A Novel Dual-arm Motion Discrimination Method Using Recurrent Probability Neural Networks for Automatic Gesture Recognition

Yuki HIRAMATSU, Taro SHIBANOKI, Keisuke SHIMA and Toshio TSUJI

***Abstract*— As gestures are mainly characterized via combinations of left and right arm movements, automatic gesture recognition requires accurate identification of separate arm motions. This paper proposes a novel dual-arm motion discrimination method that combines posterior probabilities estimated independently for left and right arm movements. In this approach, the posterior probabilities of individual single-arm motions are first estimated from measured biological signals using recurrent probabilistic neural networks. Then, the estimated posterior probabilities are combined automatically based on the motion dependency that exists between the arms, making it possible to calculate the joint posterior probability of dual-arm motions. With this method, all dual-arm motions consisting of individual single-arm movements can be discriminated through learning of single-arm motions only.**

In the experiments performed, the proposed method was applied to the discrimination of 15 dual-arm motions made up of three movements for each arm. The results showed that the method enables high discrimination performance based on learning of only three motions for each arm (average discrimination rate: 98.80 ± 0.68%).

I. INTRODUCTION

Gestures are a form of nonverbal communication in which visible bodily actions are used to convey the various messages, either in lieu of speech or in combination with it [1]. They include movements of the hands, face or other parts of the body. Estimation of human intentions based on these movements can be applied to the development of communication tools such as gesture translators and human-machine interfaces. However, high-accuracy discrimination is difficult due to the large variety of possible gestures.

The purposes of the numerous studies carried out so far on gesture discrimination can be classified as static gesture discrimination and dynamic gesture discrimination [2]−[7]. Static gestures are based on hand/arm shapes, and dynamic gestures use hand/arm motions. In regard to static gestures, Okamoto *et al*. [2] studied the discrimination of 31 hand shapes using a probabilistic neural network (PNN) based on finger joint angle signals recorded by shape sensors, and Bowden *et al*. [3] proposed a method to recognize 26 gestures from video images using a hidden Markov model (HMM).

In dynamic gesture discrimination, Sawada *et al*. [5] applied a dynamic programming (DP) matching method to classify such gestures based on the characteristics of hand velocities, distances and arm joint angles calculated using acceleration sensors. Vogler *et al*. [6] also studied the discrimination of 53 motions using HMM and DP matching based on arm positions, velocities and angles extracted from recorded video images. However, these studies focused exclusively on single-arm motion discrimination. As gestures include both left and right arm motions, it is necessary to discuss a discrimination method for dual-arm motions in addition to considering single-arm motions.

For the dual-arm motion discrimination problem, Kim *et al*. [7] focused on and investigated the discrimination of 25 dual-arm dynamic gestures using a fuzzy min-max neural network based on hand trajectories and finger angles measured using data gloves. However, since dual-arm motions encompass a wide variety of single-arm movement combinations, a large number of samples are needed to train the neural network, making the learning time potentially excessive.

This paper proposes a novel approach based on a joint posterior probability neural network (J-PNN) for dual-arm motion discrimination. The J-PNN consists of two PNNs, each of which estimates the posterior probability of single-arm motion, and can be used to estimate the joint posterior probability of dual-arm motions by combining the posterior probabilities of individual arm motions based on the dependencies between both arms. The network can thus discriminate all dual-arm motions consisting of individual single-arm motions based on learning of single-arm motions only, making it possible to increase the number of motions with no corresponding increase in the learning time or the number of learning samples required.

The structure and learning algorithm of the proposed neural network are explained in Section II, and the details of the dual-arm motion discrimination method based on the J-PNN are shown in Section III. Section IV outlines dual-arm motion experiments performed to investigate the motion discrimination performance of the J-PNN.

II. A JOINT POSTERIOR PROBABILITY NEURAL NETWORK (J-PNN)

This paper proposes a novel probabilistic NN that estimates the joint posterior probability of dual-arm motions by combining posterior probabilities estimated independently for left and right arm motions. The network structure and the learning algorithm are outlined in the following subsections.

