Adaptive touch panel user interface by type-based approach using particle filters

Norikazu Ikoma, Witold Pedrycz*, Takahiro Hyakudome**, Hiroshi Maeda

Faculty of Engineering, Kyushu Institute of Technology, Fukuoka 804-8550, Japan
ikoma@comp.kyutech.ac.jp
*Dept.of Electrical & Computer Engineering, University of Alberta, Edmonton, Canada, and Systems Research Institute Polish Academy of Sciences, Warsaw, Poland
**Panasonic Communications Co., Ltd., Japan

Abstract

A new method for adaptation of touch panel user interface is proposed. The method employs a type-based approach effectively using prior information on user population which is represented by mixture model. An advantage of the approach is that we only need to estimate weights of types which is numerically far less expensive than estimating values of all attributes of the user. State space model is formalized to estimate the weights of types by supposing smoothness prior to time evolution of the weights and mixture model with the time-varying weights. Particle filter is used to estimate the weights based on observation series of user operations up to current time. An experiment on touch panel user interface demonstrates the efficiency of the proposed method.
Keywords: adaptive user interface, type-based approach, particle filters, mixture model.

1. Introduction

Recently computer technology has been greatly developed and it is widely used for various industrial products including home electronics, mobile communications, vehicles, etc. These technologies can make the products more intelligent than before with lower cost and less capacity and weight, so there are so many possibilities to produce a new product more useful and human friendly than before. This is particular interest to elderly people or handicapped to establish so-called 'barrier free' society. According to these situations, we are interested in making human interface more intelligent than the conventional one by using the latest technologies. Especially, adaptation of the interface to a specific user is one of the most important topics to achieve the intelligent human interface.

We propose a type-based approach to the adaptation of human interface. The approach assumes types on user population which have been extracted through prior experiment using mixture model. Where, each component of the mixture model represents the characteristic of each user type. At the adaptation, we effectively use the user type information such that we only estimate weights of types which represent belongings of current user to the types. An advantage of this approach is that it requires less computational cost than estimating all attributes of the user. For the user type estimation, we propose a state space model having smooth time-evolution of the weights and mixture model conditioned on the weights. By estimating the state of the state space model, we obtain the estimate of the user type in a form of conditional distribution of the weights given the series of observation up to current time. Adaptation of user interface is achieved by combining the optimal interfaces of every type according to the estimated weights.

To evaluate our approach, we have conducted a user adaptation experiment on touch panel interface which is originated from guidance and control system for intelligent wheel chair aimed at near future vehicles for handicapped person. Function of the interface is essentially to choose a destination of the vehicle. After operating the interface, it automatically moves toward the destination. As adaptation factors of the interface, we focus on size and the number of buttons displayed on the panel. For example, if the size is too small, it is difficult to see and to choose it for a person having problem on one's eye or one's hand. On the other hand, if the size is too large, the user needs to operate the panel frequently to complete the selection of destination since one need to turn pages of the panel many times. Thus the task of adaptation is to decide the optimal size and the number of buttons based on the operation by the user.

2. Method

The proposed method consists of three parts, 1) user type extraction, 2) user type estimation [6], and 3) adaptation of interface [5] [4]. We will use mixture model to extract and represent various types of users. The mixture model is estimated in the user type extraction part beforehand the user type estimation and the adaptation of interface. Next, in the user type estimation part, state space model will be used as a model for estimating the type of user. Where, estimation result is in a form of weights of types computed from conditional distribution obtained by state estimation scheme. Finally, in the adaptation of interface part, the weights will be used to synthesize the optimal user interface for current user by combining optimal interfaces of every type according to the weights.

2.1 User type extraction

We use a mixture model to describe and extract the types of the user population. Each component of the mixture model is considered as a distribution representing characteristics of each user type. The mixture model is defined in observation space which consists of certain items possible to observe during the operation of the interface. Let E_O be the observation space.

Possible examples of elements of E_O include quickness, accuracy, and correction rate. To represent characteristics of user population consisting of several user types (say K types), we use a mixture model over E_O denoted by its density function

$$f(\mathbf{x}|\mathbf{u};\mathbf{\Theta},\mathbf{w})=\sum_{k=1}^{K}w_k f_k(\mathbf{x}|\mathbf{u};\theta_k), \qquad (1)$$

with $\mathbf{x} \in E_O$ and $\mathbf{u} \in E_U$, where $\mathbf{\Theta} = (\theta_1,\theta_2,\cdots,\theta_K) \in E_\theta$ is collection of parameter vectors of all components, $\mathbf{w} = (w_1, w_2, \cdots, w_K) \in E_S$ is vector of weights, and $\mathbf{u} \in E_U$ denotes a type of the user interface presented to user. We refer to E_U as 'interface space'. It is possible to represent dependency of $\mathbf{x}$ to external input $\mathbf{z}$ over input space E_Z, but we suppress to show this factor in this formulation. We consider that the weights sum up to 1 and are non-negative, that is $w_k \geq 0$ for all $k = 1,2,\cdots,K$ and $\sum_{k=1}^{K} w_k = 1$.

Procedure to extract user types based on the mixture model in (1) is as follows. Suppose that the interface space E_U is defined beforehand the user type extraction. First, decide the observation space E_O and the input space E_Z according to the specific features of user population. It will depend on 1)what items we can measure during operation of the user interface, 2)what items of measurements are dominant to represent the population characteristics in a form of mixture model, and 3)what factor should be incorporated as the external input to describe dependency of the observation to the other factors. Second, decide distribution of all components of the mixture model, $f_k(), k = 1,2,\cdots,K$, as well as the number of components K. The decision of components specifies the parameters space E_Θ as a result. Third, conduct experiment measuring users' operations on E_O over various situations of $\mathbf{u} \in E_U$ and $\mathbf{z} \in E_Z$ for many users as possible. Then we are ready to estimate the mixture model with data set of the measurements. Fourth and finally, we estimate the mixture model by any estimation method such as EM algorithm [1]. There are another choices on the estimation method whatever it works, e.g., maximum likelihood or moment matching method for each component. Then we obtain the estimate of parameters $\hat{\mathbf{\Theta}}_0$ and weights $\hat{\mathbf{w}}_0$.

