From Automation To Autonomy

Kentarou Kurashige[1], Yukiko Onoue[1] and Toshio Fukuda[2]
[1]*Muroran Institute of Technology,*
[2]*Nagoya University*
Japan

1. Introduction

Recently, there are many researches on intelligence in the field of engineering from various viewpoints. Representative aim is to satisfy two desires. One desire is to want more convenient machine (Kawamoto et al., 2003; Kobayashi et al., 1999; Hasegawa et al., 2004). Researchers have tried to improve existing machines or invent new machines. And now, researchers consider realizing new one by incorporating with a mechanism of life intelligence. Another desire is to want to know what intelligence is. Here, a purpose is to elucidate a mechanism of intelligence and to create it (Asada et al., 2001; Brooks & Stein, 1994; Goodwin, 1994). Researchers have expected that utility will be made known as a result of various studies.
As the milestone for intelligent machine, realizing autonomy on machine as a progress from automation is expected. The research of automation can be regarded as study how to make proper outputs by rules which human prepared. It is smarter than operating machine manually, but still not intelligent. Autonomy can be regarded as a mechanism which can make rules corresponding with surrounding environment and make proper outputs by making rules.
As one method to realize autonomy on machine, there are researches into machine learning. Especially, researches using soft computing method are so active. Essence of learning is making knowledge through trial and error and making outputs using this knowledge (Jordan, 1992). Expression of knowledge is different between each method, for example neural network (Nolfi & Parisi, 1997) has knowledge with weight matrix, but knowledge can be regarded as a rule which is mapping from input to output. Here, we have been free from necessity of a load that we must make rules to get proper outputs for all situations machine will face.
But new problem has occurred and we have gotten new load when we use learning method. We must make evaluation to learn a task or environment. In the framework of machine learning, human imagines a task which he/she gives to machine at first. Next, human must design evaluation which is a way how to teach a machine human desire. Evaluation functions expressed by numerical formula are used mostly as evaluation. The point of this is that these functions are closely related with context. So it is possible that evaluations of one output on different tasks are different values. Evaluation is strongly affected by a task, environment or a viewpoint of researchers. For this reason, a machine can work only for taught task and it is difficult to apply acquired knowledge or rules for other tasks. Human must design evaluation for all tasks individually. This load is heavy; especially in a case of

Source: Machine Learning, Book edited by: Abdelhamid Mellouk and Abdennacer Chebira,
ISBN 978-3-902613-56-1, pp. 450, February 2009, I-Tech, Vienna, Austria

robot which has the ability to achieve various tasks and cause changeful environment by its moving ability, human must persevere in design of evaluation functions.
To overcome this problem, we focus on learning based on universal evaluation. We define universal evaluation as evaluation which is independent of a task or task information and environment a machine will be used. And we try to realize a mechanism which can learn with universal evaluation. In this chapter, we show two challenges using robot as application. One challenge is study of learning with sense of pain as universal evaluation (Kurashige & Onoue, 2007). Another challenge is study about creation of evaluation functions for concrete task and environment with energy as universal evaluation (Kurashige et al., 2002). On both challenges, we show robot can learn and create proper movement for a task or environment robot will face.

2. Learning with sense of pain on robot

In this section, we show a case of learning by using sense of pain on robot as universal evaluation (Kurashige & Onoue, 2007). We think universal evaluation must be independent of information related with each task and environment robot will face. Here, we consulted evolutionary process. Instinct which life has innately is important to keep living, and is independent of concrete environment it will face to a certain extent. Sense of pain, which is a kind of instinct, is especially important to detect abnormal state. Life can learn avoiding fatal injury with this instinct. We define sense of pain on robot and make robot learn to protect itself. And it is so hard to learn various concrete tasks only with universal evaluation. So we combine learning based on universal evaluation with usual learning method. We construct a learning system with both learning and expect that operator will be able to design evaluation function for each task easier by focusing only on a task.
We explain proposed system at first, and next we show an experiment with small-sized humanoid robot.

2.1 Outline of proposed system using sense of pain

Proposed system consist of three component; usual learning method, learning by sense of pain, action adjuster. Usual learning method is for learning a task human wants to give a robot. Here operator designs evaluation function for a task. Learning by sense of pain is for learning avoiding fatal injury. This learning is not related with each task and can be used to various tasks. Each component creates or selects action independently, so these actions conflict sometimes. Proposed system must need action adjuster to solve this problem.
We show outline of proposed system in fig. 1.