Y. HIRAMATSU, T. SHIBANOKI and T. TSUJI are with Graduate School of Engineering, Hiroshima University, 1-4-1 Kagamiyama Higashi-Hiroshima Hiroshima, JAPAN hiramatsu-y@bsys.hiroshima-u.ac.jp

K. SHIMA is with Graduate School of Biomedical Science, Hiroshima University, 1-2-3 Kasumi Minami Hiroshima, JAPAN

A. Network structure

The structure of the J-PNN is shown in Fig. 1. The network consists of the two parts of time series pattern discrimination and joint probability estimation. In the time series pattern discrimination stage, the posterior probability $p(c_\alpha|\boldsymbol{x}_\alpha)$ for each class $c_\alpha(c_\alpha = 1, \ldots, C_\alpha; \alpha \in \{1, 2\})$ is estimated from time series vectors $\boldsymbol{x}_\alpha(t)$. These probabilities are then combined to calculate the joint posterior probability after modification based on the dependencies between two input vectors. The network therefore allows the estimation of posterior probability for all classes realized by combining the two input vectors. In other words, by learning for left and right arm motions independently, the network can estimate a dual-arm motion's probability. The details of each part are outlined below.

1) Time series pattern discrimination: The time series pattern discrimination part consists of dual R-LLGMNs (a type of recurrent probabilistic neural network)[9]. It involves a Gaussian mixture model (GMM) and an HMM in the structure, and copes with the time-varying characteristics of input signals. Two feature vectors $\boldsymbol{x}_\alpha(t) \in \Re^{D_\alpha}$ ($\alpha \in \{1, 2\}$; $t = 1, 2, \ldots, T$), where T is the time length of the input vector, are first input to the network independently for discrimination of individual arm motions.

In order to represent a normalized distribution corresponding to each component of the GMM as weight coefficients of the R-LLGMN, these input vectors $\boldsymbol{x}_\alpha(t)$ are converted to form the modified input vector $\boldsymbol{X}_\alpha(t)$ as follows:

$$\begin{aligned}\boldsymbol{X}_\alpha(t) \;\; = \;\; & [1, \boldsymbol{x}_\alpha(t)^{\mathrm{T}},\ x_{\alpha,1}(t)^2,\ x_{\alpha,1}(t)x_{\alpha,2}(t),\ \ldots, \\ & x_{\alpha,1}(t)x_{D_\alpha}(t),\ x_{\alpha,2}(t)^2,\ x_{\alpha,2}(t)x_{\alpha,3}(t),\ \ldots, \\ & x_{\alpha,2}(t)x_{\alpha,D_\alpha}(t),\ \ldots,\ x_{\alpha,D_\alpha}(t)^2]^{\mathrm{T}}. \qquad (1)\end{aligned}$$

The first layer of the network consists of $U_\alpha = 1 + D_\alpha(D_\alpha + 3)/2$ units, which correspond to the dimensions of the input vector $\boldsymbol{X}_\alpha(t) \in \Re^{U_\alpha}$, and the identity function is used for the activation function of each unit.

The outputs of the first layer multiplied by weight ${}^{(2)}w^{c}_{\alpha,k',k,m,u}$ are transmitted to the second layer. The relationship between the input and the output of the second layer is defined as:

$${}^{(2)}I^{c_\alpha}_{\alpha,k'_\alpha,k_\alpha,m_\alpha}(t) = \sum_{h_\alpha=1}^{U_\alpha} {}^{(1)}O_{\alpha,u_\alpha}(l) w^{c_\alpha}_{\alpha,k'_\alpha,k_\alpha,m_\alpha,u_\alpha} \quad (2)$$

$${}^{(2)}O^{c_\alpha}_{\alpha,k'_\alpha,k_\alpha,m_\alpha}(t) = \exp\Big({}^{(2)}I^{c_\alpha}_{\alpha,k'_\alpha,k_\alpha,m_\alpha}(t)\Big), \quad (3)$$

where C_α is the number of classes, $K_{\alpha,c}$ is the number of states in the HMM, and $M_{\alpha,c_\alpha,k_\alpha}$ is the number of components of the Gaussian mixture distribution in class c_α and state k_α.