2.2 User type estimation

We define estimation of user type as a task to estimate the weights of the mixture model in (1) for 'a specific user', not for population of users. That is, it is to obtain belongings of the user for each type. We refer to the specific user as "current user". The task will be achieved by observing the performance of the current user and estimating the conditional distribution of the weights given the observation series up to current time. In a simplest situation where user type is supposed to be fixed (but unknown) weights, this can be formulated by Bayesian recursive estimation such that

$$\begin{aligned} p(\mathbf{w}|\mathbf{x}(1:t)) \propto \\ p(\mathbf{w}|\mathbf{x}(1:t-1)) f(\mathbf{x}(t)|\mathbf{u}(t);\hat{\mathbf{\Theta}}_0,\mathbf{w}) \end{aligned} \qquad (2)$$

where $\mathbf{x}(t)$ and $\mathbf{u}(t)$ respectively denote observation and presented interface at (discrete) time t, and $\mathbf{x}(1:t)$ represents observation series up to time t. We begin the recursion of eq.(2) with an appropriate initial distribution of weights parameterized by the estimated weights at the user type extraction part, i.e., $p(\mathbf{w};\hat{\mathbf{w}}_0)$.

We further extend the situation above to involve time-varying weights by letting the weights to have time index t, i.e., $\mathbf{w}(t)$. Here we assume time smoothness for the change of weight parameters in a form of conditional distribution of Markov process such that

$$p(\mathbf{w}(t)|\mathbf{w}(t-1)). \qquad (3)$$

This distribution represents smooth change of weights. For example, if all weights are simply real values (i.e., not weights with constraints of non-negative and sum up to 1 property), then the smoothness can be represented by, for example, Gaussian distribution with mean $\mathbf{w}(t-1)$ and certain covariance matrix. This represents random walk model for the change of weight parameters. We can use another kind of distributions such as uniform, mixture, Cauchy, etc., depending on the problem we are in hand. However, we need to be sure to keep the constraints of non-negative and sum up to 1 property by using, for example, logarithmic transformation and normalization.

Then, the Bayesian recursion becomes more complicated form than eq.(2) and reads in the form

$$\begin{aligned} p(\mathbf{w}(1:t)|\mathbf{x}(1:t)) \propto p(\mathbf{w}(1:t-1)|\mathbf{x}(1:t-1)) \\ f(\mathbf{x}(t)|\mathbf{u}(t);\hat{\mathbf{\Theta}}_0,\mathbf{w}(t)) p(\mathbf{w}(t)|\mathbf{w}(t-1)), \end{aligned} \qquad (4)$$

where target distribution is replaced to $p(\mathbf{w}(1:t)|\mathbf{x}(1:t))$ from $p(\mathbf{w}|\mathbf{x}(1:t))$ of eq.(2), not only putting time index t, but also extending the index to range $1:t$ to have concise notation. To obtain $p(\mathbf{w}(t)|\mathbf{x}(1:t))$ formally from the result of eq.(4), we just marginalize $p(\mathbf{w}(1:t)|\mathbf{x}(1:t))$ to $\mathbf{w}(t)$.

Actual estimation using eq.(4) proceeds with particle filters [2] due to its non-Gaussian properties. That is, eq.(4) does not have closed-form solution so we need to use some approximation method to solve it. Particle filters approximate the target distribution by many samples in space of weight E_S, where we will call E_S as 'state space' according to a context of state space modeling. These samples are called 'particles'. Each particle may have weight to correct discrepancy between a distribution drawn from and the target distribution based on the idea of importance sampling. Details of state estimation by particle filters will be explained in section 3.

2.3 Adaptation of interface

Although there are many possible ways to accomplish the adaptation, we have employed a basic idea to combine the estimated weights into optimal interfaces for each user type. The optimal user interfaces of each user type are prepared beforehand the adaptation. There are also many possible ways to combine the estimated weights; among them, we explain a simple way as follows.

First, representative value of estimated weight is calculated from the distribution obtained by eq.(4). It can be done by taking average, or calculating median, or MAP (Maximum A Posteriori) of the distribution $p(\mathbf{w}(t)|\mathbf{x}(1:t))$. Then we have the representative value $\hat{\mathbf{w}}(t)$. Let $\mathbf{u}_k^*$ be the optimal user interface of k-th component (i.e., type). Then, the user interface to be presented to the current user will be synthesized conceptually by taking a weighted sum of the optimal interfaces

$$\mathbf{u}^*(t) = \sum_{k=1}^{K} \hat{w}_k(t)\mathbf{u}_k^* \tag{5}$$

with notation $\hat{\mathbf{w}}(t) = (\hat{w}_1(t), \hat{w}_2(t), \cdots, \hat{w}_K(t))$. Note that eq.(5) is only conceptual weighted sum, so actual calculation may be more complicated one than represented in eq.(5). It might proceed in attribute space of the interface with specific constraints of the space.

3. Estimation

State estimation is a task to obtain conditional distribution $p(\mathbf{w}(1:t)|\mathbf{x}(1:t))$ using eq.(4) recursively. However, there is no closed-form solution to the eq.(4). So we need to use some approximation method to proceed the calculation of eq.(4). Particle filters can approximately solve eq.(4), where the term 'particle filters' is a generic one to refer to a class of methods using particles in state space. It is also called Sequential Monte Carlo since it proceeds Monte Carlo method sequentially according to eq.(4). Here we firstly explain the idea of importance sampling followed by a review on sequential Monte Carlo using the fact derived at the importance sampling explanation.

In our model, particles are represented by a set of weight instances such that $\left\{\mathbf{w}^{(i)}(1:t)\right\}_{i=1}^{M}$, with i be an index of instance and M be the number of particles. We assume that the particles are drawn from so-called 'proposal distribution', which is denoted by $q(\mathbf{w}(1:t)|\mathbf{x}(1:t))$. Notice that it is necessary for the proposal to satisfy a condition $q(\mathbf{w}|\mathbf{x}) > 0$ for any $\mathbf{w}$ of $p(\mathbf{w}|\mathbf{x}) > 0$. To approximate the target distribution $p(\mathbf{w}(1:t)|\mathbf{x}(1:t))$ by the set of particles drawn from the proposal, we need to calculated 'weight' of particle to adjust the discrepancy between the proposal and the target distribution. The weight is calculated by a formula

$$\alpha(\mathbf{w}(1:t)) \propto p(\mathbf{w}|\mathbf{x})/q(\mathbf{w}|\mathbf{x}), \tag{6}$$

i.e., $\alpha_t^{(i)} \propto p(\mathbf{w}^{(i)}(1:t)|\mathbf{x}(1:t))/q(\mathbf{w}^{(i)}(1:t)|\mathbf{x}(1:t))$ is the weight for i-th particle. Then we obtain an approximated density of the target $p(\mathbf{w}|\mathbf{x})$ by

$$\begin{aligned} p(\mathbf{w}|\mathbf{x}) &= \alpha(\mathbf{w})q(\mathbf{w}|\mathbf{x}) \\ &\simeq \tfrac{1}{M}\textstyle\sum_{i=1}^{M} \alpha^{(i)}\delta(\mathbf{w}-\mathbf{w}^{(i)}) \equiv \hat{p}(\mathbf{w}|\mathbf{x}) \end{aligned} \tag{7}$$

