The Seventeenth International Symposium on Artificial Life and Robotics 2012 (AROB 17th '12),
B-Con Plaza, Beppu, Oita, Japan, January 19-21, 2012

Control method for a redundant robot using stored instances

Yuya Okadome[1], Yutaka Nakamura[1], and Hiroshi Ishiguro[1]

[1] Department of Systems Innovation, Graduate School of Engineering Science, Osaka University, 3-1 Machikaneyama, Toyonaka, Osaka, Japan. okadome.yuya@irl.sys.es.osaka-u.ac.jp

Abstract: A robot operating in a real environment, as opposed to industrial robot working in factory, must have the exibility to cope with disturbances and irregular factors. In this paper, we propose a control method where the control signal is selected from past experiences (stored instances) of a similar situation, according to the evaluation of each instance. We apply our method to the control of a robot with complicated structure, driven by several elastic actuators. Experimental results show that the control of a robot with many DOFs can be achieved by the proposed method.

Keywords: adaptive control, k-nearest neighbor, redundant robot

1 INTRODUCTION

In order for a robot to work in a real environment, the robot has to have the ability to cope with various kinds of situations. A robot with many DOFs is advantageous since it can generate various behaviors, and thus it is expected to have the ability to adapt to various situations. A learning method such as system identi cation or reinforcement learning can be applicable to the control of such a robot; however, this is di cult due to its complexity especially when its dynamics are in uenced by various environmental factors. For example, system identi cation methods using piece-wise linear functions [4][7] to approximate the dynamics of the controlled target have been successfully applied to the control of a robot, but a huge number of pieces becomes necessary to approximate such complex dynamics.

Autonomous learning methods such as reinforcement learning might also be applicable to the control of this type of robot, in principle (e.g. Actor-Critic[1][5] algorithm or Q-Learning[3][6]), but the application becomes di cult in case of controlling a robot with many DOFs. One of the most di cult problems is the increase in the number of parameters to be learned. In order to determine a large number of parameters, it is necessary to obtain a huge number of data (i.e. the curse of dimensionality). Besides, the amount of available data does not increase even though the dimensionality of the dynamics of the target system does increase, resulting in longer operating time and more troubles.

In this paper, we propose a control method using stored instances. In this method, a certain number of instances in the database, similar to the current state, are evaluated based on experience, and one of the instances among them is selected according to this evaluation. The control signal is determined from the selected instance. Since the control signal is determined from the database directly, an explicit modeling of the control target is not necessary. That is, this method is a kind of non-parametric method and uses the knowledge of the system implicitly included in the dataset to determine the output. Note that further knowledge about the system cannot be obtained from the database even if a system identi cation method is used.

2 CONTROL METHOD USING STORED INSTANCES

In this research, we propose a control method where the control signal is directly determined from the stored instances. Since this method does not employ a system identi cation nor a learning method, it is advantageous if the control target has a redundant structure which makes the learning of the system di cult.

2.1 Gathering Instances

To obtain instances for determining the control signal, the robot is controlled by randomly generated control signals at rst. During this procedure, the controller observes the current state of the robot $\boldsymbol{x}(t)$ and output the control signal $\boldsymbol{u}(t)$ at random. A pair consisting of the state and the control signal $\boldsymbol{d}_i = \{\boldsymbol{x}(t=i), \boldsymbol{u}(t=i)\}$ is called "an instance" and added to the database.

Each instance in the database D is denoted by $\boldsymbol{d}_i$ for the understandability of the notation and the database is de ned as a set of instances:

$$D = \{\boldsymbol{d}_i |\ i = 1, 2, \ldots, N\} \quad (1)$$

where N is the number of instances. Note that $\boldsymbol{d}_{i+1}$ is the instance composed by subsequent state and action of the i-th instance $\boldsymbol{d}_i$.

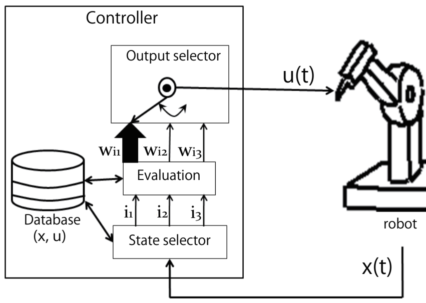


Fig. 1. Block diagram of proposed method

2.2 Control Rule

Fig. 1 shows a block diagram of the proposed method. After the controller observes the state $\boldsymbol{x}(t)$, nearest K neighbors in the database are selected by the "state selector". Indexes of nearest K instances constitute the set "KNN".

