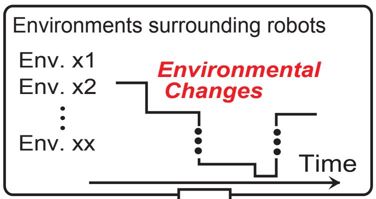
Learning Algorithms for Robots Behaving Flexibly in Dynamic Environments

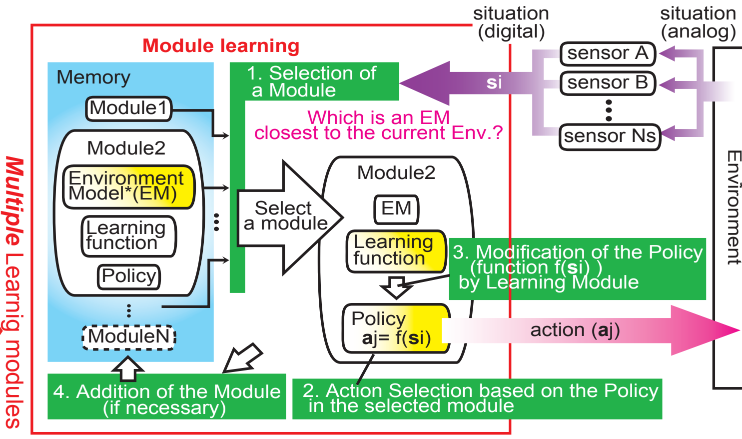
Masahiro Ono, Hiroshi Ando, Mamoru Sasaki and Atsushi Iwata
Graduate school of Advanced Sciences of Matter, Hiroshima University

Introduction



Conventional Robots:

Difficulty in Adaption to Environmental Changes(E.C.)
Reason: ***only one*** Learning module for all environments. Long time of learning prevent from following E.C. .



*: An Environment Model has the feature information of an environment

A Problem of module learning: Total memory capacity becomes enormous (ex. GB order in Robo Cup Soccer).

Reason:
1. **An EM for memory capacity is enormous(P1).**
2. **One module is required for one Env(P2).**
   (Larger num. of Env. s makes difficult to store modules)

Target of research: proposition of module learning solving P1 and P2

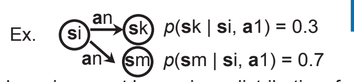
P1 ⇨ Module learning using EMs with small data size

P2 ⇨ Module learning with fast learning and compact storage

Module learning algorithm *using environment models with small data size*

Typical environment model(EM) = a set of probabilities

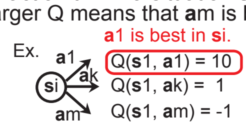
- p(**sk** | **si**, **an**) means a probabilities that an action **an** in a situation **si** results in a situation **sk**.



- Each environment has unique distributionof p.
- Problem : Enormous memory capacity is required.

Proposed EM = a set of Q–functions

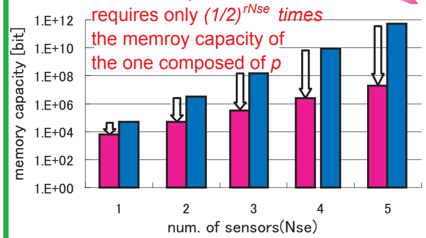
- Q(**si**, **am**) means the ***quality*** of an action **am** in a situation **si** (larger Q means that **am** is better)



- Each environment has unique distribution of Q
- Memory capacity for an EM composed of Q is ***much less*** than that of p as the number of sensors is lager.

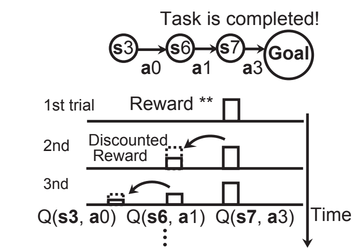
■ An EM composed of Q $Na \cdot 2^{r \cdot N_{se}} \cdot m$

■ p : $Na \cdot (2^{r \cdot N_{se}})^2 \cdot m$



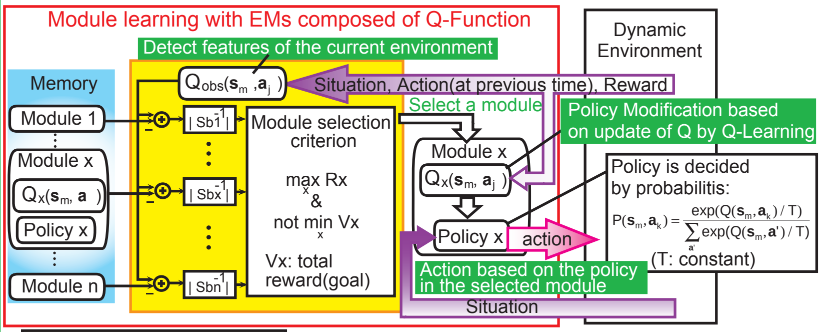
Na: the number of actions r: sensor resolution [bit] Nse: the number of sensors
m: unit data size of p and Q[bit]
(Na=100, r = 3[bits], m = 8[bits])