where $\delta(\mathbf{x})$ is Dirac delta function that yields value 1 when it is integrated over a region including 0, yields value 0 otherwise. To obtain Monte Carlo estimation of mean (say) of the target distribution, we integrate eq.(7) multiplied by $\mathbf{w}$, thus we have

$$\begin{aligned} &\mathrm{E}\left[\mathbf{w}(1:t)|\mathbf{x}(1:t)\right] \\ &\quad= \textstyle\int \mathbf{w}(1:t)p(\mathbf{w}(1:t)|\mathbf{x}(1:t))d\mathbf{w}(1:t) \\ &\quad= \frac{\int \mathbf{w}(1:t)p(\mathbf{w}(1:t)|\mathbf{x}(1:t))d\mathbf{w}(1:t)}{\int p(\mathbf{w}(1:t)|\mathbf{x}(1:t))d\mathbf{w}(1:t)} \\ &\quad\simeq \frac{\int \mathbf{w}(1:t)\hat{p}(\mathbf{w}(1:t)|\mathbf{x}(1:t))d\mathbf{w}(1:t)}{\int \hat{p}(\mathbf{w}(1:t)|\mathbf{x}(1:t))d\mathbf{w}(1:t)} \\ &\quad= \textstyle\sum_{i=1}^{M} \tilde{\alpha}_t^{(i)}\mathbf{w}^{(i)}(1:t) \equiv \hat{\mathbf{w}}(1:t) \end{aligned} \tag{8}$$

where $\tilde{\alpha}_t^{(i)}$ is normalized version of $\alpha_t^{(i)}$ having sum up to 1 property.

So far we have reviewed importance sampling method directly on the target distribution $p(\mathbf{w}(1:t)|\mathbf{x}(1:t))$. From here we will review on a sequential method for the estimation, which is so-called sequential Monte Carlo. To begin we decompose the proposal into two parts, i.e., current and past, such that

$$\begin{aligned} q(\mathbf{w}(1:t)|\mathbf{x}(1:t)) &\equiv q(\mathbf{w}(t)|\mathbf{x}(1:t), \mathbf{w}(1:t-1)) \\ &\quad\times q(\mathbf{w}(1:t-1)|\mathbf{x}(1:t-1)) \end{aligned} \tag{9}$$

notice that second term of right hand side in eq.(9) is $q(\mathbf{w}(1:t-1)|\mathbf{x}(1:t-1))$, not $q(\mathbf{w}(1:t-1)|\mathbf{x}(1:t))$. This represents the fact that we sequentially conduct the estimation so we had only used an observation series up to the time when the particles were drawn. For example, the particle at time $t-1$, denoted as $\mathbf{w}^{(i)}(t-1)$, were drawn by using observation series up to time $t-1$, so the proposal must be $q(\mathbf{w}(1:t-1)|\mathbf{x}(1:t-1))$, not $q(\mathbf{w}(1:t-1)|\mathbf{x}(1:t))$. Eq.(9) recursively applies to all the past times, i.e., $\mathbf{w}(t-1)$, $\mathbf{w}(t-2)$, and so on, thus it becomes products of first term in the right hand side of eq.(9).

Using similar decomposition of the target distribution $p(\mathbf{w}(1:t)|\mathbf{x}(1:t))$ and divide eq.(4) by both sides of eq.(9), we have the weight update formula

$$\alpha_t^{(i)} \propto \alpha_{t-1}^{(i)} \times \frac{f(\mathbf{x}(t)|\mathbf{u}(t);\hat{\mathbf{\Theta}}_0, \mathbf{w}^{(i)}(t))p(\mathbf{w}^{(i)}(t)|\mathbf{w}^{(i)}(t-1))}{q(\mathbf{w}^{(i)}(t)|\mathbf{x}(1:t), \mathbf{w}^{(i)}(1:t-1))}. \tag{10}$$

The final step of sequential Monte Carlo method is 'resampling'. It is necessary to overcome so-called 'degeneracy of the weight problem' (a weight shrinkage problem, i.e., very small number of particles have positive weights while all other particles have zero weights and it causes waste of computational burden), but not necessary for all iteration of sequential estimation. It is better to do the resampling when some criterion matches in order to avoid involving further Monte Carlo error. As the criterion, Effective Sample Size(ESS) is the most popular one, which is inverse of squared sum of normalized weight such that

$$ESS(t) = 1/\sum_{i=1}^{M}\left(\tilde{\alpha}_t^{(i)}\right)^2. \tag{11}$$

We can see that if all weights are equivalent, i.e., having the same value $1/M$ for all $\tilde{\alpha}_t^{(i)}$, $i = 1, 2, \cdots, M$, then $ESS(t)$ has value M and it means all particles are effective. On the other hand, if only one particle has value 1 weight and all the other have value 0 weights, then $ESS(t)$ becomes 1, meaning that only one particle is effective. Thus if $ESS(t)$ is below some specified threshold, e.g., $2M/3$, then we conduct the resampling step formalized as below.

Reasmpling step proceeds sampling of particles with replacement according to the probabilities equivalent to the normalized weights such that

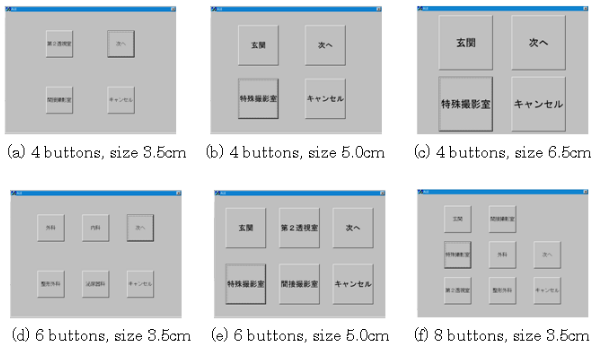
$$\tilde{\mathbf{w}}^{(i)}(1:t) \sim \begin{cases} \mathbf{w}^{(1)}(1:t) & \text{with prob. } \tilde{\alpha}_t^{(1)} \\ \mathbf{w}^{(2)}(1:t) & \text{with prob. } \tilde{\alpha}_t^{(2)} \\ \vdots & \qquad\qquad\;\; \vdots \\ \mathbf{w}^{(M)}(1:t) & \text{with prob. } \tilde{\alpha}_t^{(M)} \end{cases} \tag{12}$$

After that, we reset all the weights to the uniform value $1/M$ with letting $\mathbf{w}^{(i)}(1:t) := \tilde{\mathbf{w}}^{(i)}(1:t)$.