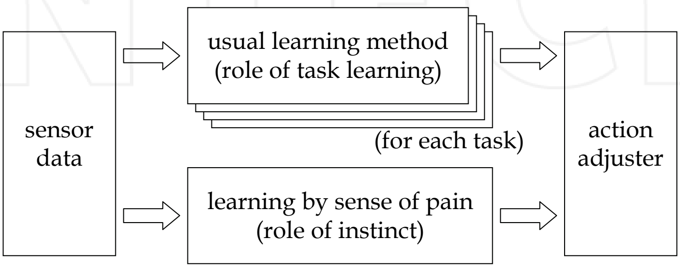


Fig. 1. Component of proposed system

2.2 Experimental robot

We use small-sized humanoid robot as application. We show the robot in fig. 2. This robot is about 50cm tall and has 23 degrees of freedom and various sensors. Especially, each servomotor has sensors about a position, a load and its temperature. This robot has processor unit on which UNIX OS runs internally. I show the detail in table 1.

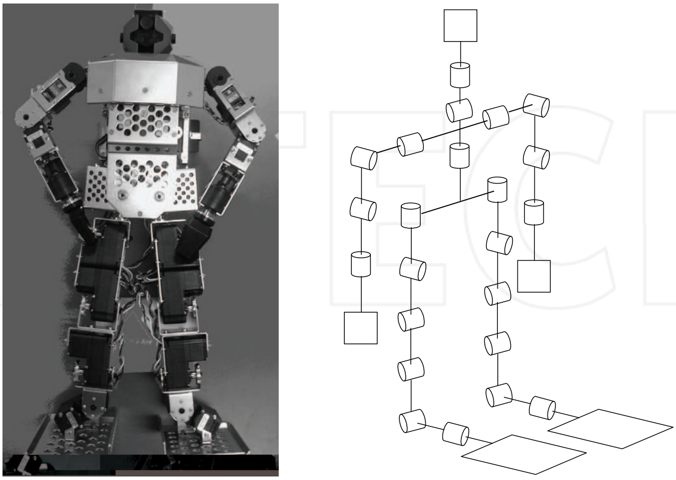


Fig. 2. The photo of the robot and the structure of the robot

Tall / Weight	50cm / 3.7kg
Degree of freedom	23 axes
sensing	single-degree-of-freedom gyro
	three-degrees-of-freedom gravity
	CMOS color camera
	2 x monaural microphone
sensing (each servomotor)	angle
	torque
	temperature
other interface	2 x LED (3 color)
	speaker
	wireless LAN (IEEE 802.11b)

Table 1. The specification of experimental robot

2.3 Definition of pain on the robot

We define pain on the robot based on its sensor values. We consider that a robot has N kinds of sensors. For each sensor, we define normal value and abnormal value. And if there is over one sensor which has abnormal value, we define a robot feels pain. In this section, we use a torque sensor which can detect a load on a servomotor and define pain on experimental robot. Using L_i which is the value of i-th torque sensor, we define the state which the sensor has abnormal value as $L_i > L_i'$. By this, we define $pain_i$ as follows; value of 1 means robot feels pain on place of i-th sensor, value of 0 means robot doesn't feel pain on it.

$$L_i > L_i' \rightarrow pain_i = 1$$
$$L_i \leq L_i' \rightarrow pain_i = 0 \quad (1)$$

To determine L_i', we examine pre-experiment which made the robot move randomly, collect data of values of L_i and calculate average μ_i and deviation σ_i. By these values, we define L_i' as follows

$$L_i' = \mu_i + 3\sigma_i \quad (2)$$

Using $pain_i$, we define $pain$ as follow.

$$pain = \bigcup_i pain_i \quad (3)$$

2.4 Learning a given task and avoiding fatal injury using RL as learning method

We give the robot a task which is to select action human want the robot to do. Here we decide desired action as follows.

learning task :

a. If the robot detects load on arm in back and forth, desired action is to move its arm back and forth.

b. If the robot detects load on arm in right and left, desired action is to move its arm right and left.

At the same time, we expect that the robot learn by sense of pain and avoiding fatal injury.

learning by sense of pain :

c. If the robot detects abnormal load on arm, desired action is avoidance action.

We use reinforcement learning (Sutton & Barto, 1998) to realize these learning. We adopt Q learning as a learning method (eq. 4). This way, we applied same equation to both learning.

$$Q_{\#}(s_{t\,\#}, a_{t\,\#}) \leftarrow Q_{\#}(s_{t\,\#}, a_{t\,\#}) + \alpha_{\#}\left[r_{t+1\,\#} - Q_{\#}(s_{t\,\#}, a_{t\,\#})\right] \quad (4)$$

Here, $S_{t\,\#}$ is a current state, $a_{t\,\#}$ is a selected action, $r_{t\,\#}$ is a reward obtained by the action. Subscript symbol "*t*" is discrete time step, and "#" is whether "*pain*" or "*task*". For example, $a_{t\,pain}$ is action *a* at time *t* considering at learning based on sense of pain. And we adopt Softmax Action Selection defined by eq. 5 to select action *a*.

$$\pi_{\#}(s_{t\,\#}, a_{t\,\#}) = \frac{e^{Q_{\#}(s_{t\,\#}, a_{t\,\#})/\tau_{\#}}}{\sum_{a_{\#}} e^{Q_{\#}(s_{t\,\#}, a_{t\,\#})/\tau_{\#}}} \quad (5)$$

Here, $\tau_{\#}$ is a positive constant called temperature. Other is same meaning as upper case.