We employ a simple evaluation function, where the weight of each instance is determined from the distance between the trajectory starting from each instance in the KNN and the given goal. Thus, the weight of the i-th instance ($i \in KNN$) is de ned as:

$$w_i \quad = \quad \min_{j<p} \{||\boldsymbol{G} \quad \boldsymbol{d}_{i+j}||_2\} \tag{2}$$

where p is the length of the trajectory to be considered in the evaluation, and $\boldsymbol{G}$ denotes the goal state. $||\boldsymbol{G} \quad \boldsymbol{d}_{i+j}||_2$ is the Euclidean distance from the state $\boldsymbol{x}_{i+j}$ of the $i+j$-th instance.

After the evaluation for all instances in KNN, one of the KNN instances is selected by the control signal selector. The probability to select the i-th instance is

$$P(i) = \begin{cases} \frac{\exp(w_i T)}{\sum_{j=1}^{K} \exp(w_j T)} & i \in KNN \\ 0 & \text{otherwise} \end{cases}. \tag{3}$$

T is a parameter called inverse temperature which modi es the probability. When T is large, the instance with maximum weight is selected almost deterministically. On the other hand, when it is small, every instance can be selected with equal probability.

3 CONTROL OF ELASTIC BINARY MANIPULATORS

We applied the proposed method to a control task (reaching task) of a redundant robot (Fig. 2). This robot is a variation of binary manipulators [2] with elastic links.

3.1 Elastic Binary Manipulators

The Elastic Binary Manipulators has several links with a truss-like structure. As seen in Fig. 2, each link (line) and joint (black point) is modeled by a spring damper without mass, and a mass-point, respectively. The control signal to the robot consists of the natural lengths of the links, and changes the equivalent point (posture). In the simulation, the natural lengths of the 12 links represented in the gure by dashed lines can be changed according to the control signal, and the others are xed. The length of each variable link can have a larger or smaller value and the former is 1.5 times longer than the latter.

The force applied to the i-th mass-point is de ned by:

$$\boldsymbol{F}_i \quad = \quad \sum_{j \in C_i} \left\{ \quad k(l_{ij} \quad \bar{l}_{ij}) \frac{\boldsymbol{s}_i \quad \boldsymbol{s}_j}{||\boldsymbol{s}_i \quad \boldsymbol{s}_j||_2} \quad D(\boldsymbol{v}_i \quad \boldsymbol{v}_j) \right\}, \tag{4}$$

$\boldsymbol{s}_\iota$ and $\boldsymbol{v}_\iota$ denote the position and the velocity of the ι-th joint. l_{ij} and $\bar{l}_{ij}$ are the current length and the natural length of the link connecting the i-th and the j-th joints. k and D are the spring and the damping constants.

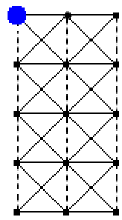


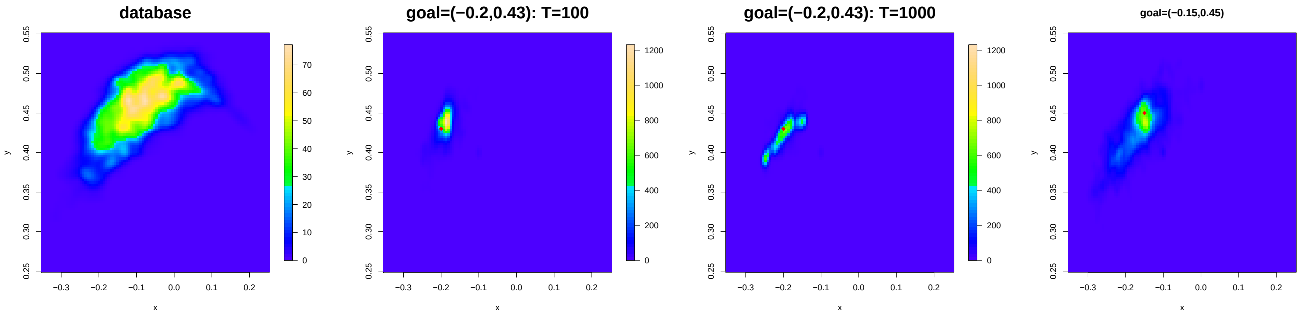
Fig. 2. Structure of the EBM

3.2 Reaching task

The purpose of the reaching task is to move the end-e ector (blue point) to the goal position. During the instance-gathering phase (500,000 time steps), 6 links are selected randomly and their natural lengths are altered (small to large/ large to small) every 1,000 time steps. To generate meaningful motions, the control signal is maintained over a certain time period (1,000 time steps).

