This is a preprint of an article published in IEEE Transactions on Human-Machine Systems.
The final authenticated version is available online at: https://doi.org/10.1109/THMS.2022.3207372

Q-Mapping: Learning User-Preferred Operation Mappings with Operation-Action Value Function

Riki Satogata, Mitsuhiko Kimoto, Yosuke Fukuchi, Kohei Okuoka, and Michita Imai

***Abstract*—User interfaces have been designed to fit typical users and their usage styles as assumed by designers. However, it is impossible to cover all the possible use cases. To address this problem, we propose Q-Mapping, which is a method for user interfaces to acquire the operation mapping, or mapping from user operations to their effects. Q-Mapping has an advantage over previous techniques in that it can acquire operation mapping interactively. The core idea of Q-Mapping is that what a user selects as an ideal action has a tendency to be the same as the action which has the highest Q value. On the basis of this concept, we defined the operation-action value function, which can be calculated from the value that a user expects to gain when a particular mapping is given in that state and is updated each time an operation occurs. We conducted a simulation experiment and a user study to investigate the Q-Mapping performance and the effects of the acquisition of interactive operation mapping. The simulation results showed that the changeability of operation mapping could be controlled by a coefficient called the balancing parameter. As for the user study, we found that Q-Mapping with a balancing parameter that decays with time was able to acquire operation mapping that was easy for users to understand. These results demonstrate the importance of balancing consistency and adaptability in the interactive acquisition of operation mapping.**

***Index Terms*—Human-device Interaction, Human-computer interface, Q-learning**

I. Introduction

WHEN humans encounter a device for the first time, how do they know how to operate it? Typically, we acquire operation methods by reading a manual or imagining what to do, and over time we become accustomed to the operation through actual use. To enable users to operate devices smoothly, interfaces are carefully designed to facilitate intuitive operations. However, sometimes the design does not suit certain users. For example, some large-handed users may find a better way to operate, and physically challenged people may find it difficult to operate the default design [1]. A typical solution is to change system settings such as "key assignment" and manually define an operation mapping between an operation and its effect, but this is often inconvenient for general users. In some cases, key assignment alone is not enough. For example, if a person with a movement disorder wants to operate, he/she needs to consider a completely new operation method. Jeebithashre et al. [2] have developed a gaze-based pointing device for people with movement disorders. Furthermore, even if a new operation method is adopted, the appropriate operation may change depending on the type and the degree of the disability and on personal preference [3].

If a system can adapt to users and provide user-preferred operation, it will lead to a significant improvement in usability. Previous studies have proposed various systems to learn user-favorite operations [4], [5]. While these systems are based on novel ideas that do not require careful design of operation mapping, the user needs to take additional steps to train the system before starting operation.

In this paper, we propose Q-Mapping, a new method for acquiring operation mappings. Q-Mapping can build operation mappings without prior beliefs about how a user operates a device, so it can be adaptively fit to each user. In addition, Q-Mapping acquires operation mappings interactively, so no additional explicit training process is required. In Q-Mapping, we define a "operation-action value function" that can be calculated from the action value function—Q value—in Q-learning [6]. Q-Mapping updates the operation-action value function when receiving user operations and uses it for operation mapping. Q-Mapping is similar to Inverse Reinforcement Learning (IRL) in that it uses the results of Reinforcement Learning (RL), but the two have different purposes: IRL estimates the intent of an expert, while Q-Mapping estimates the intent of each user.

The main contributions of this paper are as follows:

- We have shown that we can create an intelligent controller that personalizes the operation method by adapting a simple internal model based on the value of user operation.
- We test the hypothesis that user operations change from the exploration phase to the exploitation phase with such intelligent controllers and provide a guideline for building them.

In section II, we introduce related work that has tackled the inflexible nature of conventional interfaces and ways to improve their usability. In section III, we explain the proposed approach. In section IV, we introduce the task used for the two experiments conducted in this study, and we describe the method and results of the simulation experiment in section V and of the user study in section VI. In section VII, we discuss the results obtained from the two experiments and compare them with related studies. After discussing the limitations in section VIII, we conclude in section IX with a brief summary and mention of future work.

Manuscript received February 3, 2021; revised November 6, 2021.

This work was supported by JST CREST Grant Number JPMJCR19A1, Japan, and in part by JSPS KAKENHI Grant Number JP19J01290.

R. Satogata, Y. Fukuchi, M. Kimoto, K. Okuoka and M. Imai are with the Department of Information and Computer Science, Keio University 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa 223-8522, Japan (e-mail: satogata@ailab.ics.keio.ac.jp, kimoto@ailab.ics.keio.ac.jp, fukuchi@ailab.ics.keio.ac.jp, okuoka@ailab.ics.keio.ac.jp, michita@ailab.ics.keio.ac.jp)

II. Related work

A. Conventional interface design

Typical interfaces are designed based on strong assumptions about the usage situation and user characteristics. For example, the layout of the Dvorak Simplified Keyboard [7] is designed to make fingering more efficient when entering English text. However, with such interfaces, there is always a potential risk that users will find it difficult to actually use the device.

There are two main reasons such interfaces may be difficult to use. The first is the "fitting limit". Due to individual differences in the psychological measure of ease of use or physical characteristics, there is a limit to the number of users who can be satisfied with a single interface. For example, an interface designed on the basis of average hand size may be difficult for people with large / small hands to use. Furthermore, no matter how carefully the interface is tested, there still may be users who cannot operate it due to physical constraints. For example, a user with a handicapped finger may have difficulty using the keyboard. Such a user would have to find some other input method.

Studies on the control by motion [8], [9] or gaze [10], [11] have made more intuitive operations possible by utilizing the analogy of the physical world. These studies have provided options other than general-purpose input devices such as keyboards and mice. By increasing the choices of new interfaces, users can choose the interface that suits them best, and the problem of "fitting limit" can be alleviated. For example, it will be possible for people with movement disorders to use their eyes to operate devices that they could not before [2], [3]. However, these new operation methods are also designed on the basis of specific ideas, which leads to the second problem.

The second problem is "collapse of premise". This problem arises when the interface is used in a situation that is different from the use case envisioned at design time. For example, users who want to play a game on the keyboard will put their hands in a position different from the general home position. There are many cases like this, where individuals will want to use an interface differently than the general purpose. The problem of "collapse of premise" also includes the situation where the user models assumed by designers do not match the actual users. For example, an interface made for experts is difficult for beginners to use.

B. Personalized mapping

One of the solutions for these problems is to personalize the operation method. For example, "key assignment" is a personalization approach that provides a function for the user to change the operation mapping by him/herself. However, users do not always know which mapping is best for them, so trial and error is required. As this is a time-consuming process, many users simply give up and get used to the default mapping. This problem can be solved with the approach that helps users become accustomed to operation mapping [12], but systems with personalized mapping can be easier to use. Emacs Key Binding Recommender System (EKBRS) [13] is a system that recommends appropriate key assignments for specific software. The system scores key assignments in accordance with various rules (for example, using the key of the first letter of the word representing the function) and makes recommendations on the basis of the score. The advantage of EKBRS is that the system takes the initiative in adjusting the operation method, rather than leaving it to the user. However, evaluation rules for the recommendations are highly dependent on the expectations of the designers. In order to avoid "collapse of premise", it is necessary to design as few of these rules as possible.

Other research has personalized the operation mapping on the basis of the data acquired from the user in advance without designing the operation method in advance. Niwa et al. [4] showed the action of a humanoid robot to users in advance and had them predict the corresponding operation. By performing pattern matching based on the result of this prior process, they could obtain the operation mapping expected by users. Li et al. [5] also proposed a method for acquiring mapping for robot arms using a similar procedure. The advantage of these techniques is that the system learns operation mapping, which reduces the design assumptions. However, a problem is that additional processes are required to obtain the operation mapping. For example, in the system by Niwa et al., the user needs to perform 540 unresponsive operations while watching a robot that moves automatically. Users may find it inconvenient to undergo such a process before actually performing the operation.

Koyama et al. [14] proposed an efficient method to find appropriate parameters that provide a preferable design. Although the small number of iterations that their method performs is also helpful for the domain of online design, their idea cannot fit the domain of online operation mapping without modification. Specifically, the proposed design parameters are not appropriate for the type of user input utilized in online operation mapping. What the user needs to do should be very simple: that is, they should only have to input direct control operations for the agent. Requiring users to adjust to indirect design parameters prevents them from selecting the preferable operation intuitively. It is essential to focus the user's attention on controlling the agent while eliminating indirect input.

C. Interactive personalization

Research on personalization based on user models is being actively conducted. Dai et al. [15] proposed a POMDP-based adaptive workflow in crowdsourcing and provided worker-specific tasks. Sguerra et al. [16] proposed adapting the UI to reduce cognitive load by modeling human working memory. Some studies have dynamically personalized the interface during user operations. Examples include Adaptive Interface, or Intelligent User Interface (IUI), particularly for use with GUI [17]–[19]. Todi et al. [20] proposed an adaptive UI that can perform tasks quickly by using reinforcement learning. These provide a more optimal UI based on user operation. Torok et al. [21] developed a controller that interactively changes the position and the size of buttons on the touch interface. In this system, relocation is performed on the basis of the user's operation history so that erroneous operations

are reduced. Pelegrino et al. [22] adjusted button positions and added / removed buttons in accordance with in-game context. These approaches solve the "fitting limit" problem by implementing the design process while the user is actually playing. However, they are only used for UI adjustments and do not address the response to user input.