The input of unit $\{c_\alpha, k_\alpha, k'_\alpha\}$ in the third layer integrates the output of units $\{c_\alpha, k_\alpha, k'_\alpha, m_\alpha\}$ $(m_\alpha = 1, \ldots, M_{\alpha,c_\alpha,k_\alpha})$ in the second layer, and the output of the fourth layer is also fed back to the third layer. These are expressed as follows:

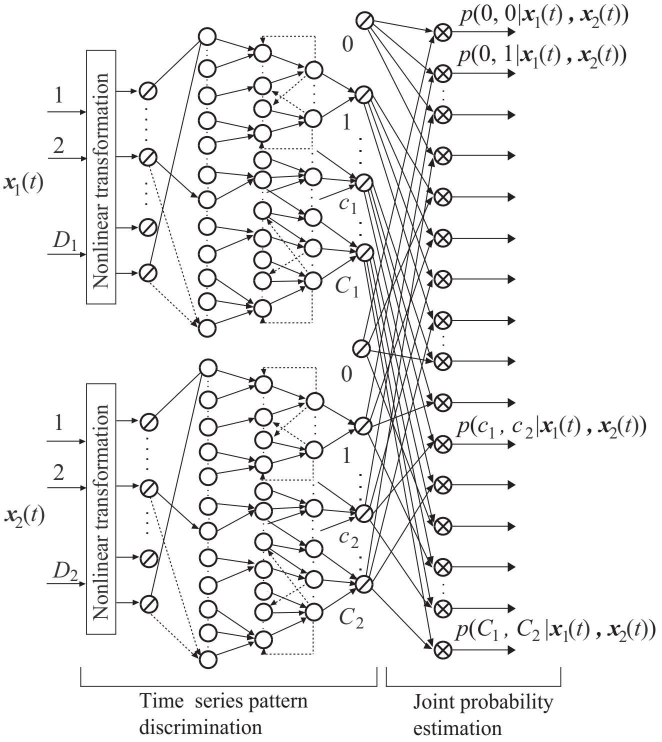

$p(0, 0|\boldsymbol{x}_1(t), \boldsymbol{x}_2(t))$
$p(0, 1|\boldsymbol{x}_1(t), \boldsymbol{x}_2(t))$
1
2
$\boldsymbol{x}_1(t)$
D_1
Nonlinear transformation
0
1
c_1
C_1
0
1
2
$\boldsymbol{x}_2(t)$
D_2
Nonlinear transformation
1
c_2
C_2
$p(c_1, c_2|\boldsymbol{x}_1(t), \boldsymbol{x}_2(t))$
$p(C_1, C_2|\boldsymbol{x}_1(t), \boldsymbol{x}_2(t))$
Time series pattern discrimination
Joint probability estimation

Fig. 1. Structure of the joint posterior probability neural network

$${}^{(3)}I^{c_\alpha}_{\alpha,k'_\alpha,k_\alpha}(t) = \sum_{m_\alpha=1}^{M_{\alpha,c_\alpha,k_\alpha}} {}^{(2)}O^{c_\alpha}_{\alpha,k'_\alpha,k_\alpha,m_\alpha}(t) \quad (4)$$

$${}^{(3)}O^{c_\alpha}_{\alpha,k'_\alpha,k_\alpha}(t) = {}^{(4)}O^{c_\alpha}_{\alpha,k'_\alpha}(t-1)\,{}^{(3)}I^{c_\alpha}_{k'_\alpha,k_\alpha}(t), \quad (5)$$

where${}^{(4)}O^{c_\alpha}_{k'_\alpha}(0) = 1.0$ is for the initial state.