We conducted experiments with the goal G = (0.2, 0.43). K was set to 5 and T was set to 100 and 1,000. These values were determined by trial and error.

Fig. 3(a) shows the density of state-instances stored in the database visualized by a kernel density estimation method. Fig. 3(b) and Fig. 3(c) show the density of the state while performing the reaching task. The yellow-colored area indicates that the end-e ector stays



(a) (b) (c) (d) with broken link

Fig. 3. Result of reaching task: Elastic Binary Manipulators

in these areas for a long time. The red point in each gure represents the goal position of the reaching task, and it can be seen that the density concentrates to the area close to the goal in both cases. These results show that our control method allows the robot to achieve the reaching task. The performance of the case $T = 1000$ is worse than that of $T = 100$. This suggests that stochastic choice of control signal makes the robot escape from the local optima.

We conducted an additional experiment to investigate whether our method can exploit the redundancy of the robot (Fig. 3(d)). Even when a controllable link is broken (the natural length is xed), the proposed method allows the robot to achieve the task if there is a link which is able to compensate the broken link.

4 CONTROL OF HUMAN-LIKE ROBOTIC ARM

We applied the proposed method to a control task of a human-like robotic arm (Fig. 4). We conducted a reaching task by this robot.

4.1 Human-Like Robotic Arm

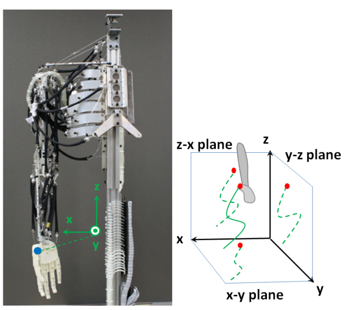


Fig. 4. Human-Like Robotic Arm

The human-like robotic arm consists of 6 links and has a high degree of freedom (9-DOFs). More than 30 pneumatic actuators (arti cial muscles) are connected to these links, and are controlled by 26 air- ow control valves. Each actuator expands (contracts) when the air pressure decreases (increases). Actuators are located at positions similar to those of human muscles. The air ow of each valve monotonically increases according to the control signal, i.e. the output of the D/A converter (output range: 0 6[V]). To control this robotic arm, it is necessary to determine the control signal consisting of 26 analog values depending on the state of the robot.

4.2 Reaching task

The purpose of the reaching task is to move the hand (represented by the blue point in Fig. 4) to the goal position. During the instance-gathering phase (1500[s]), 13 actuators are selected randomly and their input values are altered randomly in every 1[s]. We conducted two experiments with two di erent goals, $G = G1(0, 150, 140)$ and $G2(\ 40, 100, 80)$. In each reaching task, the operation time was 150[s] and the control signal is altered once in 0.5[s]. K and T were set to 10 and $\frac{2}{15}$ by trial and error.

Fig. 5(a) and Fig. 5(b) show the density of the hand position in the database. Fig. 5(c) Fig. 5(f) show the densities of the position of the hand during the reaching tasks projected on the X-Y and X-Z planes. The red point in each gure represents the goal position, and we can see that the density concentrates to the goal in both cases. These results show that our control method allows the robot to achieve the reaching task.

Reaching G2 seems worse than in the case of G1, since the density spreads widely. If there are fewer instances approaching the goal in the database, it is dif- cult to achieve the task since such a instance is often cut o from the k-NN. The reason why reaching G2 is more di cult is that there are fewer instances useful for achieving the task.