Another research direction has focused on changing the operation interactively through operation assistance by Shared Control (SC). SC is a research field that aims to support the human operations of a machine with a system's intervention. In SC, a human and a system share the control of a machine. Of particular interest here is that an SC system can receive human input and modify its effect, which can be regarded as changing an operation mapping. For example, in research dealing with upper limb assistive devices [23], [24], if the user performs an operation that deviates from the target, the system encourages the user to return to the appropriate route by increasing the operation resistance. The walking support robot developed by Garrotte et al. [25] performes Q-learning while accepting user operations and then converts these into operations that prevent it from hitting obstacles. In studies on semi-automatic driving [26], [27], the system calculates the steering for the ideal state separately from the user operation, and the action is then selected on the basis of both. Focusing on the scrolling operation of a tablet, Fukuchi et al. [28] transformed the normal scrolling operation of the user into an intelligent operation that skips any screen the user does not want to see and stops at the one he or she does. The common problem with the SC approaches mentioned so far is that they only deal with short-term goals, which are specifically pre-defined depending on the particular situation. In rehabilitation or semi-automatic driving, obstacles to avoid are relatively easy to define but when it comes to acquiring operation mapping, pre-defining a short-term goal is likely to cause a "collapse of premise" because short-term goals tend to be highly domain-dependent. We feel it is imperative to acquire operation mapping with as few goals as possible and to make sure they are long-term.

D. Value estimation

What a user intends to do with an operation is strongly connected to his/her goals or values. Various works have focused on estimating values or goals computationally on the basis of user behavior, and our work utilizes the results of such studies to acquire an operation mapping. Inverse reinforcement learning (IRL) [29]–[31], which deals the value estimation problem by structuring it into Markov decision processes, is one such study. IRL has mainly been utilized in the field of imitation learning, which involves the learning and imitating of operations by human experts. Bayesian optimization is also used for value estimation. Andrew et al. [32] aimed to achieve a personalized programming tool by estimating the user's skill using Bayesian inference in block programming. We argue that the value estimation method as described above is also useful for acquiring operation mapping. Simply, if we can dynamically determine a user's value from his/her actions, user operation can be interpreted without assumptions such as use cases.

For example, also in SC, IRL is used by the system to estimate the goal from the user's operation [33]. Reddy et al. [34] applied the value estimated by reinforcement learning to operations by adjusting the operation mapping through pre-training and increased the achievement rate of difficult tasks. To do this, they developed a method to calculate the dynamics of user operation by using the value (Q value) obtained by RL. In this way, various studies have explored estimating the operation target, but virtually all of them focused on adjusting operation mappings from default ones, which can cause "collapse of premise". No research has targeted the problem of acquiring operation mappings interactively in a phase where even default mappings have not been defined.

III. Learning user-preferred operation mappings

In this work, we propose Q-Mapping, a method for interactively acquiring the operation mapping without defining the default one.

A. Formulation

We formulate the user operation process of a system as a Markov decision process (MDP), which is represented as a tuple (S, A, T, R). Let $s_t \in S$ be the state of the system at step t, which is a value that is incremented each time the state s changes.. In addition, to consider the process of transitioning the system state from s_t to the next state s_{t+1}, the system's internal trigger with regard to the system state transition is set to the variable $a \in A$, and also called *action*. The trigger a is assumed to make a deterministic transition of the state of the system. Under this assumption, the transition can be expressed with transition function T as:

$$s_{t+1} = T(s_t, a). \tag{1}$$

The user selects s_{t+1} on the basis of the reward $r \in R$, which means that the user intends to achieve a final goal through the selected state. We define the inverse function of the transition function T that returns a trigger to achieve a specific next state from the current state, and express it as

$$a = T^{-1}(s_t, s_{t+1}). \tag{2}$$

Incidentally, when the user operates the system, we assume he or she first imagines the target state s_{t+1} at the next time point to be achieved based on the current state s_t. With Eq. (2), we can consider the trigger a that corresponds to the state transition.

Here, the user cannot directly input a trigger a to the system because what the user input to control the system is operation $o \in O$ specified by the interface. The user has an operation mapping M in his or her mind and uses it to select o. Note that for the sake of simplicity, we assume a one-to-one correspondence between operation o and action a is assumed. In reality, user operations and intended actions may not be one-to-one, as discussed later in the section VIII.

$$o = M(a) \tag{3}$$

This is a preprint of an article published in IEEE Transactions on Human-Machine Systems.
The final authenticated version is available online at: https://doi.org/10.1109/THMS.2022.3207372

The goal of this work is to estimate which trigger a the user's operation o assumed when performing the operation. Namely, the problem is finding the inverse function of M:

$$a = M^{-1}(o). \tag{4}$$

In reinforcement learning, an agent's policy, or how an agent behaves, is learned while obtaining information from the interaction with the environment on the basis of the MDP settings. In Q-learning [6], which is a method of reinforcement learning, a function that takes a combination of state s and action a is utilized as an argument and returns the expected value of the sum of rewards obtained under a specific policy π called an action value function. It is expressed as

$$Q^{\pi}(s,a) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty}\gamma_t r_{t+1} | s_0 = s, a_0 = a\right]. \tag{5}$$

In order to calculate this expected value, we need to calculate the next states, but this is generally difficult. In Q-learning, the Q value is updated while checking the state of the result of the actual action.

$$Q^{\pi}\left(s_t, a_t\right) \leftarrow (1-\alpha) Q\left(s_t, a_t\right) + \alpha\left[r_{t+1} + \gamma \max_{a} Q\left(s_{t+1}, a\right)\right] \tag{6}$$

Here, α is the learning rate and γ is the discount rate. That is, the Q value for a certain policy is approached to the Q value for a policy that is maximized in the next state by the learning rate α. This propagates its value to the policy that can reach a state where more rewards can be obtained. By repeating a large number of trials, the Q value of each policy converges so that the more effective the policy is for the purpose of the task, the higher the Q value. As a result, it is possible to take the best action to achieve the goal by always selecting the action so that the Q value is maximized. In this study, we apply the Q function-like concept to users, and assume that they select the action based on it.

B. Q-Mapping

Q-Mapping utilizes a Q function-like concept to obtain the mapping between actions and operations. If Q-Mapping can retrieve the user's action evaluation for navigating the agent using a certain operation, the mapping is estimated by a new function $Q^{M^{-1}}$ proposed in Eq. (7).

$$Q^{M^{-1}}(o,a) = \mathbb{E}_{M^{-1}}\left[Q_u^{\pi}(s_t,a) | o_0 = o\right]. \tag{7}$$

Equation (7) expresses that the appropriateness of the combination between actions and the user's operation o is related to the user's action evaluation Q_u^{π} when the user imagines making the agent achieve a goal while performing the operation o. The right side of Eq. (7) allows the user not to have an explicit mapping in his/her mind because there is no direct connection between the action value and the operation. The loose coupling between the action and the operation covers a situation where the user knows what an appropriate action is even though he/she does not know what operation can trigger it. Such cases often appear in the early stages of adaptation. The consensus of the mapping emerges when the user repeats the trial of operating and the agent behaves according to the highest value of Q_u^{π}. Suppose the system can obtain the user's evaluation of actions. In that case, it can acquire the mapping by storing the values of the user's action evaluations as the values of the combination between the user's operation and the action the user wants to choose.

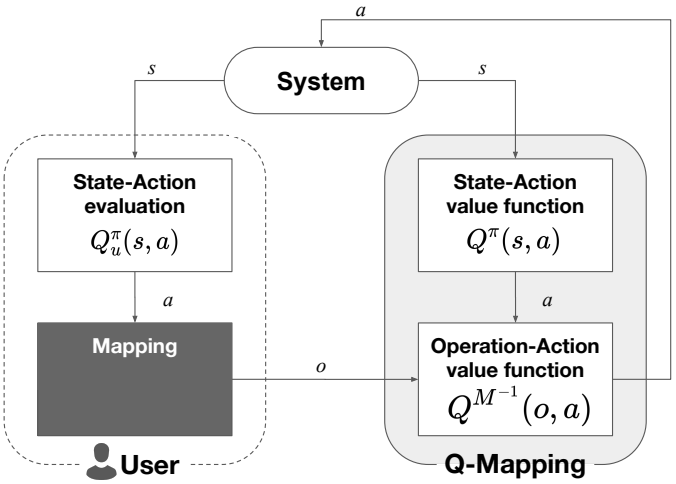


Fig. 1. Overview of Q-Mapping. The left side shows the user ' s cognitive model, and the right side is the Q-Mapping system. According to the assumption that humans also select their actions on the basis of a function similar to the action value function, the operation mapping is inferred from the user operation by using the "operation-action value function".