Next, we define states and actions to use reinforcement learning. For learning task, we define these as table 2. And for learning by sense of pain we define these as table 3.

We use plural learning which is for task and is based on sense of pain, so plural actions will be selected. To make the robot move actually, one action must be selected. We consider action adjuster to select an action the robot will act. On this mechanism, an action which has maximum value in $\pi_{\#}$ at "#" is selected. We show the outline of action adjuster in fig. 3.

Using proposed system, we realize to learn given task and to learn avoiding fatal injury at the same time. At the experiment, the learning for given task is tried 100 times in each state.

$s_{0\,task}$	load detection in back and forth
$s_{1\,task}$	load detection in right and left

(a) state

$a_{0\,task}$	move arm back and forth
$a_{1\,task}$	move arm right and left

(b) action

Table 2. States and actions for learning task

$s_{0\,pain}$	$pain = 0$ (robot doesn't feels pain)
$s_{1\,pain}$	$pain = 1$ (robot feels pain)

(c) state

$a_{0\,pain}$	continue a present action (no action for avoidance)
$a_{1\,pain}$	return the servo to an initial position (avoidance action)

(d) action

Table 3. States and actions for learning by sense of pain

Learning for given task	positive reward	5
	negative reward	-3
	α_{task}	0.1
	τ_{task}	3
Learning by sense of pain	reward if return the servo to an initial position	-1
	reward if servo become to be abnormal state	-100
	α_{pain}	0.5
	τ_{pain}	0.5

Table 4. The parameter for the experiment

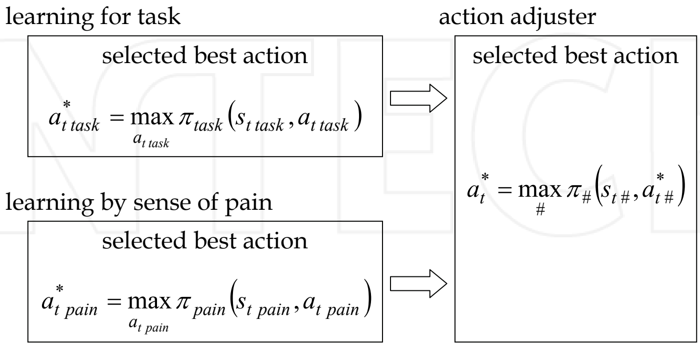
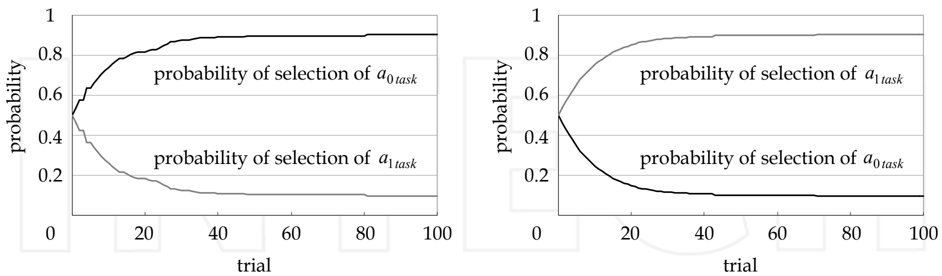


Fig. 3. Outline of action adjuster

And the learning by sense of pain is done once every 500msec. Other parameter is shown in table 4.

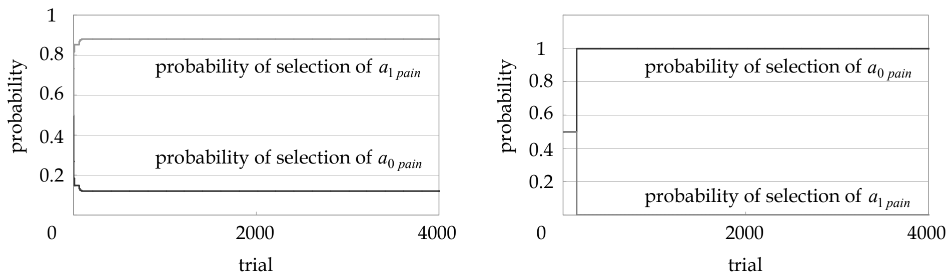
2.5 Result

We show results in fig. 4 and fig. 5 and table 5. The transition of action selection probability in learning for given task is shown in fig. 4. It shows that the selection probability of the best action was rising with progress of the trial time. The transition of action selection probability in learning by sense of pain is shown in fig. 5. It shows that the learning was done and the robot got the ability of avoiding fatal injury.