4. Experiment

Touch panel interface for control and guidance of intelligent wheel chair is the target user interface of proposed method. The intelligent wheel chair is aimed at a near future vehicle for handicapped person having functions of autonomous run to a desired destination, avoidance of obstacle, etc. We consider a touch panel interface for the intelligent wheel chair used inside a hospital in this experiment. As an instance of the touch panel interface, we employ a simple interface having destination buttons, next page button, and cancel button as shown in Figure 1. User will choose destination button of desired destination on this interface. There are many destinations so they cannot be displayed on one panel simultaneously, the next page button will act as it turn the page to the next panel. The cancel button plays a role to return to the first page (panel).

Adaptation on the interface is on size and the number of buttons. If the size of buttons is large, it is easy to see the label of button especially for a person having weak eye sight. However too large button leads to too small margin of buttons, and it may cause difficulty to operate the interface especially for a person having problems on their hand. Additionally, large size of button leads to the small number of buttons displayed on one panel. It requires many times turning of the pages so it is not necessarily good interface especially for a person only injured ones leg and other parts are fine. There is also a factor of perception ability of individuals. Then our method will change the size and the number of buttons using the type-based approach. We have already shown several sizes and numbers of the buttons of the interface available in the experiment in Figure 1.



(a) 4 buttons, size 3.5cm (b) 4 buttons, size 5.0cm (c) 4 buttons, size 6.5cm

(d) 6 buttons, size 3.5cm (e) 6 buttons, size 5.0cm (f) 8 buttons, size 3.5cm

Figure 1: Adaptive user interface of touch panel.

4.1 Experimental environment

Before going to experimental result, we should mention about the experimental environment. First of all, mention about the intelligent wheel chair. Although we suppose the touch panel interface so as to control an intelligent wheel chair, the wheel chair is near future technology and is still under research and development. We mainly focus on the interface of it and it is not our main purpose to develop the wheel chair in this paper. According to these situations, we emulate the function of the intelligent wheel chair by virtual reality system with two LCD projectors and polarizing glasses, and sitting down on a fixed wheel chair.

Secondly, we mention about user population. The best way to do the experiment is to have test subjects as actual users of the wheel chair, i.e., elderly people, handicapped, injured legs, and so on. Obviously it will involve hard problems for having experiment with these users at initial stage of research and development. To circumvent this difficulty, we emulate the performance of these users by wearing special equipment. The equipment is a market product aimed at experiencing the situation of elderly person for non-elderly person. Figure 2 shows a person who wear this equipment with all parts. We assume three kinds of persons; (1)elderly person (having weak eye sight and may have hand problem), (2)handicapped (having problems on one's hand but eye sight be normal), and (3)injured (only one's leg) by wearing selected parts of the equipment as shown in Table 1.

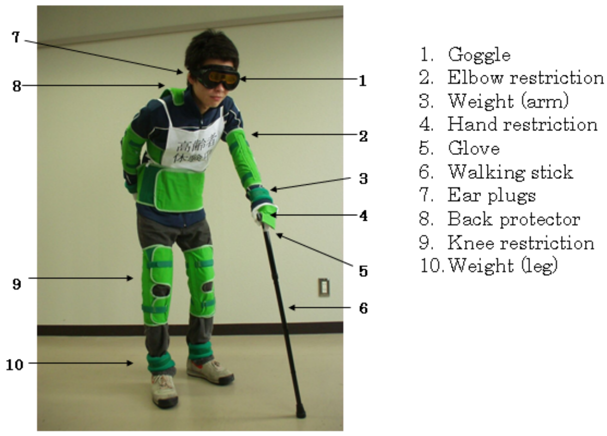


Figure 2: Constraint equipment for having experience the elderly person's situation.

Table 1: Conditions for wearing parts of the equipment to emulate three kinds of users.

Kind of user	Parts of equipment(Part No)
Elderly person	- Knee restriction (9) - Weight(leg) (10) - Elbow restriction(2) - Hand restriction(4) - Glove(5) - Weight(arm)(3) - Goggle(1) - Back protector(8)
Handicapped	- Knee restriction (9) - Weight(leg) (10) - Elbow restriction (2) - Hand restriction (4) - Glove(5) - Weight(arm) (3)
Injured	- Knee restriction (9) - Weight(leg) (10)

4.2 User type extraction

According to the procedure for extracting the user types described in section 2.1, we have conducted user type extraction experiment as follows. Beforehand the procedure, we have decided that the interface space E_U is defined as variation of button sizes consisting of threes sizes; large(6.5cm), middle(5.0cm), and small(3.5cm), and three variations of the number of buttons consisting of 4, 6, and 8 buttons, as shown in Figure 1.

First of the procedure, we have decided the observation space E_O as follows. Before the decision, we have listed up the items that we can measure during operation of the user interface; which are button press time, button press position on the panel, and kind of buttons to be pressed. Then the observation space E_O has been decided to consist of three items (1) quickness, (2) accuracy, and (3) correction rate. Quickness is measured by time interval of button presses where we only deal with short time interval less than 5 second (say). Accuracy is measured by the button press position to calculate discrepancy between the center of button and the pressed position with smaller discrepancy higher accuracy. Correction rate is measured by frequency of cancel button press. Next, we assume here that there is no external input so the input space E_Z is empty set.

Second, we have decided distribution of all components of the mixture model of eq.(1) as follows. It depends on a fitness of component models to the collected data. So it has been actually done after the data collection. Among component models consisting of Gaussian, Beta, and Gamma distributions, the best fit model having maximum likelihood among them is Gamma distribution. So we have employed Gamma distribution as the components $f_k(), k = 1, 2, \cdots, K$. Thus the parameters space E_Θ is defined as two parameters of Gamma distribution (actually its K-th product space) as a result. Next, the number of components K, which corresponds to the number of types, has been decided by following consideration. We assume that there are three major kind of users in a population in previous subsection, which are (1)elderly person, (2)handicapped, and (3)injured. Corresponding to these kinds, we assume the number of components K be three.

Third, we have conducted experiments measuring users' operations on E_O over various situations of $\mathbf{u} \in E_U$ for 12 test subjects in our laboratory. All test subjects perform three (all) kinds of users mentioned above, i.e., elderly person, handicapped, and injured, by wearing the equipment as shown in Table 1. As the various situations of $\mathbf{u} \in E_U$, we have examined all (six) the interfaces shown in Figure 1. The task of test subject is to operate the interface in order to move the virtual intelligent wheel chair to a designated destination. 8 tasks were examined by a test subject for each condition, i.e., for a certain kind of user and for a certain kind of interfaces. Thus the number of runs of the experiments is 12[test subjects]×3[kinds of users]×6[kinds of interfaces] ×8[tasks]=1728.