However, the system cannot obtain the evaluation Q_u^{π} of the actions from the user directly. Instead of Eq. (7), we utilize Eq. (8) to avoid this difficulty.

$$\forall a, Q^{M^{-1}}\left(o_t, a\right) \leftarrow (1-\beta) Q^{M^{-1}}\left(o_t, a\right) + \beta Q^{\pi}\left(s_t, a\right) \tag{8}$$

The idea of Eq. (8) is to use the Q value obtained in Q-learning. We expect the user's evaluation Q_u^{π} of actions to be similar to the Q value if the task offers an optimal policy that the user can find intuitively (see also Fig. 1). That is, the appropriateness of actions that the user considers and the Q value in Q-learning have a similar trend, and Eq. (8) deals with them as the same. Also, we believe that a simple task has the advantage of offering an intuitive policy, and the user can easily find it. As long as the user and the system obey a similar policy related to a task, a consensus of the mapping between the user and the system emerges throughout the iteration of Eq. (8).

We refer to this function $Q^{M^{-1}}$ as the operation-action value function in relation to the action value function Q^{π}. Therefore, $Q^{M^{-1}}$ is updated iteratively for all actions regardless of which action is the target of a performed operation. The point is that the update of Eq. (8) is done on all actions defined in the task every time. For example, imagine the beginning of the adaptation where Q-Mapping has not yet acquired an exact mapping. If the user selects an operation x for intending to move an agent upwards and the task has four types of actions (up, down, left, right), the agent moves according to the highest Q^{π} in the current state. Here, suppose that the agent moves upwards in accordance with the similarity of the user's action

evaluation and the Q value. Equation (8) updates not only the value of $Q^{M^{-1}}$ between the operation x and the upward action but also between x and the rest of the three actions. Since the values of Q^{π} for each action in each state are different, the update gives different values to each pair. We can expect the user to select the other operation y for intending to achieve an additional action: left movement. The operation y will face another balance of Q^{π} for each action; Q^{π} for moving left has a higher value than the others. The different balance assigns the other action to the operation y. The process is the one that Eq. (8) offers to produce the mapping between operations and actions. The iteration of Eq. (8) gives each operation a different balance of values related to actions. The agent obeys the user's operation by selecting an action from four actions by referring to the highest value from the performed operation's value balance.

In addition, β in Eq. (8) is a weighting coefficient that plays a role similar to the learning rate in Q-learning. In Q-Mapping, it is called a *balancing parameter* because it has a role to balance the action value function and the operation-action value function. Using the operation-action value function $Q^{M^{-1}}$ defined above, the system uses the following formula to select the action a to output from operation o.

$$M^{-1}(o) = \arg\max_{a} Q^{M^{-1}}(o, a). \tag{9}$$

For the sake of simplicity, in this study, we assumed a one-to-one correspondence between operation o and action a and used argmax. Since $Q^{M^{-1}}$ of each operation holds the operation-action values of all actions updated by Eq. (8), Q-Mapping can choose an appropriate action by finding the $Q^{M^{-1}}$ that has the highest value regarding the operation the user performed.

The balancing parameter is important because it determines how well Q-Mapping adapts to the user. We decided to define this parameter with reference to the characteristics of human manipulation. When humans perform operations using a controller, they initially perform exploratory operations with an awareness of the operation mapping between operations and actions, but as they get used to it, the operations become habitual and they concentrate only on the actions [35]. Biswas et al. explained that when the user does not know the operation method, the sub-optimal operation is performed, and when the user does know the operation method, the optimum operation is performed [36]. On the basis of the above insights, we expect the following hypothesis to hold even in a system that adapts to human operations without designing operation mapping, such as Q-Mapping: they operate in an exploratory manner at first, but gradually exploit the acquired operation method. Therefore, we presume it is better to converge the balancing parameters so that the user's operations converge to his or her preference. For this reason, we designed the balancing parameter β to be large to adapt well at the beginning and to decrease as the steps progress:

$$\beta(t) = \frac{1}{2}\left(1 - \frac{1}{1 + e^{-k(t-t_0)}}\right). \tag{10}$$

The second term of Eq. (10) represents a logistic curve for t, which draws a flipped s-shaped curve. Here, k represents the steepness of the curve and t_0 represents the midpoint of the curve. β is a function that depends only on the time step t. However, the update timing of β is not intuitive due to the structure of t. The state s changes each time a user operates, which in turn causes the incrementation of t. Thus, β is updated every time the user operates. The Decay β legend in Fig. 2 shows the change in β when t is changed in Eq. (10).

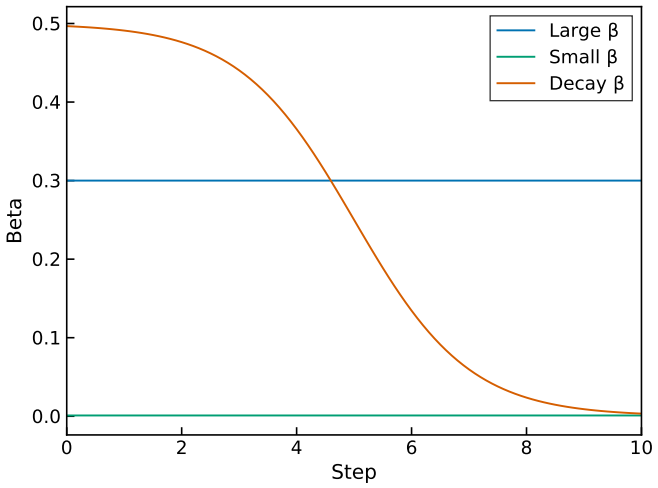


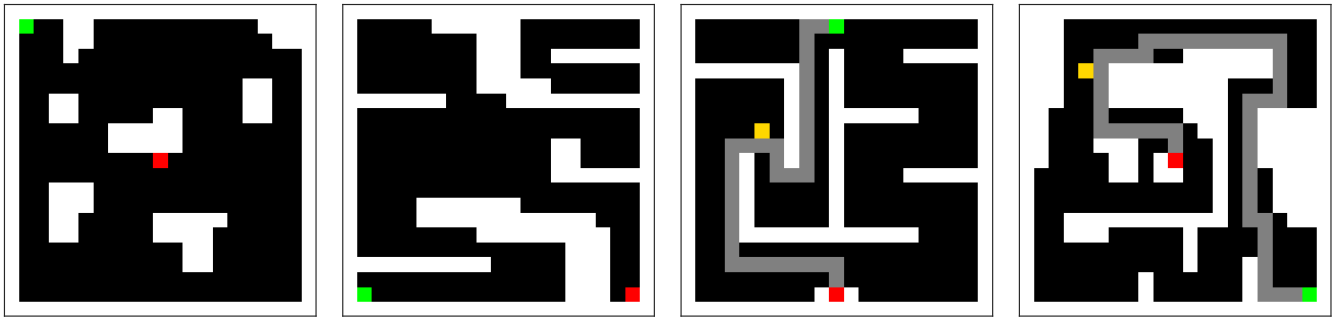
Fig. 2. A graph showing changes of beta. The three lines represent each of the three conditions used in the experiment.

The balancing parameter marks a difference between Q-Mapping and IRL. What makes Q-Mapping differ from IRL is that it changes the dependency on the behaviors of the experts to achieve personalization. IRL is a method of estimating the reward for the task based on the behavior of the expert. In contrast, Q-Mapping estimates the action related to the individual user's input. However, it is not always appropriate to bring the action closer to the behavior of the expert because the intention of each individual is not necessarily the same as that of the expert. The balancing parameter β plays the role of changing the dependency on the expert, who is Q value. Actions on IRL always come from the expert's action selection. In contrast, Q-Mapping varies the dependency according to the balance parameter. Decreasing the dependence after obtaining the mapping enables the user to take his/her own style of operation.

IV. Experiment Design

We conducted an experiment to evaluate whether the user-preferred operation mapping could be obtained by Q-Mapping. To this end, we developed an interface with which a user solves a maze task named GridWorld (Fig. 3). GridWorld is a standard MDP example and is often treated as a Q-learning problem [37], [38]. We decided to use this problem because it is simple and easy for humans to understand. Moreover, the simple task has an advantage of easily analyzing the characteristics of Q-Mapping. In GridWorld, a square environment is divided into a grid, and a road (black) or a wall (white) is arranged for each grid, making it a maze-like task. In this experiment, we set the grid to 21 $\times$ 21 and the controllable agent (red) and the goal (green) one by one

This is a preprint of an article published in IEEE Transactions on Human-Machine Systems.
The final authenticated version is available online at: https://doi.org/10.1109/THMS.2022.3207372



(a) (b) (c) (d)

Fig. 3. Four mazes used as GridWorld problems in the simulation experiment and the user study. Gray lines in (c) and (d) are the routes used in the simulation experiment, and orange dots are the relay points utilized in the user study. These are set for the purpose of confirming whether Q-Mapping can handle it even if users intend to take a non-optimal route.

in the maze. The agent's possible actions are represented by $a \in \{up, down, left, right\}$, with each element representing the agent moving one square in that direction. If there is a wall ahead of the agent, it will not move. When the agent reaches the green goal, the next maze is displayed. In Fig. 3, a user solves the mazes in the order of (a), (b), (c), and (d).