(a) The case in the state $s_{0\,task}$ (b) The case in the state $s_{1\,task}$

Fig. 4. The transition of probability of action selection in learning for given task



(a) The case in the state $s_{0\,pain}$ (b) The case in the state $s_{1\,pain}$

Fig. 5. The transition of probability of action selection in learning by sense of pain

	$a_{0\,pain}$ (no avoidance action)		$a_{1\,pain}$
	$a_{0\,task}$	$a_{1\,task}$	
$s_{0\,task}$ and $s_{0\,pain}$	99.67%	3.33%	0%
$s_{1\,task}$ and $s_{0\,pain}$	6.67%	93.33%	0%
$s_{0\,task}$ and $s_{1\,pain}$	0%	0%	100%
$s_{1\,task}$ and $s_{1\,pain}$	0%	0%	100%

Table 5. The result of action selection after 120 times learning

After learning, we experimented to confirm the result of the learning. We give the robot given task at 120times including the case caused abnormal state. The result of this confirmation is shown in table 5.

3. Creation of evaluation functions with energy

In this section, we show a study about creation of evaluation functions by using energy of robot as universal evaluation (Kurashige et al., 2002). How to evaluate robot's action changes in different contexts, in different tasks or environment. For usual learning, human must design evaluation functions for concrete task or environment robot will face. We have tried to create proper evaluations along concrete task and environment by universal evaluation. We proposed a method based on motivation to drive action on life. Here, we show motivation model we proposed and experiment on computer simulation.

3.1 Proposed concept "motivation model" based on life

Life has desire to feel satisfaction, especially when they feel insufficiency. They act for the aim of being satisfied with their status. The force that causes life to take action by desire is called "motive" in the field of psychology (Atkinson et al., 1999). Motive is classified roughly into two types; one is called basic motive and another is called derived motive. Basic motive is considered as motive which life has innately and which is equally among life or a species. Derived motive is considered as motive which is acquired through individual experience and is different on each other. And derived motive is considered as one gained based on basic motive. But this acquisition process isn't fixed on yet.

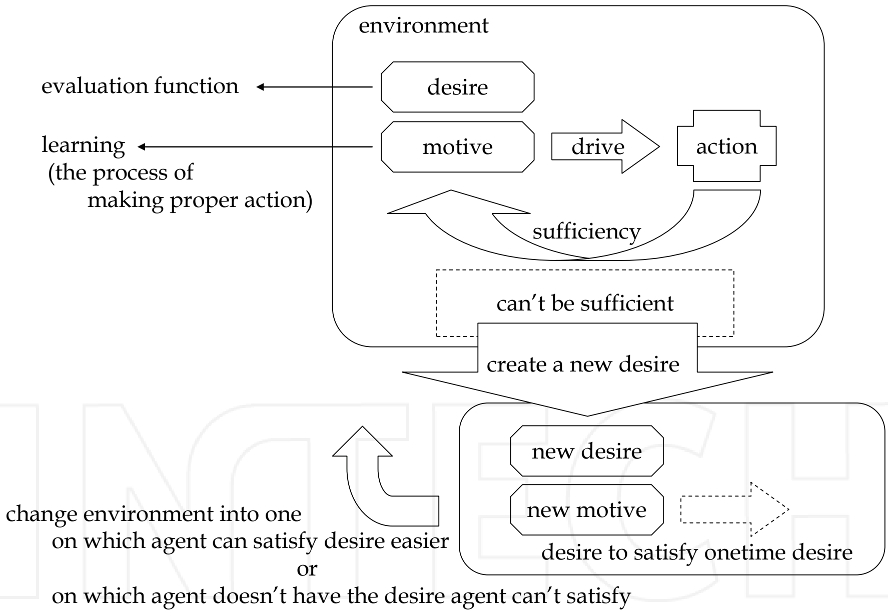


Fig. 6. Proposed concept named motivation model

Here, we thought of basic idea based on this knowledge as follows. Desire on agent, which is robot or etc., is a direction or index of satisfaction. And motive on agent is the process which agent creates or selects action to satisfy its desire. We consider desire as evaluation function and motive as learning process. If agent can learn and satisfy its desire, there is no problem. If it is hard or impossible to make proper action for satisfaction of its desire, there is problem that agent can't satisfy its desire. To solve this problem, we consider that agent creates new desire which is to satisfy one time desire. By action caused by new motive to try to satisfy corresponding desire, agent tries to change an environment into the others on which agent can satisfy its desire easier or on which agent doesn't have the desire it can't satisfy. Especially by the latter case, agent tries to avoid an environment on which agent can't satisfy its desire, and tries to learn proper action on other environment to satisfy its desire. This is outline of idea named "motivation model". We show proposed concept in fig. 6. Next, we construct concrete algorithm by motivation model.

