the formulation of reliability index for common object and the construction of the action selector which takes account of the relationship between the reliability index and action selection.

References

[1] N. Ito, K. Shinoda, K. Nakagawa, K. Iwata, X. Du and N, Ishii, “Action Decision with Incomplete Information: A Temporal Database Approach”, IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man and Cybernetics, Vol. 6, pp.722-727,1999
[2] T. Schmitt, R. Hanek, S. Buck and M. Beetz, “Cooperative Probabilistic State Estimation for Vision-Based Autonomous Soccer Robots”, RoboCup 2001: Robot Soccer World Cup V, pp.193-203, 2002.
[3] R. C. Atkinson, R. M. Shiffrin, “Shuman memory: A proposed system and its control processes,” in K. W. Spence, J. T. Spence (Eds.), The psychology of learning and motivation: Advances in research and theory. Vol. 2. New York: Academic Press, 1968.
[4] J. Brown, “Some tests of the decay theory of immediate memory” , Quarterly Journal of Experimental Psychology, Vol.10, pp.12-21, 1958.
[5] L.R. Peterson, M. J. Peterson, “Short-term retention of individual verbal items”, Journal of Experimental Psychology, Vol.58, No.3, pp.193-198, 1959.
[6] G. A. Kamink, P. U. Lima, and R. Rojas (Ed.), RoboCup 2002: Robot Soccer World Cup VI, pp.435--442, Springer, 2003.
[7] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, E. Osawa, “RoboCup: The Robot World Cup Initiative”, Proceedings of the IJCAI-95 Workshop on Entertainment and AI/Life, pp 19-- 24, 1995.
[8] F. Torterolo and C. Garbay, “A Hybrid Agent Model, Mixing Short Term and Long Term Memory Abilities An Application to RoboCup Competition” , RoboCup 98: Robot Soccer World Cup II, pp.246--260, Springer, 1999.