Maze (a) is a simple task that can be accomplished simply by moving up and left. We expect it will be easier to estimate the user's Q value in this task, which will be advantageous for Q-Mapping. In maze (b), in addition to up and left, which was acquired a little in maze (a), the down and right operations are indispensable. The reason mazes (a) and (b) are so simple is to let the users express the intention they have through the operation. In contrast, for mazes (c) and (d), there are two types of branch route. If the action value of Q-learning is used as it is for guessing the user's operation intention, it tends to be interpreted as an operation that follows the optimum route. We designed our experiments to verify whether Q-Mapping can properly adapt the operation mapping even if the user does not want to follow the optimal route.

We trained the system to learn the action value functions in GridWorld using the Deep Q-Network (DQN) [39]. We utilized DQN for the future extension of this research because Q-Mapping has the potential to be applied to more complex tasks than mazes. The reward design was $+1$ when the agent reached the goal, and -0.1 for all other states. In order to get the action value function faster and more accurately, we used UCB-1 [40], which actively visits unsearched environments, as an exploration algorithm. Furthermore, the learning was repeated with all squares except the goal and the wall as the starting position.

In this study, pressing the alphanumeric keys on the keyboard was used as the operation o. The reasons for choosing the keyboard were that it is general and relatively easy to prepare in online user studies, and since it has a large input type for the number of actions in GridWorld, variations for each participant can be expected.

The initial value of $Q^{M^{-1}}(o, a)$ in the Eq. (8) is set to 0.25 so that $\sum_a Q^{M^{-1}}(o, a) = 1$ to make all actions have an equivalent value. Also, for determining the rate of decaying the value of β, we performed the task with several patterns of k and t_0 and set the values that we empirically considered appropriate ($k = 1, t_0 = 5$). With these values, we believe that β asymptotes to 0 while the user is playing the second maze and thus stabilizes the Q-Mapping.

V. Simulation Experiment

In the simulation experiment, we analyzed the operation mapping acquired by Q-Mapping and particularly focused on the effect of the balancing parameter β in the Q-Mapping equation. Our intention here was to observe whether the system could output the target action of the simulated user that imitates human operation while changing β.

A. Hypotheses

We defined the balancing parameter β on the basis of the hypothesis that users operate in an exploratory manner at first and then gradually do what they like (Eq. (10)). In this experiment, we expected that Q-Mapping with Eq. (10) would work best for the simulated human that operates on the basis of this hypothesis. We also expected that when the balancing parameter is large, the action value function would be prioritized and the tendency to select the optimum policy would become stronger, while in contrast, when the balancing parameter is small, the operation-action value function would be prioritized and the tendency to select the action with the estimated operation mapping would become stronger. Therefore, we make the following hypotheses:

- H1 Q-Mapping with a large β should adapt well, so it should work in the first half, but errors increase in the latter half.
- H2 Q-Mapping with a small β should be highly consistent, so although there might be many errors in the first half, it should work well in the latter half.
- H3 Q-Mapping with β that decays with time (Eq. (10)) should work well with few errors throughout.

B. Conditions

To test the effect of balancing parameter β, we prepared Q-Mapping with three different balancing parameters β and compared them:

- Large β Q-Mapping with a relatively large balancing parameter $\beta = 0.3$.

Small β Q-Mapping with a relatively small balancing parameter $\beta = 0.001$.

Decay β Q-Mapping with decaying balancing parameter $\beta = \beta(t)$ (Eq. (10)).

C. Simulated user model

The user operates on the basis of operation mapping M to execute the action a that is executable in the environment. Here, we modeled an operation mapping M in which the user assigns one operation to one a. For example, when a simulated user wants to move to the left, that user sends an arbitrary signal such as o_1 under the intention of moving to the left. The simulated user also assigns o_2, o_3, and o_4 to each of the other three actions. Since this study assumes a one-to-one correspondence between operation o and action a, there are four types of o.

We implemented a desire of the simulated user that initially performed exploratory operations and then gradually performed consistent operations, which is based on the explanation of human operation by Keogh [35]. In the first two mazes ((a) and (b)), the simulated user performs a random operation with a probability of ϵ and a greedy operation($\arg\max_p Q_s(s,p)$) with a probability of $(1-\epsilon)$. As a simulation of getting used to it gradually, we subtracted ϵ in increments of 0.01 for each step:

$$\epsilon = \begin{cases} \epsilon_0 - 0.01t & (\epsilon_0 > 0.01t) \\ 0 & (otherwise). \end{cases} \tag{11}$$

In the latter two mazes ((c) and (d)), the simulated user operates the agent so as to follow the preset route indicated by the gray line in Fig. 3. We set the preset route to the detour of the two branches in mazes (c) and (d). This phase was designed assuming free movement by the user. At this time, the simulated user operates randomly with a probability of 5 % to reproduce the average error of human operation. If the simulated user deviates from the preset route, it operates to return to the route.

We assigned operation signals to the simulated user one by one for each of the four possible actions in GridWorld.

D. Measurements

We performed 1000 trial experiments under each condition while changing $\epsilon \in \{5, 10, 15, 20\}$. We measured the following values:

- the location where misinterpretation occurred during the task, and
- the ratio of misinterpretations to all operations during the task.

We define misinterpretation as the difference between the output intended by the agent and the actual output of the system. By measuring misinterpretation, we show how few errors the system had when guessing the intent. If the mapping adaptation fails, the system can expect to choose the most rational action from its current state, which manifests itself as misinterpretation. These measurements are divided into the first two and latter two mazes for analysis. By looking at these measurements in the first two mazes, we can see whether the model is capable of adapting to early operations, and by looking at the latter two, we can see whether the model was able to perform the desired operation.

E. Results and Discussion

The heat maps in Fig. 4 show the location where misinterpretation occurred during the task. Figure 5 shows the misinterpretation rate of each model in the first two mazes. A Kruskal-Wallis test showed that the difference of β affected the misinterpretation rate under all noise conditions: $H(2) = 9.13, p = .01$, $H(2) = 23.7, p < .001$, $H(2) = 25.8, p < .001$, and $H(2) = 73.5, p < .001$, respectively. The results of a Steel-Dwass test was used to compare all pairs showed that the misinterpretation rate for both Large β and Decay β was significantly lower than that for Small β. Also, there was no significant difference between Large β and Decay β under all noise conditions.

Figure 6 shows the misinterpretation rate of each model in the latter two mazes. A Kruskal-Wallis test showed that the difference of β affected the misinterpretation rate under all noise conditions: $H(2) = 2663, p < .001$, $H(2) = 2688, p < .001$, $H(2) = 10686, p < .001$, and $H(2) = 2672, p < .001$, respectively. The results of a Steel-Dwass test for all noises was used to compare all pairs showed that the misinterpretation rate of Small β was significantly smaller than that of Large β, and that of Decay β was significantly smaller than that of either Large β or Small β. It is notable that the misinterpretation rate of Decay β was 0 % under all conditions.

According to Large β in Figs. 5 and 6, hypothesis **H1** is supported. This suggests that Large β works well when the user selects an action that is close to the optimum. On the other hand, if the route the user wants to follow is clear and not optimal, it will not work well. This is because the action for the operation changes frequently due to the large balancing parameter β. This adaptive feature can sometimes cause the user to have a non-intuitive experience. We can also read this from the position of misinterpretation (Fig. 4). Especially in mazes (c) and (d), more misinterpretations were distributed throughout than in the other two conditions. Furthermore, in maze (d), misinterpretations were distributed at positions different from the preset route. In other words, when noise operations different from the optimum piled up, the Q-Mapping of Large β ignored the intention of the operation and selected the optimum action.

According to Small β in Figs. 5 and 6, hypothesis **H2** is supported. We conclude that the small balancing parameter of Small β delayed the early estimation compared to other conditions. In contrast, the small balancing parameter also provided stability in the latter mazes. The position of misinterpretation (Fig. 4) indicates there were few misinterpretations as a whole. However, in maze (d), misinterpretations were distributed at positions different from the preset route, as in Large β. In other words, although it was less than Large β, the influence of noise may accumulate and provide an action that is different from the user's intention. This possibility is unavoidable as long as the balancing parameter β is a constant.

This is a preprint of an article published in IEEE Transactions on Human-Machine Systems.
The final authenticated version is available online at: https://doi.org/10.1109/THMS.2022.3207372

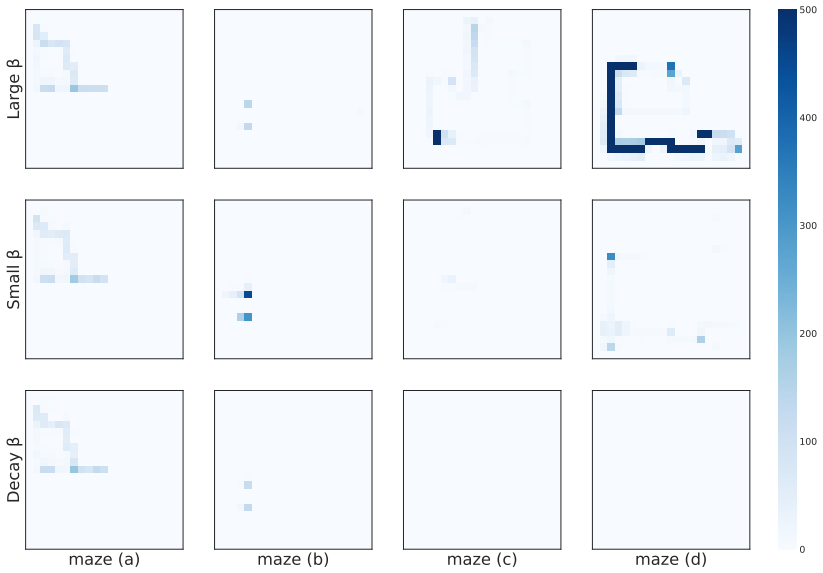


Fig. 4. Heat maps showing misinterpreted positions for each model when $\epsilon = 0.15$. The shade of color expresses the number of times from 0 to 500 (the darkest shade indicates 500 or more).