Fourth and finally, we have estimated the mixture model as follows. In this situation, we already know the kind of user for each run of the experiment, so we can divide the data set into three subsets of user kinds, elderly person, handicapped, and injured. Then components have been estimated each by each using the corresponding subset of the data. As the estimation of the components having Gamma distribution, we have used the moment matching method.

We mention about details of measurement. Measurement items are defined as above, i.e., (1) quickness, (2) accuracy, and (3) correction rate. We can measure one observation vector consisting of these three items by one run (i.e. one task) of the experiment. Detailed calculations of each item are as follows; Quickness Q is averaged time interval of button presses

$$Q = \frac{1}{N-1}\sum_{t=2}^{N}(p_t - p_{t-1}) \qquad (13)$$

where p_t denotes button press time (with unit 'second') of t-th operation in a task and N is the total number of operations in the task. Next, accuracy A is averaged precision of button press position with respect to the center of button calculated by

$$A = \frac{1}{N}\sum_{t=1}^{N} d_t \qquad (14)$$

where d_t is button press precision (with unit 'millimeter') of t-th operation in a task. Finally, correction rate C is calculated by averaged time interval of cancel button presses

$$C = \frac{1}{L-1}\sum_{t'=2}^{L}(c_{t'} - c_{t'-1}) \qquad (15)$$

where $c_{t'}$ denotes button press time (with unit 'second') of t'-th operation of cancel button press in a task and L is the total number of operations of cancel button press in the task.

Plot of measured data for Quickness-Accuracy and Correction rate-Accuracy are shown in Figure 3, and Figure 4, respectively. To illustrate the estimation results, we show density functions of components in Figure 5 for a case of the interface with 4 buttons of size 6.5cm, which is shown in Figure 1 (c).

4.3 User type estimation

Using the types extracted at the user type extraction part (in previous subsection), this part examines the user type estimation for the current user. A series of experiment, which is different from the experiments conducted at the user type extraction part, has been

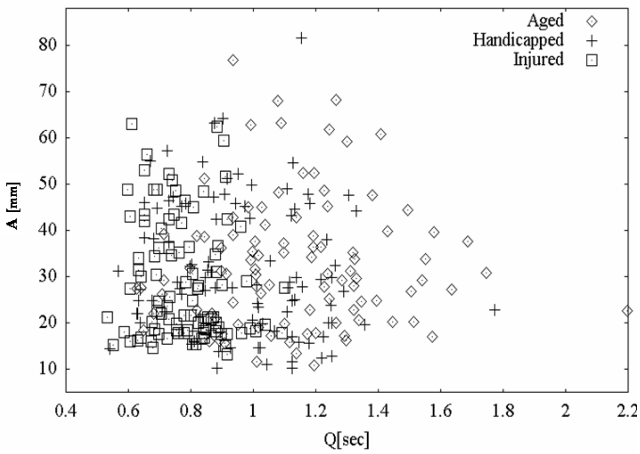


Figure 3: Measured data for user type extraction (Quickness-Accuracy).

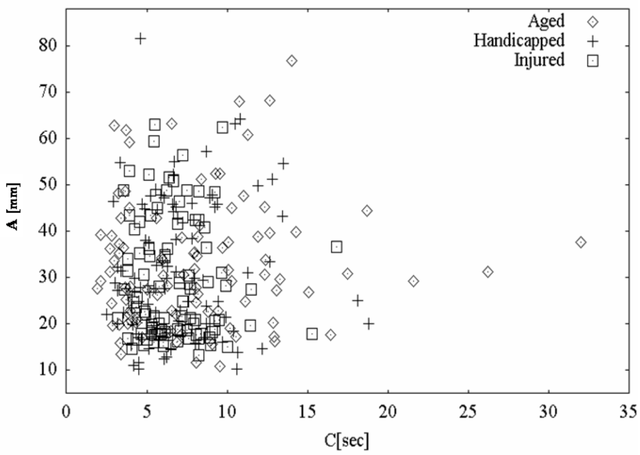
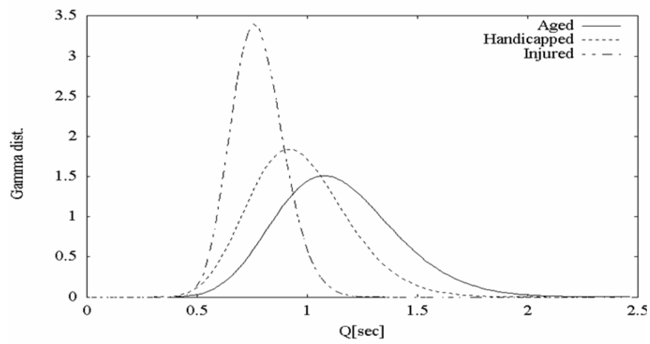


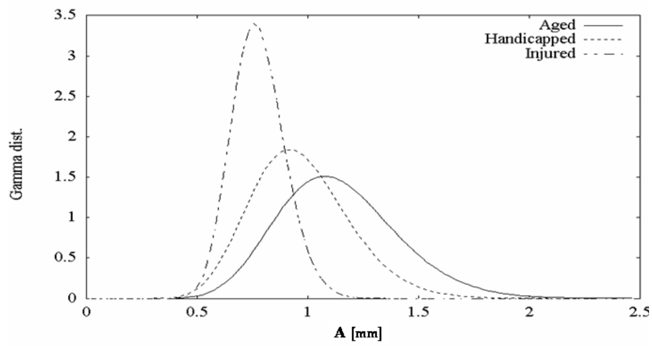
Figure 4: Measured data for user type extraction (Correction rate-Accuracy).

done as follows. A new user performed the emulation of three kinds of user, i.e. elderly person, handicapped, and injured by wearing the equipment as shown in Table 1. Then, the user operated the touch panel by displaying a certain interface among six interfaces shown in Figure 1, where all six interfaces have been examined individually. Task for the user is similar to one of the user type extraction experiment, i.e., moving the virtual intelligent wheel chair to a designated destination. 8 tasks has been executed for each kind of user emulation.

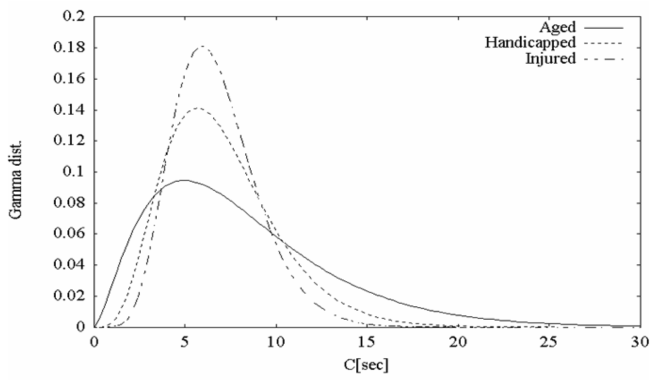
In this part, we have used performance measurement directly with quickness, accuracy, and correction rate instead of their averaged version used in user type extraction part (equations (13) (14), and (15)). That is, button press time interval $\tilde{Q} = p_t - p_{t-1}$, is used as quickness, button press precision $\tilde{A} = d_t$ is used as accuracy, and time interval of cancel button presses $\tilde{C} = c_{t'} - c_{t'-1}$ is used as correction rate.