3.2 The algorithm to generate evaluation functions based on motivation model

We propose an algorithm to generate evaluation functions based on motivation model. Here, we construct the algorithm by modifying reinforcement learning (Sutton & Barto, 1998). The outline of proposed algorithm is shown in fig. 7. Evaluation μ_i is i-th evaluation and produces reward which is decided according to an agent's state. Knowledge space is the space composed by μ_i, s, a and is made by learning. If agent can get high reward and be sufficient by learning, there is no problem. If it is hard or impossible to get high reward, we think there is problem and try to make agent create new evaluation to solve the problem. We explain when agent creates new evaluation, and next explain the algorithm how to create it.

We define the timing to create new evaluation by a shape of knowledge space. At first, we define knowledge space corresponding to i-th evaluation as $M_i : \mu_i \times (s,a)$ and show outline in fig. 8. We classify this under four typical types to explain a concept of creation of new

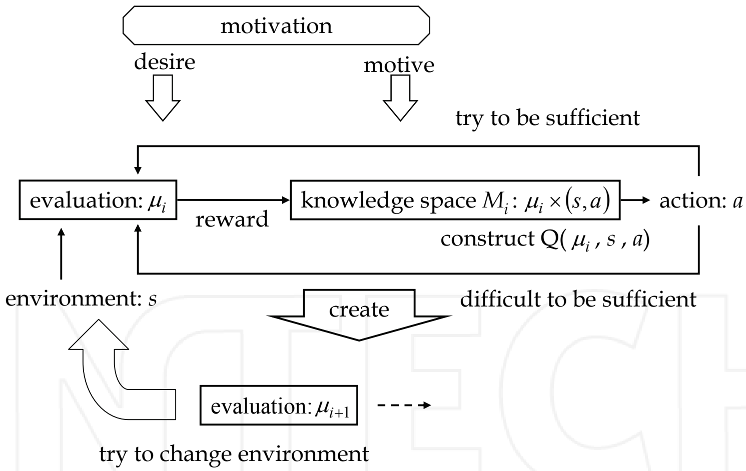


Fig. 7. The outline of proposed algorithm

evaluation as shown in fig. 9. In the case of fig. 9(a), both an agent's action and a state of environment agent faces influence an evaluation score, so they have the strong relationship. In the case of fig. 9(b) and (c), the relationship between an agent's action and a state is weaker than in the case of fig. 9(a). Evaluation score depends only on a state of environment in the case of fig. 9(b) and depends only on an agent's action in the case of fig. 9(c). Lastly, there is no relationship between an agent's action and a state of environment in the case of fig. 9(d). Here, we focus on cases of fig. 9(a) and (b). In these cases, an agent can't control its evaluation score only by its action. The evaluation score depends on a state of environment. So we consider that an agent creates new evaluation in these cases, and by created action under new evaluation an agent tries to be in a state which has possibility to get high reward. To judge whether new evaluation must be created or not, we use joint probability distribution $P(\mu_i, s, a)$. By this, we can calculate marginal probability distribution $g(\mu_i, a)$ as shown in eq. 5.

$$g(\mu_i, a) = \sum_s P(\mu_i, s, a) \tag{5}$$

At this time, we can calculate existence probability *p* on r_{μ_i} as follows.

$$p = g(r_{\mu_i}, a_t) \tag{6}$$

Here, r_{μ_i} is reward for an action a_t under a state s_t according to evaluation μ_i. Using existence probability *p*, we define the probability of generation of evaluation function as *1-p*.

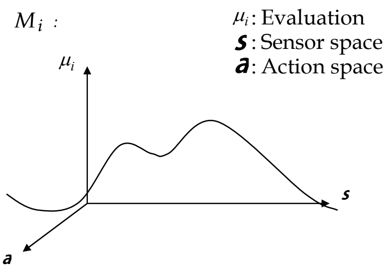
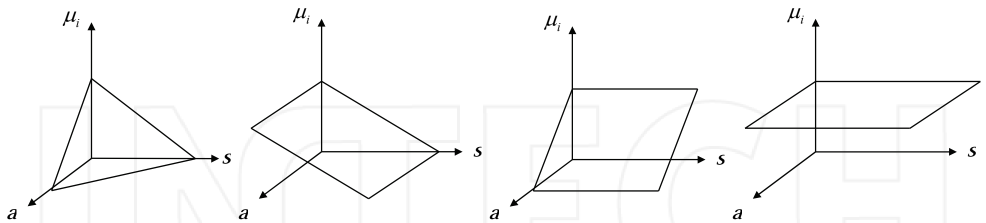