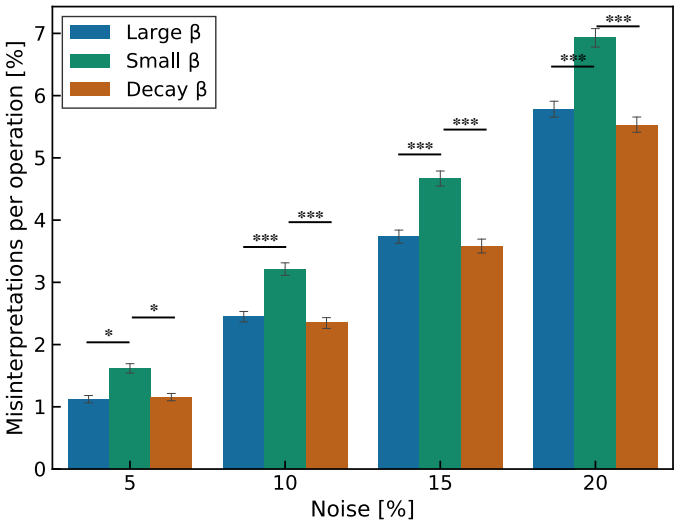


Fig. 5. Misinterpretation rate of each model for noise changes in first two mazes ((a) and (b)). An asterisk (*) means $p < .05$ and triple asterisk (***) means $p < .001$. Error bars show standard errors.

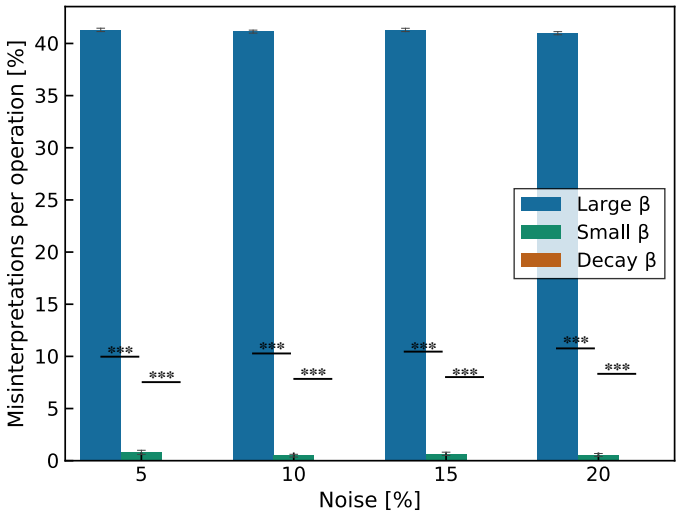


Fig. 6. Misinterpretation rate of each model for noise changes in latter two mazes ((c) and (d)). Triple asterisk (***) means $p < .001$. Error bars show standard errors.

According to Decay β in Figs. 5 and 6, hypothesis **H3** is supported. This result demonstrate that the advantages of Large β and Small β can be combined by the decaying balancing parameter (Eq (10)). Furthermore, the action is rarely changed by the balancing parameter beta being not a constant but gradually decaying. From the above results, we conclude that Decay β's Q-Mapping has the property that the interpretation gradually becomes difficult to change, and returns an action that is easier for the user to understand.

Finally, we discuss the commonalities between each condition. In the first two mazes, the misinterpretation rate increased as the noise increased under all conditions. In contrast, the misinterpretation rate in the latter two did not seem to be influenced by noise, or by the results in the first two. This indicates that Q-Mapping increases the misunderstanding linearly with respect to noise when the operation noise is large. Next, focusing on mazes (a) and (b) in Fig. 4, we can see that misinterpretations occurred at similar points in all models. In (b), misinterpretations were not widely distributed, but rather occured often at certain positions. These locations appear to match where the simulated user entered the first operation in the attempt. This suggests that it is difficult to perform the intended operation in the earliest stage when the operation mapping acquisition is insufficient. Therefore, the earliest environment will need to be carefully designed so that the user's intentions match the optimal behavior. We conclude that it is necessary to strengthen affordance [41] during the earliest stage of the acquisition of operation mapping.

This is a preprint of an article published in IEEE Transactions on Human-Machine Systems.
The final authenticated version is available online at: https://doi.org/10.1109/THMS.2022.3207372

VI. User Study

For the user study, we asked human participants to actually use Q-Mapping and evaluated its behavior. We focused particularly on the balancing parameter and analyzed the operation of human users by comparing the results with those of the simulation experiments. Our idea was to determine the appropriate adaptive controller behavior by comparing the features of β obtained in the simulation experiment with the results of the user study. As measures for evaluating operation mappings acquired by Q-Mapping, we examined whether the task was accomplished and asked the participants for their impressions.

A. Hypotheses

In the simulation experiment, we implemented the simulated user on the basis of the hypothesis that users operates in an exploratory manner at first and then gradually do what they like and found that the Q-Mapping characteristics changed in accordance with the size of the balancing parameter. As hypothesized, Decay β gave the best performance in the simulation experiment. Here, we wanted to see if the same result could be obtained for human users. In a user study, it is difficult to confirm the intentions of all operations so as not to interfere with the task, so we made two additional hypotheses as follows.

- H4 Participants using Q-Mapping with Decay β will have a significantly higher task achievement rate than with other conditions.
- H5 Participants using Q-Mapping with Decay β will feel significantly stronger that they were able to control operations than with other conditions.

B. Conditions

To counterbalance individual differences among participants we designed this experiment using a within-participant approach, where each user performed tasks in all conditions. For comparison with the results of the simulation experiment, the same three conditions were used: **Large** β, **Small** β, and, **Decay** β. We then compared the results of both experiments, to determine which features of Q-Mapping had a positive effect on the user. The content of the task was to solve four GridWorld mazes (from (a) to (d) in Fig. 3, introduced in section IV) and reach the goal, but in the latter two mazes, participants were required to pass through the relay point indicated by the orange square in Fig. 3. This relay point was placed at the point other than the shortest path in the non-optimal branch that makes an action having highest Q value inappropriate. The purpose was to measure whether the operation desired by the user could be output, the same as the preset route in the simulation experiment.

C. Participants

We recruited 30 participants (21 men and nine women; average age: 41.1 years) for 165 Japanese Yen per a person using a crowdsourcing service. All participants agreed to provide the information obtained from the experiment. To counterbalance the order effects, we used the Latin square method to prepare three patterns corresponding to the order of the three conditions and had ten participants perform each pattern.

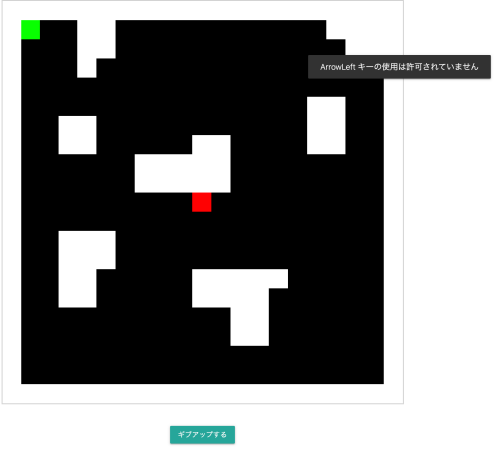


Fig. 7. Screen presented to participants during execution of the task. A large maze is displayed in the center of the screen. The upper right text written in Japanese appears as an alert message when participants press an unauthorized key or when they reach the goal without passing through the relay point. The box at the bottom of the maze is a button for allowing participants to give up on a trial. If participants decide they cannot reach the goal, they can move to the next maze by pressing it.

D. Procedure

The experimental procedure was carried out on the Web with participants using their own PCs. All the documents discussed below were displayed on the Web. Participants first entered their personal information (age and gender) and then received an explanation about GridWorld. They were then given instruction on how to play. We instructed participants to use all alphanumeric keys on the keyboard as controllers. Next, we gave the following instructions: "The key mapping is not fixed. The system guesses the interpretation based on your operation. Try it any way you like." The flow of the experiment was explained as follows: "You need to solve four mazes three times. The system that interprets your operation method is different each time." Participants were asked to repeatedly solve the same maze on three different conditions before proceeding to the task.