Update Method of Q: Q-Learninig (a type of Reinforcement Learning)



** : Reward is given to robots when robots achieve goal.

Proposed algorithm schematic diagram

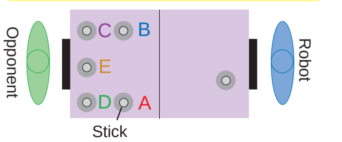


$$P(s_m, a_k) = \frac{\exp(Q(s_m, a_k)/T)}{\sum_{a'} \exp(Q(s_m, a')/T)}$$

Numerical Simulation

In the games, the opponent changes its stick position to defense from A to E by 1 set (20 points). (Robot's position is fixed.)

Opponent's characteristics = stick positions

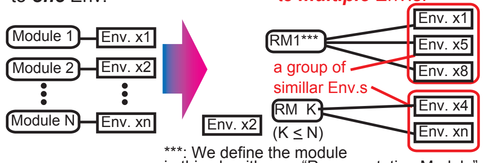


Assumption :
- The robot has already had the strategies for A, B, C and D by learning before the experiment. (called S_A, S_B, S_C and S_D, respectively)
- The opponent has only a simple fixed strategy.
- The performance of the robot is the same as that of the opponent.

Module learning algorithm *with fast learning and compact storage of modules*

Typical Modular Learning
Each module is assigned to ***one*** Env.

Proposed idea
Each module is assigned to ***multiple*** Env.s.



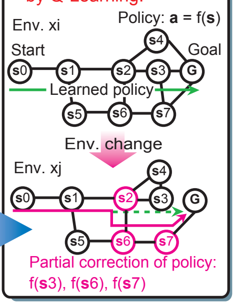
***: We define the module in this algorithm as "Representation Module"

The number of modules ***does not*** become ***very large*** even if the number of Env.s becomes large (***compact storage of modules***)

Problem:
Fast re-learning of Policy is required for fast adaptation to environmental changes

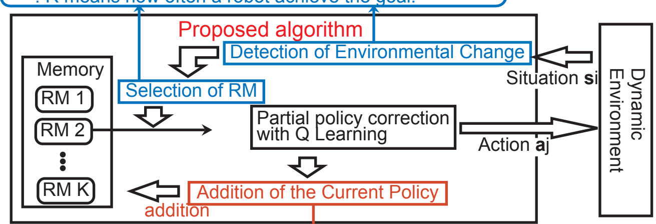
Partial policy correction with Q Learning

Policies required for re-learning are updated by Q-Learning.



Proposed algorithm schematic diagram

If R**** > Rg then RM is selected in order of index of RM.
(Example: If current RM is RMi, then next one is RMi+1.)
****: R means how often a robot achieve the goal.

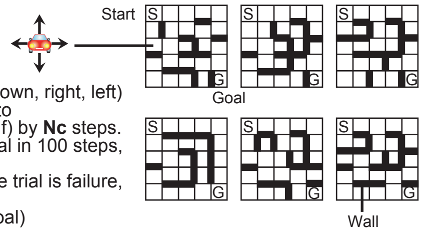


Correlation between the current policy and RMi: ri (i = 1, ... N) is calculated.
If **ri > rth** (constant value) then **Elimination** else **Addition.**

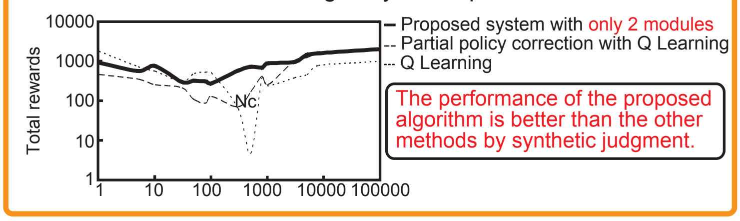
Numerical Simulation

Maze Problem

Restrictions:
- Situation:25 (5x5), Action:4(up, down, right, left)
- Environment changes randomly to
- one of six environments (Env.a - f) by **Nc** steps.
- If the robot does not reach to Goal in 100 steps, the trial is failure.
- If the robot reaches to Goal or the trial is failure, the robot is returned to Start.
- Reward: 1(Goal) 0 (except for Goal)



■ Result of the simulation on condition that the environment changes by Nc steps



The performance of the proposed algorithm is better than the other methods by synthetic judgment.

Conclusions

We proposed two module learning algorithm for solving a problem that the conventional module learning needs enormous memory capacity.

1. A module learning algorithm using environment models with small data size
2. A module learning algorithm with fast learning and compact storage of modules

They are confirmed by the simulation experimentsthat both algorithms are robust to environmental changes and use memory more effectively.
As next step, we'll incorporate EM used in the first algorithm into detection of environmental changeand selection of RM in the second algorithm.