Fig. 8. The outline of knowledge space



(a) strong relationship (b) insensitive to action (c) insensitive to env. (d) weak relationship

Fig. 9. Four typical types of knowledge space

Next, we explain how to create new evaluation. We think that an agent tries to be in a state which agent can get higher reward by an action derived by new evaluation. So we define new evaluation μ_j with a state *s*. On knowledge space M_i , we can calculate marginal probability distribution $f(\mu_i, s)$ as shown in eq. 7.

$$f(\mu_i, s) = \sum_a P(\mu_i, s, a) \tag{7}$$

An action a_t at time t under evaluation μ_j is action to make profitable environment under evaluation μ_i. So we define μ_j using a state s_{t+1} derived by a_t as follows. And we show the concept of how to create new evaluation in fig. 10.

$$\mu_j = \mu_j(s_{t+1}) = E(\mu_i(s_{t+1})) = \sum_{r_{\mu_i}} r_{\mu_i} \cdot f(r_{\mu_i}, s_{t+1}) \tag{8}$$

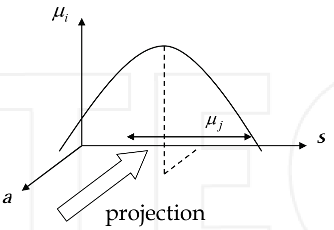


Fig. 10. Concept of how to create new evaluation

Finally, we explain how to select action using these evaluation functions. On each evaluation μ_i, an action a_i which can take $\mu_{i\,\max}$ is selected. Here, $\mu_{i\,\max}$ is maximum value of evaluation μ_i. The number of candidate actions is equal to the number of evaluation functions. We define probability of selection for each action a_i as eq. 9. An agent decides an action based on this probability of selection.

$$q(a_i) = \frac{\mu_{i\,\max}}{\sum_i \mu_{i\,\max}} \tag{9}$$

3.3 Burden-carrying task

We applied proposed algorithm to burden-carrying task. The object environment is shown in fig. 11. The task is to carry burdens from loading station to unloading station. The robot which is the agent at this task can get energy β per one burden as a reward for work. In the environment, there are several kinds of hindrances. They are walls and burdens. Walls bar robot's way. If the robot puts burden down on any place except unloading station, it will become hindrance.

For this task and environment, the robot can takes several actions: Load, Unload, Forward, Left, Right and Stop. The robot needs energy to execute each action whether the robot can do or not. So if the robot fails to execute an action, for example the robot tries to go through a wall, the robot loses same amount of energy when the robot succeeds to take that action and a state of the robot doesn't change. In this task, we set energy to take any action as α. Actions the robot can take and perceptions the robot can use are as follows.

Load	get a burden in front of the robot
Unload	put a burden down in front of the robot
Forward	take a step forward
Left	turn to the left
Right	turn to the right
Stop	stop

Table 6. Actions the robot can take

$state_{direc}$	state around the robot (direc : forward, right, left, back)
$state_{burden}$	state whether the robot has burden or not
$(x, y, direc)$	current location and direction
$\Delta energy$	change of energy

Table 7. Perceptions the robot can use

We define initial evaluation function by using the change of energy of the robot as eq. 10. This is basic motive at this task. And it plays the role of universal evaluation because of the definition which is independent of environment.

$$\mu_1 = \Delta energy = \begin{cases} -\alpha \\ \beta \end{cases} \tag{10}$$

Here, α is energy to take an action and β is a reward for work when the robot can get at unloading station.