Figure 7 shows a screenshot of the interface during the experiment. A large maze is displayed in the center of the screen. Japanese texts in the boxes of the upper right and bottom are as follows. The upper right text appears as an alert message when participants press an unauthorized key or when they reach the goal without passing through the relay point. The box at the bottom of the maze is a button for allowing participants to give up a trial. If participants decide they cannot reach the goal, they can move to the next maze by pressing it.

After each task, participants answered a questionnaire about the system. When they had repeated the tasks and the questionnaire three times, they were informed that all the procedures of the experiment were completed.

This is a preprint of an article published in IEEE Transactions on Human-Machine Systems.
The final authenticated version is available online at: https://doi.org/10.1109/THMS.2022.3207372

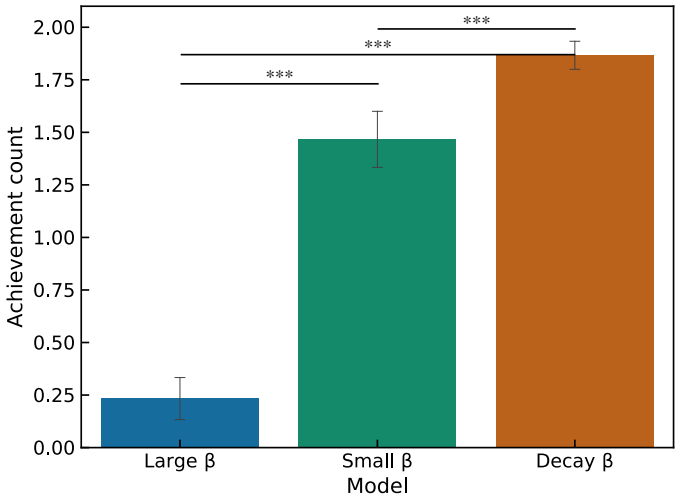


Fig. 8. Achievement count of each condition in mazes (c) and (d). Triple asterisk (***) means $p < .001$. Error bars show standard errors.

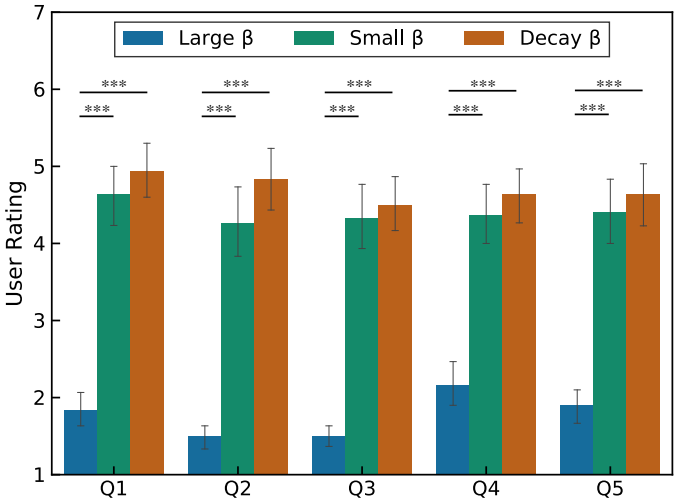


Fig. 9. The results of 7-point Likert scale questionnaire, where 7 is the most positive. Triple asterisk (***) means $p < .001$. Error bars show standard errors.

E. Measurements

Two user operations were recorded as objective indicators for later analysis: valid key inputs and pushing the button to give up. Also, in order to determine if hypothesis H4 was supported, the percentage of tasks for which the give-up button was not pressed was recorded. For each condition, participants were asked to agree on a 7-point Likert scale with the following five statements:

Q1 I was able to operate smoothly.
Q2 I was able to operate what I wanted to do.
Q3 The system guessed what I wanted to do.
Q4 I was able to grasp the operation method.
Q5 The operation method was easy to understand.

Q1 relates to how participants felt about the operation, and the higher the points, the better the impression. Q2 and Q3 examine whether or not participants felt the system was under their control. Particularly in Q3, we ask if participants felt that "The system guesses the interpretation based on your operation" was true. Q4 and Q5 are also questions about whether participants felt they were in control of the system, but regardless of whether the system guessed. In other words, if Q4 and Q5 have high scores but Q2 and Q3 are low, we can presume that the system did not be guess well; rather, the participants adapted to the system. If all the scores from Q1 to Q5 are high, it means that the system was able to guess the operation mapping well.

F. Results

Figure 8 shows the achievement count of the latter two mazes, which is to reach the target via the relay point. In the first two mazes ((a) and (b)), the achievement rate was 100 % under all conditions, so they were excluded from the graph. A non-parametric Friedman test on differences of achievement count was conducted and rendered a chi-square value of 45.4 which was significant ($p < .001$). The results of multiple comparisons using a Durbin-Conover procedure on the achievement count showed that the achievement count was significantly higher when using the Decay β system compared to either Small β or Large β ($p < .001$). These results indicate that Q-Mapping with the Decay β condition was the most helpful for accomplishing the latter two mazes; that is, it could be operated as intended. Thus, hypothesis **H4** is supported.

Figure 9 shows the results of the questionnaire. A Friedman test of differences among repeated measures was conducted for each question. The chi-square values (with p-value) for Q1-5 were 31.3 ($p < .001$), 32.0 ($p < .001$), 35.3 ($p < .001$), 22.0 ($p < .001$), and 30.8 ($p < .001$), respectively. The results of multiple comparisons on the user rating showed that the rating when users used the system with Large β was significantly smaller than with the other two conditions ($p < .001$) in all questions. However, for p-value, the result of the pairwise comparisons between Small β and Decay β for Q1-5 were 0.156, 0.173, 0.199, 0.210, and 0.258, respectively. Therefore, although the rating of Decay β was higher than that of Small β for all the questions, there was no significant difference between the two conditions. Thus, hypothesis **H5** is partially supported.

G. Discussion

First, we discuss the result that supported H4. The reason the number of tasks completed was the highest under the Decay β condition is probably that Q-Mapping with Decay β was easy to understand. Since the operation of Q-Mapping with Decay β rarely changed along the way, it seems that most users understood the operation method. Results showing the same tendency were obtained in the simulation experiment and the user study, demonstrating the effectiveness of Decay β *for the task*. Now we need to ask: did Q-Mapping with Decay β really provide the best operation method for the users?

The questionnaire results showed there was no significant difference between Small β and Decay β, which indicates that although Decay β contributed significantly to the accomplishment of the task, it had little effect on the user's impression.

This is a preprint of an article published in IEEE Transactions on Human-Machine Systems.
The final authenticated version is available online at: https://doi.org/10.1109/THMS.2022.3207372

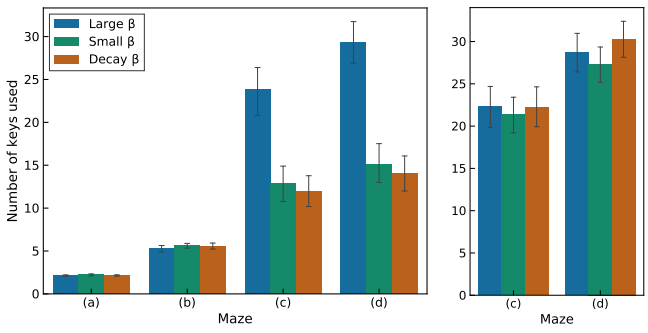


Fig. 10. The number of keys used for operation by the user. The figure on the left shows the result when participants played from mazes (a) to (d). The right figure shows the result when participants played only (c) and (d). Error bars show standard errors.

Let us summarize the results of the user study. Q-Mapping was able to help users achieve their tasks by attenuating β over step.

H. Additional analysis

Since it is possible that the equation for Decay (Eq. (10)) did not work well, we counted the types of operation in each maze to explore the hypothesis that users operate in an exploratory manner at first but gradually do what they like.

Figure 10 shows the mean of the number of key types pressed in each maze. The number of key types in mazes (c) and (d) under Small β and Decay β was reduced compared to that under Large β (the left graph in Fig. 10). The fact that the key variation in (a) is smaller than that in (b) simply indicates that (a) is a simpler maze than (b) and can be cleared with fewer operations. However, we cannot conclude that the adaptation in mazes (a) and (b) resulted in the more efficient acquisition of operation.

We conducted the same experiment on another 30 participants. However, only mazes (c) and (d) were used, in that order. The graph on the right of Fig. 10 shows the mean of the number of key types used at that time.

First, we compared the number of key types used in the first maze played by the user ((a) and (c)) and the second maze played ((b) and (d)). A Kruskal-Wallis test showed that maze (a) ($H(1) = 129, p < .001$) and maze (b) ($H(1) = 120, p < .001$) had significantly fewer keys thanr maze (c) and (d). This is presumably because the mazes (a) and (b) were designed to be simpler than the mazes (c) and (d), which made it easier for the user to search for the operation method.

Second, we compared how the number of key types in mazes (c) and (d) changed due to the presence of mazes (a) and (b) earlier. A Kruskal-Wallis test showed that there were significantly fewer types of operations in both maze (c) and maze (d) when mazes (c) ($H(1) = 12.5, p < .001$) and (d) ($H(1) = 23.5, p < .001$) were performed later in the task. This result demonstrates that the presence of mazes (a) and (b) allowed the user to learn how to operate and complete the task with fewer operations.