(a) Quickness



(b) Accuracy



(c) Correction rate

Figure 5: Estimated densities of components in user type extraction.

During the operation of the interface, we have estimated the type of the user by applying the method shown in section 2.2 with the estimation method shown in section 3. Conditions for the estimation are as follows. First, as the state space for the estimation, we have not used the space of weights $E_S = [0,1]^K$ directly since we need to have special care to do not violate the constrains for weights (non-negative and sum up to 1). Instead, we have employed a space of log-transformed weights, denoted by $\mathbf{W}(t) = (W_1(t), W_2(t), \cdots, W_K(t))$, with $W_k(t) = \log w_k(t)$. As the smoothness prior for weight vector represented by a Markov transition model $p(\mathbf{w}(t)|\mathbf{w}(t-1))$, we have used a model derived from equations of transition and normalization

$$\tilde{W}_k(t) = W_k(t-1) + V_k(t), \qquad V_k(t) \sim (0, \tau^2) \quad (16)$$

$$W_k(t) = \tilde{W}_k(t) / \sum_{j=1}^{K} \tilde{W}_j(t) \qquad (17)$$

for $k = 1, 2, \cdots, K$. Second, the number of particles used in the particle filters is set to 1000. Third, as the initial distribution, $p(\mathbf{w}; \hat{\mathbf{w}}_0)$, is set to a distribution derived from the log-transformed version of the initial distribution where weights are according to normal distribution with mean at the uniform weights and diagonal covariance matrix with all diagonal parts be τ_0^2. Forth, as proposal distribution, we have employed one-step-ahead prediction $p(\mathbf{w}(t)|\mathbf{w}(t-1), \mathbf{x}(1:t-1))$ thus the algorithm is reduced to the simple particle filter, called Monte Carlo filter (MCF) [7] or bootstrap filter [3]. Its weight values are calculated simply by likelihood function. Finally, as the resampling, we simply apply the resampling procedure for all iteration without evaluating the effective sample size $ESS(t)$.

Results of user type estimation are summarized in Table 2, where performance of type estimation is evaluated with 5 grades, excellent, good, well, confusion, and bad. Typical results of these grades are shown in Figure 6, Figure 7, Figure 8, Figure 9, and Figure 10.

4.4 Adaptation of interface

Adaptation scheme explained in section 2.3 has been implemented to the touch panel interface with the mechanism of user type estimation in the previous subsection. To synthesize the optimal interface for the current user, we will use eq.(5), weighted sum of optimal interfaces for each type, as follows. First, in order to have the optimal interfaces of all types, we have conducted inquiry for all test subjects of user type extraction experiment. The inquiry is simple, just choose one interface among all (which are shown in Figure 1) after conducted full series of task with emulating a certain kind of user among three (elderly person, handicapped, and injured). Then we have obtained preference rate over interfaces for each kind of user as shown in Table 3.

By using the result of Table 3 as performance index over interfaces for each kind of user (i.e., type), which are denoted by $I_k(\mathbf{u})$ with $\mathbf{u} \in E_U = \{1, 2, \cdots, 6\}$ for $k = 1, 2, 3$, we form a total performance index to choose the optimal user interface

$$I(\mathbf{u}) = \sum_{k=1}^{K} \hat{w}_k(t) I_k(\mathbf{u}) \qquad (18)$$

as a realization of conceptual form of eq.(5). Then the interface that attains the maximum value of eq.(18) will be displayed on the touch panel.

There is a choice of timing when the interface adapts to user. Frequent changes of the interface may cause difficulty to use the interface, so we employ the timing to change it when one task has been completed or cancel button were pressed. Then the first time the

Table 2: Summary of estimation results of user type.

Button size[cm]	Number of buttons			Kind of user
	4	6	8	
3.5	excellent	excellent	confusing	Elderly person
	well	confusing	confusing	Handicapped
	well	confusing	confusing	Injured
5.0	bad	well	—	Elderly person
	confusing	good	—	Handicapped
	excellent	excellent	—	Injured
6.5	good	—	—	Elderly person
	bad	—	—	Handicapped
	confusing	—	—	Injured

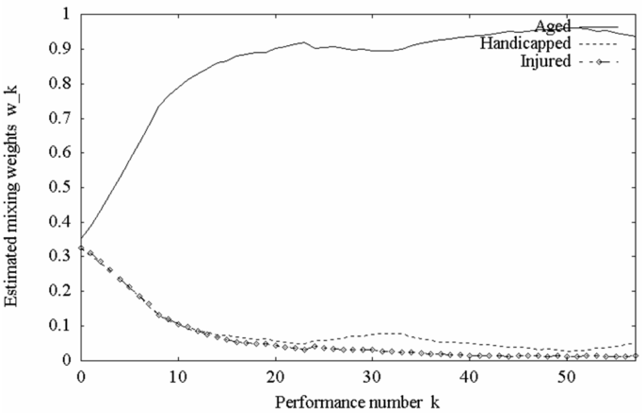


Figure 6: Estimation result of user type for 4-buttons size 3.5[cm] interface by elderly person user, typical case of 'excellent' graded.

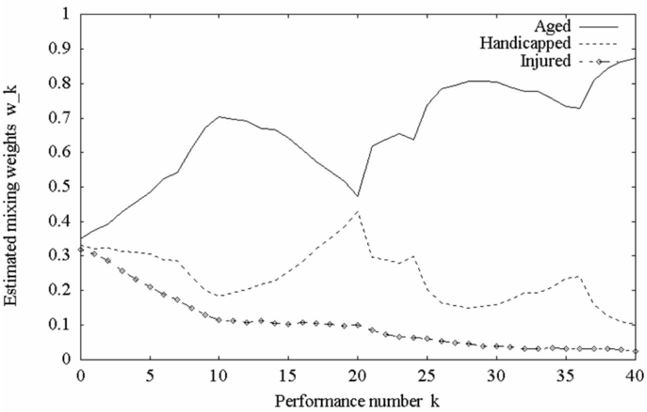


Figure 7: Estimation result of user type for 4-buttons size 6.5[cm] interface by elderly person user, typical case of 'good' graded.