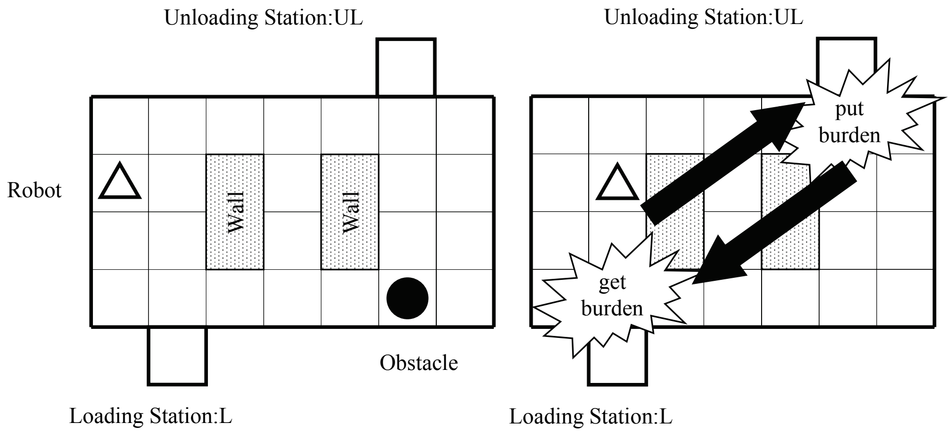


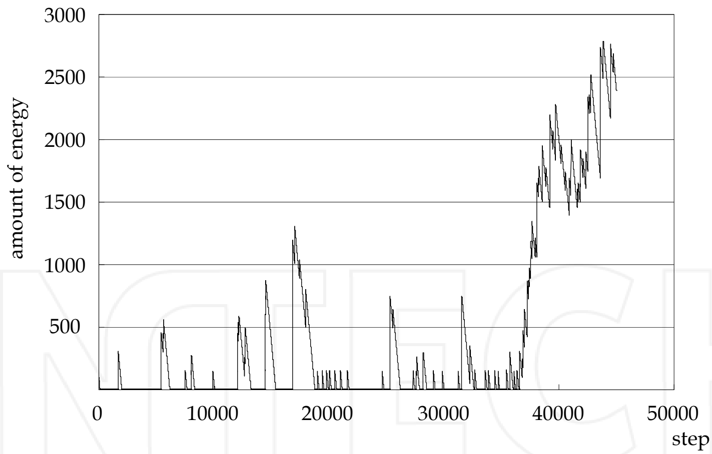
Fig. 11. Outline of load-carrying task

3.4 Results of computer simulation

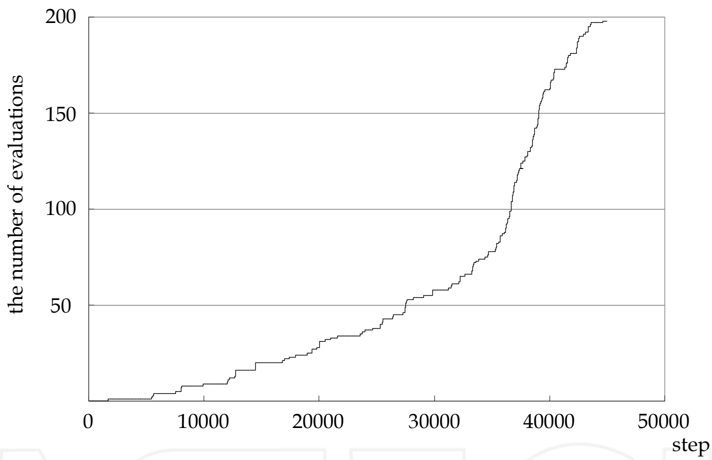
We experiment burden-carrying task on computer simulation. The robot has energy φ as initial energy. If energy of the robot drops to zero, we give the robot energy γ in the midst of learning as recharging. The number of burden which the robot can carry at once is expressed as χ. We show the parameter of simulation in table 8.

α	-1
β	150
φ	100
γ	10
χ	10

Table 8. The parameter of simulation



(a) Transition of the amount of energy robot keeps



(b) Transition of the number of evaluation

Fig. 12. Results through learning

The results of simulation under this condition are shown in fig. 12. Figure 12(a) represents transition of amount of energy on the robot. Figure 12(b) represents the number of evaluation which the robot creates with proposed algorithm.

In the first part of fig. 12(a), the amount of energy which the robot kept was low. We consider it was occurred because the robot took actions randomly in this phase which is early phase of learning. And increasing the number of evaluation, we can see the amount of energy which robot kept was rising.

And we show the existence probability of the robot on the environment from 40000 step to 50000 step in fig. 13. This shows the robot went round between loading station and unloading station.

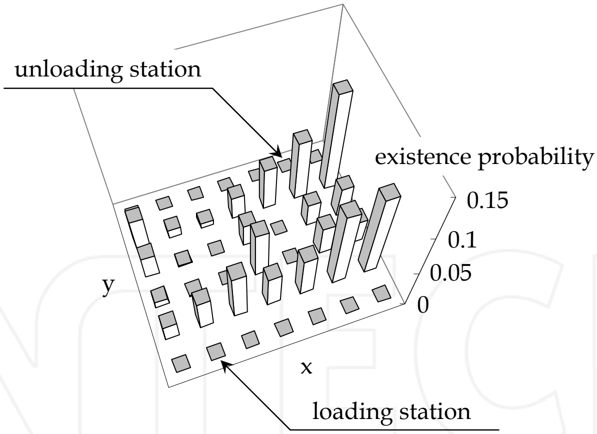


Fig. 13. Existence probability of the robot on learning between 4000 step and 5000 step