This additional analysis showed that users with adaptive controllers performed exploratory operations early in the task.

VII. Interface Personalization based on Q-Mapping

A. Cognitive Gap and Process of Personalization

The experimental results of changing the balancing parameter provide insights for interface personalization. It is important to reduce the cognitive gap in the action value between the user and the system to induce the adaptation based on Q value. Changing the balancing parameter plays a significant role in achieving a personal adaptation according to the phase of the learning process. However, we also need to consider the complexity of a task, which is another important factor related to the cognitive gap and whose importance is obscured behind changing the parameter. The system must choose appropriate values of β according to the phase of the learning process. For example, the system should show a high degree of adaptability at the beginning of the task, when the user is looking for a way to operate the system. Then, it must become more consistent as he/she gets used to it. However, it is not sufficient to change only the balancing parameter.

The system needs to take account of the type of task to achieve interface personalization. In particular, the relation between the value of the balancing parameter and the type of the task is vital for personal adaptation in terms of the cognitive gap. Since the Q-Mapping with the high value of β absorbs the action-operation mapping as it observes, we need to be aware of preparing the task type for the initial stage to use the Q-Mapping. A complex task with multiple goals and multiple candidates for actions causes disrupts the adaptation because the optimal action based on Q value does not necessarily correspond to the one the user chooses. Since the adaptation mechanism utilized by the Q-Mapping eliminates the state from the Q value function of the action and binds it to the operation the user performs, the cognitive gap between the user and the system results in producing an inappropriate mapping, and the system adaptation tends to fail. Therefore, we need to prepare a simple task to offer common action values between the user and the system at the beginning of the adaptation.

Along with the progress of the adaptation, the system should not only decrease the value of β but also introduce a more complex task than the one used at the initial stage. Combining the low value of β with a complex task gives the user opportunities to confirm the adapted controllability on multiple goals and multiple candidates of actions on the complex task. Therefore, the combination of the low value of β and the complex task is crucial for the user to perceive the consistency of the personalization.

Let us summarize the above discussion. The findings of the experiment revealed the importance of combining the value of β and the task complexity to achieve the personalization based on Q value. The critical issue behind personalization is how to reduce the cognitive gap of the action value between the user and the system.

B. Characteristic of Q-Mapping

Here, we discuss whether Q-Mapping solves the two problems in the interface design introduced in related work. The Q-

Mapping we have proposed can be applied to various devices, and it will be easier to use interfaces that previously required proper selection and adjustment [2], [3], [8]–[11]. This device-unrestricted feature solves the "fitting limit" problem.

In addition, Q-Mapping has the feature of personalizing the mapping based on user interaction. This feature eliminates the need to pre-design for "general" users and alleviates the "collapse of premise" problem. The concept of Q-Mapping, which adapts interactively during operation, is based on the concept of IUI [15], [17]–[20] and SC [24]–[26], [28]. Since research on interactive adaptation in the domain of operation methods is still ongoing, we hope that Q-Mapping will serve as a stepping stone. Q-Mapping is similar to value estimation methods such as IRL or Bayesian inference. In one study, research was conducted using the operation of the expert for training to learn the value of the operation and facilitate the operation [34]. The problem with applying value estimation to the personal adaptation of operations is that it takes a long time to train [4], [5]. In contrast, Q-Mapping does not require any training with users. Koyama et al. [14] proposed a process for interactively identifying a user's favorite image using Bayesian inference, but it is difficult to apply a similar process for a domain of operations in which the correct answer is not immediately known.

VIII. Limitation

Q-Mapping can be used at the beginning of a game or in the introduction part of an application because the goals are clear, but it would be difficult to apply Q-Mapping to tasks where the user cannot estimate the optimum action by looking at the state of the environment. For example, in an environment that is so complicated the users do not know what kind of action is possible the first time they see it, they do not perform an operation toward the target state directly but rather an exploratory operation to grasp how it moves. In other words, Q-Mapping is difficult to adapt based on a long-term goal. One idea for Q-Mapping to handle the long-term goals is to apply the cognitive model used by Biswas et al. [36] to evaluate the interface for the acquisition of the operation mapping. Such evaluation of the operation mapping will enable Q-Mapping to acquire the mapping even throughout goal-oriented interactions in a complicated environment.

In this paper, we empirically determined the parameters k and t_0 in Eq. (10) by conducting a trial on maze tasks (a), (b), (c), and (d). However, it is unclear whether these parameters are optimal for the user, as they have not been fully verified. Since Eq. (10) has a great influence on the adaptation method of Q-Mapping, it may be possible to obtain knowledge on the stability of Q-Mapping by comprehensively investigating the parameters. In addition, Q-Mapping that repeats adaptation and forgetting may be possible (depending on the situation) by dynamically changing the parameters as the user becomes accustomed to the adaptive controller.

Q-Mapping targets discrete input devices (for example, keyboard, button, or tap). In order to handle a continuous input device such as a joystick, one of the possible solution is to classify it into discrete values. When classifying continuous inputs, we need to be careful not to include the designer's intention.

In Q-Mapping, assumptions are eliminated as much as possible to prevent "premise collapse", but some assumptions inevitably remain. One of the major assumptions is that one operation always results in one action. First, if the user has the intention of assigning multiple operations to one action, the current model may not be able to adapt well. In our user study, it seemed that few users tried to operate using multiple operations, but we feel that a design that allows multiple operations is necessary to utilize it in various other tasks. Second, this assumption makes it difficult for Q-Mapping to handle cases where the user wants to interpret multiple inputs as one action. Commands on CUI and gesture operations are examples of operations that require multiple inputs to be interpreted as a single action. Future work will need to address how to handle operations that consist of multiple inputs.

IX. Conclusion

Under the assumption that human operation intentions and action value functions are similar, we proposed Q-Mapping, which infers the operation mapping by using the learned Q values. With Q-Mapping, the system can interactively acquire personalized operation mapping for each user. The weighting between Q value and operation mapping can be changed by a balancing parameter that determines the sensitivity of the Q-Mapping response. It is a key parameter in the interaction between humans and systems that changes the response interactively.

We performed simulation experiments to analyze what kind of interpretation change the difference in balancing parameters brought about. We found that Q-Mapping with a large balancing parameter is better in situations where non-optimal (noisy) operations are performed frequently, and Q-Mapping with a smaller one is better when there are many optimal operations.

We also performed a user study and found that participants successfully achieve the given maze tasks under Q-Mapping with the decaying balancing parameter. On the other hand, the design in which the balancing parameter was large in the early stage and gradually attenuated did not affect the user impression very much.

These results demonstrate that when users utilize a controller that adaptively acquires operations, they perform exploratory operations in order to determine the operation method in the early stage, and then exploit the operation method step by step. This conclusion will help in designing controllers that are interactively personalized.

References

[1] K. Gerling, F. Schulte, J. Smeddinck, and M. Masuch, "Game design for older adults: Effects of age-related changes on structural elements of digital games," in *The International Conference on Entertainment Computing (ICEC '12)*, Bremen, Germany, 2012, pp. 235–242.

[2] D. Jeevithashree, K. Saluja, and P. Biswas, "A case study of developing gaze controlled interface for users with severe speech and motor impairment," *Technology and Disability*, vol. 31, pp. 63–76, 06 2019.

[3] V. Sharma, M. L R D, K. Saluja, V. Mollyn, G. Sharma, and P. Biswas, "Webcam controlled robotic arm for persons with ssmi," *Technology and Disability*, vol. 32, pp. 1–19, 06 2020.

[4] M. Niwa, S. Okada, S. Sakaguchi, K. Azuma, H. Iizuka, H. Ando, and T. Maeda, "Detection and transmission of "tsumori": an archetype of behavioral intention in controlling a humanoid robot," in *International Conference on Artificial Reality and Telexistence (ICAT 2010)*, Adelaide, AU, 2010, pp. 193–196.

[5] M. Li, D. P. Losey, J. Bohg, and D. Sadigh, "Learning user-preferred mappings for intuitive robot control," 2020.

[6] C. Watkins and P. Dayan, "Q-learning," *Mach Learn*, vol. 8, pp. 279–292, 1992.

[7] A. Dvorak, "There is a better typewriter keyboard," *National Business Education Quarterly*, vol. 12, no. 2, pp. 51–58,66, 1943. [Online]. Available: https://ci.nii.ac.jp/naid/10012049615/

[8] J. Gray, P. Jia, H. H. Hu, T. Lu, and K. Yuan, "Head gesture recognition for hands-free control of an intelligent wheelchair," *Industrial Robot: An International Journal*, 2007.

[9] D. Bassily, C. Georgoulas, J. Guettler, T. Linner, and T. Bock, "Intuitive and adaptive robotic arm manipulation using the leap motion controller," in *ISR/Robotik 2014; 41st International Symposium on Robotics*. VDE, 2014, pp. 1–7.

[10] Y. Matsumotot, T. Ino, and T. Ogsawara, "Development of intelligent wheelchair system with face and gaze based interface," in *Proceedings 10th IEEE International Workshop on Robot and Human Interactive Communication. ROMAN 2001 (Cat. No. 01TH8591)*. IEEE, 2001, pp. 262–267.