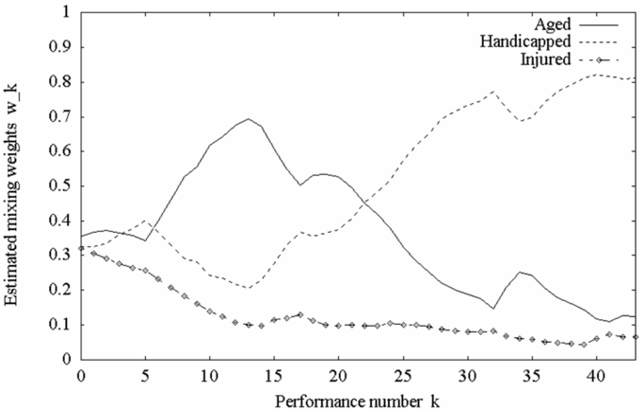


Figure 8: Estimation result of user type for 4-buttons size 3.5[cm] interface by handicapped user, typical case of 'well' graded.

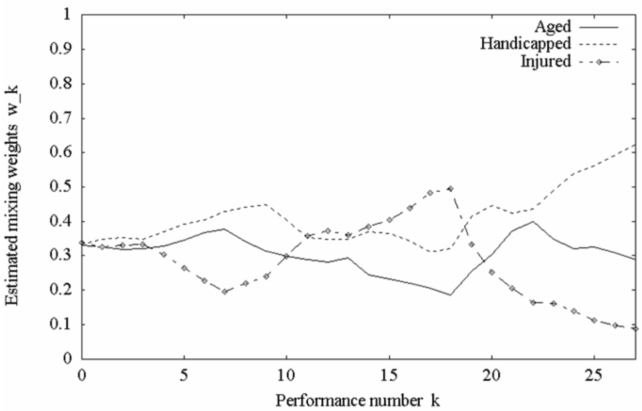


Figure 9: Estimation result of user type for 8-buttons size 3.5[cm] interface by handicapped user, typical case of 'confusing' graded.

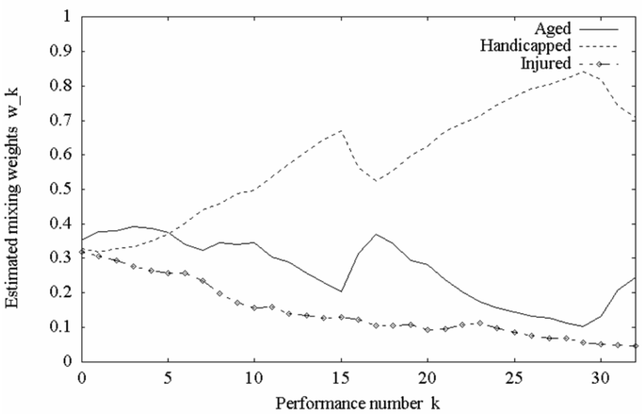


Figure 10: Estimation result of user type for 4-buttons size 5.0[cm] interface by elderly person user, typical case of 'bad' graded.

Table 3: Inquiry result of preference of interfaces.

Interface			Kind of user (type)		
Inter-face No.	Number of buttons	Button size[cm]	Elderly person	Handi-capped	Injured
1	4	3.5	0.00	0.00	0.00
2	4	5.0	0.16	0.04	0.16
3	4	6.5	0.40	0.16	0.00
4	6	3.5	0.00	0.00	0.00
5	6	5.0	0.36	0.72	0.56
6	8	3.5	0.08	0.08	0.28

user will see the changed interface is always the same situation where the first page of the panel is displayed.

Adaptation experiment has been conducted as follows. Three kinds of users, emulating elderly person, handicapped, and injured, have operated the adaptive user interface of touch panel with tasks moving the virtual intelligent wheel chair to designated destinations. 15 tasks were performed for each kind of users. Results of adaptation are as follows. User type estimation results are shown in Figure 11, Figure 12, and Figure 13, respectively for elderly person, handicapped, and injured. Adaptation result is depicted by plotting the time evolution of interface number of the interface displayed to the user. The results are shown in Figure 14, Figure 15, and Figure 16. Here symbols show the timing of adaptation caused by completion of destination selection or cancel button press.

By looking at the results of user type estimation in adaptation, two of three kinds of users are correctly estimated their user type, while one (handicapped user) is not correct as shown in Figure 12. However looking more detail at result in Figure 12, we recognize that at final part of the series it can estimate the type of user correctly.

Looking at the adaptation results, we can interpret the results as follows. First, for elderly person user case, interface no.5 was displayed at the beginning part, but it became to show interface no.3 instead of no.5. Note that no.3 interface is the most preferred interface for elderly person user as shown in Table 3. Second, for handicapped user case, it initially displayed no.5 interface stably, then it became to try to show no.3 interface, and eventually it returned to no.5 interface with changes of user type estimation to correct one. Here no.5 interface is the most preferred one for handicapped user as shown in Table 3. Third and finally, for injured user case, it stably displayed interface no.5, which is the most preferred one as shown in Table 3.

5. Conclusion

We have proposed a new method for adaptive user interface in general form and applied it to a touch panel user interface which is used to select the destination of intelligent wheel chair. The method consists of three parts, user type extraction, user type estimation, and adaptation of interface. Mixture model is used at the user type extraction part. At the user type estimation part, we have proposed a state space model consisting of system equation of smoothness prior to time evolution of mixing weights and observation equation of mixture model with the time varying mixing weight. State estimation is conducted by particle filters to obtain the user type estimation in a form of conditional distribution of the time-varying mixing weight given the series of observation up to current time. At the adaptation of interface part, the estimated types are used to synthesize the optimal interface by combining the optimal interfaces for each user type. Through experiments of adaptive touch panel interface for intelligent wheel chair, we have explored how the proposed method works and obtained result matched to preference of users.

Acknowledgements

This study was carried out under the ISM Cooperative Research Program (2004-ISM·CPR-2020).

References

[1] A.Dempster, N.Laird, and D.Rubin, "Maximum likelihood estimation from incomplete data via EM Algorithm", *Journal of Royal Statistical Society*, B, vol.39, pp.1-38, 1977.

[2] A.Doucet, J.F.G.de Freitas, and N.J.Gordon (eds), *Sequential Monte Carlo Methods in Practice*, New York, Springer, 2001.

[3] N.J.Gordon, D.J.Salmond, and A.F.M.Smith: "Novel approach to nonlinear / non-Gaussian Bayesian State Estimation", *IEE Proceedings-F*, **140**, No.2, pp.107-113, 1993.

[4] T.Hyakudome, H.Baba, N.Ikoma, and H.Maeda, "Adaptive Touch Panel Interface for Intelligent Wheel Chair using Online Estimation of User Type by Time-Varying Mixing Weights Model", *Abstract. of the Brain-inspired Information Technology (BrainIT2004), The international Symposium on Bio-Inspired Systems*, p.73, 2004.