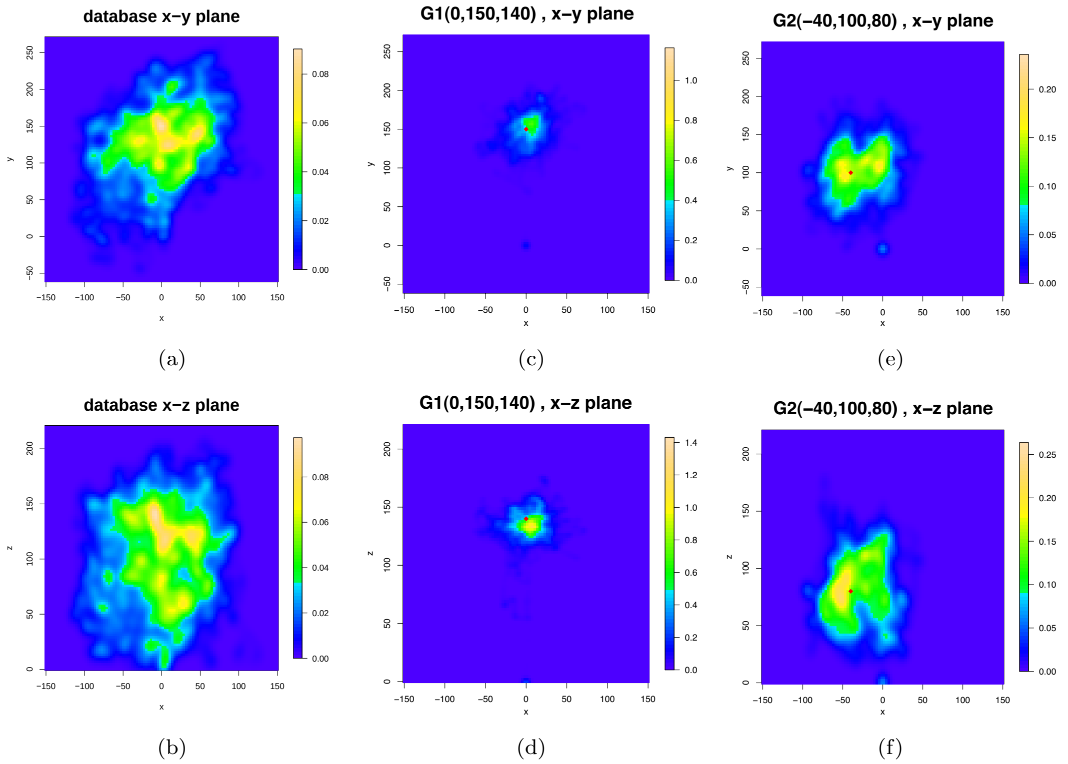


Fig. 5. Result of reaching task: Human-Like Robotic Arm

5 CONCLUSION

In this paper, we proposed a control method using stored instances which does not need an explicit modeling of the control target. We applied the proposed method to the control tasks of redundant robots and showed that the reaching task can be achieved by our method.

One of the problems of our method is performance degradation due to the bias of the distribution of the instance in the database. To overcome this, it might be useful to omit similar instances from the database or employ an importance sampling method. To develop a method for adding the instances in an online manner is also part of our future work.

REFERENCES

[1] H. Kimoto and S. Kobayashi: "An Analysis of Actor/Critic Algorithms using Eligibility Traces: Reinforcement Learning with Imperfect Value Functions" Journal of Japan Society of Arti cial Intelligence, Vol.15, No.2, pp.267-275,2000

[2] Lees, D.S. and Chirikjian, G.S.: "A Combinatorial Approach to Trajectory Planning for Binary Manipulators" Proc of ICRA, pp.2749 – 2754, 1996.

[3] M. Asada, S. Noda, S. Tawaratsumida, and K. Hosoda: "Vision-Based Reinforcement Learning for Purposive Behavior Acquisition." Proc. of ICRA, pp.146–153, 1995.

[4] M. Haruno , D.M.Wolpert, and M. Kawato: "Mosaic: Module selection and identi cation for control," Neural Computation, vol.13, no.10, pp.2201–2220, 2001

[5] Vijaymohan R. Konda and Vivek S. Borkar: "Actor-critic–type learning algorithms for Markov decision processes" SIAM J. Control Optim. Vol. 38, No. 1, pp.94–123, 1999

[6] Watkins, C. J. C. H. and Dayan, P.: "Technical Note: Q-Learning" Machine Learning 8, pp. 279–292, 1992

[7] Y. Tazaki, J. Imura: "Graph-based Model Predictive Control of a Planar Bipedal Walker" 17th International Symposium on Mathematical Theory of Networks and Systems, pp.128–133, Kyoto, Japan, July 24–28, 2006.