[11] D. Purwanto, R. Mardiyanto, and K. Arai, "Electric wheelchair control with gaze direction and eye blinking," *Artificial Life and Robotics*, vol. 14, no. 3, p. 397, 2009.

[12] S. Malacria, G. Bailly, J. Harrison, A. Cockburn, and C. Gutwin, *Promoting Hotkey Use through Rehearsal with ExposeHK*. New York, NY, USA: Association for Computing Machinery, 2013, p. 573ˆˆe2ˆˆ80ˆˆ93582. [Online]. Available: https://doi.org/10.1145/2470654.2470735

[13] S. Zhong and H. Xu, "Intelligently recommending key bindings on physical keyboards with demonstrations in emacs," in *Proceedings of the 24th International Conference on Intelligent User Interfaces*, ser. IUI '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 12ˆˆe2ˆˆ80ˆˆ9317. [Online]. Available: https://doi.org/10.1145/3301275.3302272

[14] Y. Koyama, I. Sato, and M. Goto, "Sequential gallery for interactive visual design optimization," *ACM Trans. Graph.*, vol. 39, no. 4, Jul. 2020. [Online]. Available: https://doi.org/10.1145/3386569.3392444

[15] P. Dai, C. H. Lin, Mausam, and D. S. Weld, "Pomdp-based control of workflows for crowdsourcing," *Artificial Intelligence*, vol. 202, pp. 52–85, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S000437021300057X

[16] B. Massoni Sguerra and P. Jouvelot, ""an unscented hound for working memory" and the cognitive adaptation of user interfaces," in *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization*, ser. UMAP '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 78ˆˆe2ˆˆ80ˆˆ9385. [Online]. Available: https://doi.org/10.1145/3320435.3320443

[17] C. Wiecha, W. Bennett, S. Boies, J. Gould, and S. Greene, "Its: a tool for rapidly developing interactive applications," *ACM Transactions on Information Systems (TOIS)*, vol. 8, no. 3, pp. 204–236, 1990.

[18] D. R. Olsen Jr, S. Jefferies, T. Nielsen, W. Moyes, and P. Fredrickson, "Cross-modal interaction using xweb," in *Proceedings of the 13th annual ACM symposium on User interface software and technology*, 2000, pp. 191–200.

[19] K. Z. Gajos, D. S. Weld, and J. O. Wobbrock, "Automatically generating personalized user interfaces with supple," *Artificial Intelligence*, vol. 174, no. 12-13, pp. 910–950, 2010.

[20] K. Todi, G. Bailly, L. Leiva, and A. Oulasvirta, "Adapting user interfaces with model-based reinforcement learning," 05 2021, pp. 1–13.

[21] L. Torok, M. Pelegrino, J. Lessa, D. Trevisan, and E. Clua, "Adaptcontrol: An adaptive mobile touch control for games," in *SIGGRAPH Asia 2014 Mobile Graphics and Interactive Applications*, ser. SA '14. New York, NY, USA: Association for Computing Machinery, 2014. [Online]. Available: https://doi.org/10.1145/2669062.2669081

[22] M. Pelegrino, L. Torok, D. Trevisan, and E. Clua, "Creating and designing customized and dynamic game interfaces using smartphones and touchscreen," in *2014 Brazilian Symposium on Computer Games and Digital Entertainment*. IEEE, 2014, pp. 133–139.

[23] J. L. Emken, J. E. Bobrow, and D. J. Reinkensmeyer, "Robotic movement training as an optimization problem: designing a controller that assists only as needed," in *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.*, 2005, pp. 307–312.

[24] E. Wolbrecht, V. Chan, D. Reinkensmeyer, and J. Bobrow, "Optimizing compliant, model-based robotic assistance to promote neurorehabilitation," vol. 16, no. 3, pp. 286–297, Jun. 2018.

[25] L. Garrote, J. Paulo, J. Perdiz, P. Peixoto, and U. J. Nunes, "Robot-assisted navigation for a robotic walker with aided user intent," in *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2018, pp. 348–355.

[26] V. A. Shia, Y. Gao, R. Vasudevan, K. D. Campbell, T. Lin, F. Borrelli, and R. Bajcsy, "Semiautonomous vehicular control using driver modeling," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 6, pp. 2696–2709, 2014.

[27] Y. Gao, A. Gray, A. Carvalho, H. E. Tseng, and F. Borrelli, "Robust nonlinear predictive control for semiautonomous ground vehicles," in *2014 American Control Conference*. IEEE, 2014, pp. 4913–4918.

[28] Y. Fukuchi, Y. Takimoto, and M. Imai, "Adaptive enhancement of swipe manipulations on touch screens with content-awareness," in *12th International Conference on Agents and Artificial Intelligence, ICAART 2020*. SciTePress, 2020, pp. 429–436.

[29] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*, 2004, p. 1.

[30] N. D. Ratliff, J. A. Bagnell, and M. A. Zinkevich, "Maximum margin planning," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 729ˆˆe2ˆˆ80ˆˆ93736. [Online]. Available: https://doi.org/10.1145/1143844.1143936

[31] D. Ramachandran and E. Amir, "Bayesian inverse reinforcement learning." in *IJCAI*, vol. 7, 2007, pp. 2586–2591.

[32] A. Emerson, M. Geden, A. Smith, E. Wiebe, B. Mott, K. E. Boyer, and J. Lester, "Predictive student modeling in block-based programming environments with bayesian hierarchical models," ser. UMAP '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 62ˆˆe2ˆˆ80ˆˆ9370. [Online]. Available: https://doi.org/10.1145/3340631.3394853

[33] S. Javdani, S. S. Srinivasa, and J. A. Bagnell, "Shared autonomy via hindsight optimization," *Robotics science and systems: online proceedings*, vol. 2015, 2015.

[34] S. Reddy, A. Dragan, and S. Levine, "Where do you think you're going?: Inferring beliefs about dynamics from behavior," in *Advances in Neural Information Processing Systems*, 2018, pp. 1454–1465.

[35] B. Keogh, "A play of bodies: A phenomenology of videogame experience," 2015.

[36] P. Biswas and P. Robinson, "Automatic evaluation of assistive interfaces," in *Proceedings of the 13th International Conference on Intelligent User Interfaces*, ser. IUI '08. New York, NY, USA: Association for Computing Machinery, 2008, pp. 247–256. [Online]. Available: https://doi.org/10.1145/1378773.1378806

[37] R. Dearden, N. Friedman, and S. Russell, "Bayesian q-learning," in *Aaai/iaai*, 1998, pp. 761–768.

[38] H. Hasselt, "Double q-learning," in *Advances in Neural Information Processing Systems*, J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds., vol. 23. Curran Associates, Inc., 2010, pp. 2613–2621.

[39] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," 2013.

[40] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," vol. 47, pp. 235–256, 2002.

[41] D. A. Norman, "Affordance, conventions, and design," *interactions*, vol. 6, no. 3, pp. 38–43, 1999.

Riki Satogata Riki Satogata was born in Japan, in 1995. He received the B.E. and M.E. degrees in computer science from Keio University, Yokohama, Japan, in 2019 and 2021, respectively.

Mitsuhiko Kimoto Mitsuhiko Kimoto received M. Eng. and Ph.D. degrees from Doshisha University, Kyoto, Japan in 2016 and 2019. He is currently a research scientist at Interaction Science Laboratories (ISL), the Advanced Telecommunications Research Institute International (ATR). His research interests include social robotics, human-agent interaction, and interactive AI.

Yosuke Fukuchi Yosuke Fukuchi was born in Japan, in 1994. He received the B.E. and M.E. degrees in computer science from Keio University, Yokohama, Japan, in 2017 and 2019, respectively. From 2019 to 2021, he was an Assistant Professor at Keio University. From 2021 to 2022, he was a Project Researcher with the Keio Leading-edge Laboratory of Science and Technology (KLL). He is currently a Project Researcher at the National Institute of Informatics, Tokyo, Japan. His research interests include human^^e2^^80^^93agent interaction, artificial intelligence, and theory of mind. He is a member of the Japanese Society for Artificial Intelligence.

Kohei Okuoka Kohei Okuoka received the M.S. and B.S. in Computer Science from Keio University in 2018 and 2020, respectively. He is currently a Ph.D. student in the Graduate school of science and technology at Keio university. His research interests include autonomous robots, human-agent interaction, and cognitive science. He is a member of the Japanese Cognitive Science Society.

Michita Imai Michita Imai is a Professor of the Faculty of science and technology at Keio university and a Researcher at ATR Intelligent Robot Laboratories. He received his Ph.D. degree in Computer Science from Keio Univ. in 2002. In 1994, he joined NTT Human Interface Laboratories. He joined the ATR Media Integration & Communications Research Laboratories in 1997. He was a visiting scholar of The University of Chicago from 2009-2010. His research interests include autonomous robots, human-robot interaction, speech dialogue systems, humanoids, and spontaneous behaviors. He is a member of Information and Communication Engineers Japan (IEICE-J), the Information Processing Society of Japan, the Japanese Cognitive Science Society, the Japanese Society for Artificial Intelligence, Human Interface Society, IEEE, and ACM.