4. Conclusion

Our goal is to realize a system which keeps adapting various tasks and environment with universal mechanism which is independent of concrete tasks and environment. In this chapter, we proposed the concept of universal evaluation as a kind of universal mechanism. Here, we showed two experiments as instances. One is the study using sense of pain as universal evaluation. With this universal evaluation, the robot could avoid being injured by unexpected load. Another is the study to create evaluation functions for concrete environment by universal evaluation. We showed recursive algorithm to create evaluation functions by existing evaluation functions. And we used evaluation about energy on robot as the beginning and universal evaluation. We showed the robot could take more proper action as it created evaluation functions by proposed algorithm.
As the future works, we try to find and propose better universal mechanisms. For example, we consider that a rule how to interact environment can be used as a universal mechanism. As the first step of this, we have tried to create evaluation functions for concrete task and environment with an interaction rule which is defined by variance of sensor data (Kurashige, 2007). By importing a concept of universal mechanism into learning method, we try to divide between how to design a robot and how to use a robot, and we try to realize a system which can get necessary knowledge whenever it is necessary only with an operation of its information. We think that is next step for autonomy.

5. References

Asada, M.; MacDorman, K. F.; Ishiguro, H. & Kuniyoshi, Y. (2001). Cognitive Developmental Robotics As a New Paradigm for the Design of Humanoid Robots, *Robotics and Autonomous System*, 37, (2001), pp. 185-193

Atkinson, R. L.; Smith, E. E.; Bem, D. J. Nolen-Hoeksema, S. (1999). *Hilgard's introduction to psychology*, Thirteenth edition, Wadsworth Publishing

Brooks, R. A. & Stein, L. A. (1994). Building Brains for Bodies, *Autonomous Robots*, 1, 1, (November, 1994), pp.7-25

Goodwin, R. (1994). Reasoning about when to start Acting, *Proceedings of the 2nd International Conference on Artificial Intelligence Planning Systems*, pp.86-91, Chicago, IL, USA, May, 1994

Hasegawa, Y; Yokoe, K; Kawai, Y. & Fukuda, T. (2004). GPR-based Adaptive Sensing -GPR Manipulation According to Terrain Configurations, Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.3021-3026, Sendai, Japan, October, 2004

Jordan, I. M. (1992). Forward models: Supervised learning with a distal teacher, *Cognitive Science*, 16, (1992), pp.307-354

Kawamoto, H.; Lee, S.; Kanbe, S. & Sankai, Y. (2003). Power Assist Method for HAL-3 using EMG-based Feedback Controller, *Proceedings of International Conference on Systems, Man and Cybernetics*, pp.1648-1653, Washington DC, USA, October, 2003

Kobayashi, E.; Masamune, K.; Sakuma, I.; Dohi, T. & Hashimoto, D. (1999). A New Safe Laparoscopic Manipulator System with a Five-Bar Linkage Mechanism and an Optical Zoom, *Computer Aided Surgery*, 4, 4, pp.182-192

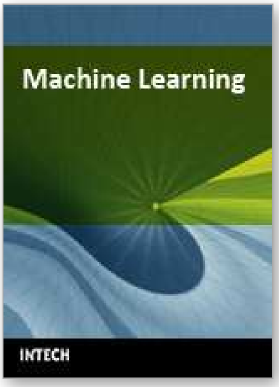
Kurashige, K. (2007). A simple rule how to make a reward for learning with human interaction., *Proceedings of the 2007 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp.202-205, Jacksonville, FL, USA, June, 2007

Kurashige, K.; Aramaki, S. & Fukuda, T. (2002), Proposal of the motion planning with Motivation Model, *Proceedings of Fuzzy, Artificial Intelligence, Neural Networks and Computational Intelligence*, pp. 467-472, Saga, Japan, November, 2002

Kurashige, K. & Onoue, Y. (2007). The robot learning by using "sense of pain", *Proceedings of International Symposium on Humanized Systems 2007*, pp.1-4, Muroran, Japan, September, 2007

Nolfi, S. & Parisi, D. (1997). Learning to adapt to changing environments in evolving neural networks, *Adaptive Behavior*, 5, 1, (1997), pp.75-98

Sutton, R. S.; Barto, A. G. (1998). *Reinforcement Learning An Introduction*, The MIT Press, Cambridge



Machine Learning
Edited by Abdelhamid Mellouk and Abdennacer Chebira

ISBN 978-953-7619-56-1
Hard cover, 450 pages
Publisher InTech
Published online 01, January, 2009
Published in print edition January, 2009

Machine Learning can be defined in various ways related to a scientific domain concerned with the design and development of theoretical and implementation tools that allow building systems with some Human Like intelligent behavior. Machine learning addresses more specifically the ability to improve automatically through experience.