[5] T.Hyakudome, N.Ikoma, and H.Maeda, "An adaptive user interface based on online estimation of time-varying mixing weight of mixture model by particle filters", *Proc. of –the 30th Anniversary of the Information Criterion (AIC)–, Science of Modeling*, The Institute of Statistical Mathematics, Yokohama, Japan, Dec.14-17, pp.374-375, 2003.

[6] N.Ikoma, W.Pedrycz, K.Baba, T.Hyakudome, Y.Matsumoto, N.Nakamura, and H.Maeda, "User type identification by mixing weight estimation of mixture models based on State Space Modeling", *Proc. of the 2003 IEEE International Symposium on Virtual Environments, Human-Computer Interfaces, and Measurement Systems (VECIMS 2003)*, Lugano, Switzerland, July 27-29, pp.20-25, 2003.

[7] G.Kitagawa: "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models", *Journal of Computational and Graphical Statistics*, **5**, No.1, pp.1-25, 1996.

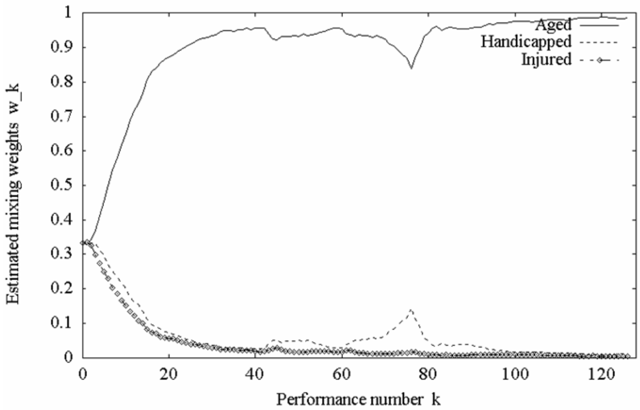


Figure 11: Estimated weight with adaptation of interface by elderly person user.

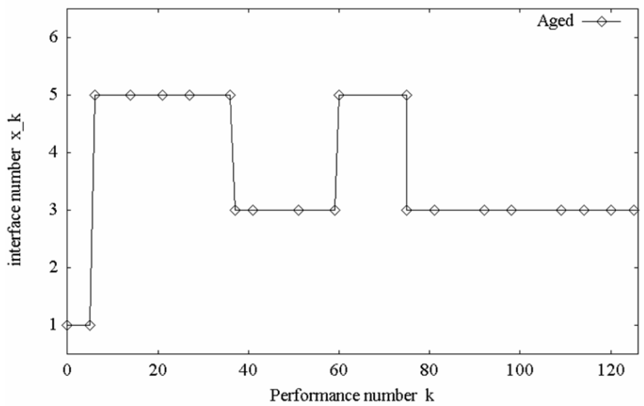


Figure 14: Adaptation result of interface by elderly person user.

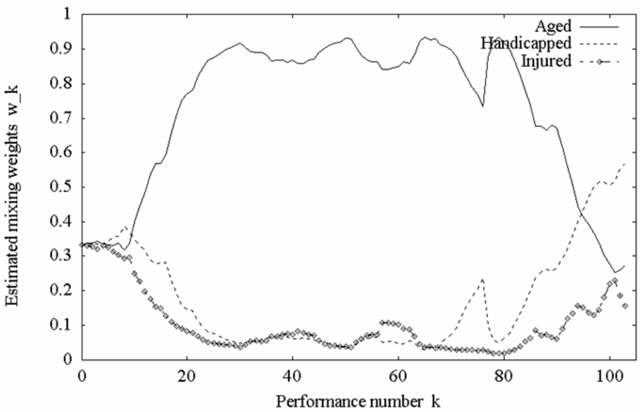


Figure 12: Estimated weight with adaptation of interface by handicapped user.

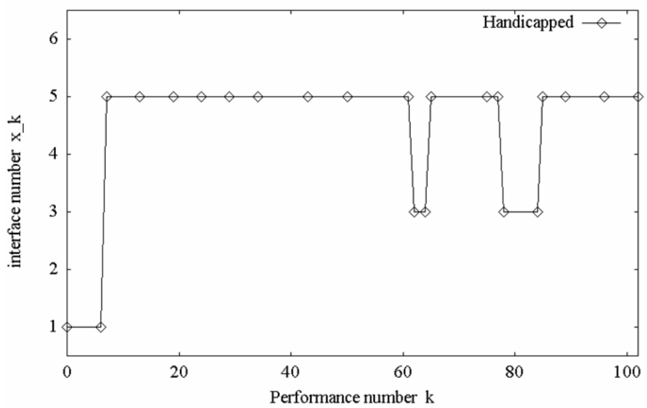


Figure 15: Adaptation result of interface by handicapped user.

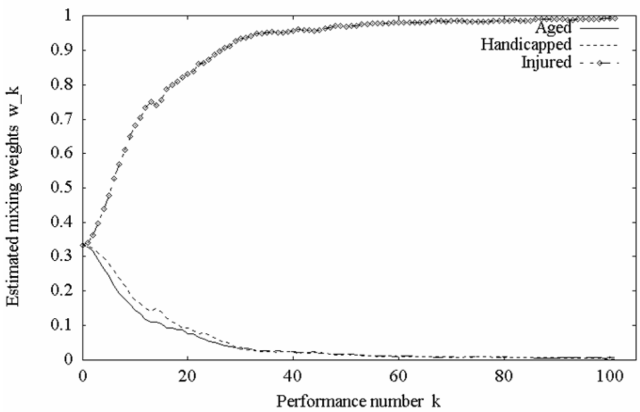


Figure 13: Estimated weight with adaptation of interface by injured user.

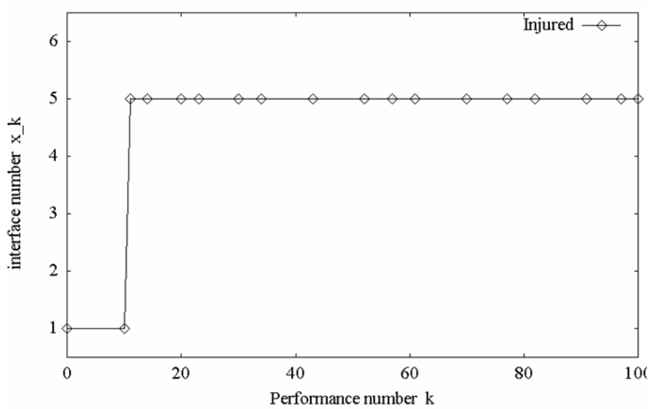


Figure 16: Adaptation result of interface by injured user.