The relationships in the fourth layer are defined as

$${}^{(4)}I^{c_\alpha}_{\alpha,k_\alpha}(t) = \sum_{k'_\alpha=1}^{K_{\alpha,c}} {}^{(3)}O^{c_\alpha}_{\alpha,k'_\alpha,k_\alpha}(t) \quad (6)$$

$${}^{(4)}O^{c_\alpha}_{\alpha,k_\alpha}(t) = \frac{{}^{(4)}I^{c_\alpha}_{\alpha,k_\alpha}(t)}{\sum_{c'_\alpha=1}^{C_\alpha} \sum_{k'_\alpha=1}^{K_{\alpha,c'_\alpha}} {}^{(4)}I^{c'_\alpha}_{k'_\alpha}(t)}. \quad (7)$$

Finally, unit c in the fifth layer integrates the outputs of $K_{\alpha,c}$ units $\{c_\alpha, k_\alpha\}$ $(k_\alpha = 1, \ldots, K_{\alpha,c})$ in the fourth layer. The relationships in this layer are defined as

$${}^{(5)}O^{c_\alpha}_\alpha(t) = {}^{(5)}I^{c_\alpha}(t) = \sum_{k_\alpha=1}^{K_{\alpha,c}} {}^{(4)}O^{c_\alpha}_{\alpha,k_\alpha}(t) \quad (8)$$

$${}^{(5)}O^{c_\alpha}_\alpha(t) = {}^{(5)}I^{c_\alpha}(t). \quad (9)$$

Here, ${}^{(5)}O^{c_\alpha}_\alpha(t)$ indicates posterior probability $p(c_\alpha|\boldsymbol{x}_\alpha(t))$ $(\alpha \in \{1, 2\}; c_\alpha = 1, \ldots, C_\alpha)$ for class c_α.

In the J-PNN, the calculations in the third and fourth layers are associated with feedback connections, so time-varying features in the time series can be used. The R-LLGMN can therefore be adopted to model the posterior probability of each class through GMM-based learning.

2) Joint probability estimation: The Joint probability estimation part estimates joint posterior probability $p(c_1, c_2|\boldsymbol{x}_1(t), \boldsymbol{x}_2(t))$ as

$$p(c_1, c_2|\boldsymbol{x}_1(t), \boldsymbol{x}_2(t)) = \prod_{\alpha\in\{1,2\}} \frac{\eta_{\alpha,c_\alpha} p(c_\alpha|\boldsymbol{x}_\alpha(t))}{\sum_{c_\alpha=0}^{C_\alpha} \eta_{\alpha,c_\alpha} p(c_\alpha|\boldsymbol{x}_\alpha(t))}. \tag{10}$$

Here, $c_\alpha = 0, 1, \ldots, C_\alpha$, where $c_\alpha = 0$ is defined as a complementary event class, and the modifying coefficient of posterior probability η_{α,c_α} calculated based on entropy H_α, which indicates the ambiguity of each posterior probability, is defined as:

$$\eta_{\alpha,c_\alpha} = \begin{cases} 1 - H_\alpha & (\{H_\alpha > H_{th}|(c_1, c_2) \in \Xi\}) \\ 1 & (\text{otherwise}) \end{cases} \tag{11}$$

$$H_\alpha = -\frac{1}{\log C_\alpha} \sum_{c_\alpha=0}^{C_\alpha} p(c_\alpha|\boldsymbol{x}_\alpha(t)) \log p(c_\alpha|\boldsymbol{x}_\alpha(t)). \tag{12}$$

Here, Ξ represents a set consisting of a combination of two class types (c_1 and c_2), for which one class may influence the discrimination result of the other, and is defined in advance (these are referred to as *dependent classes*). This means that Eq. (11) modifies the posterior probability of the class belonging to the dependent class set Ξ when individual posterior probabilities are ambiguous. Using Eqs. (10) and (11), the J-PNN can be adopted to estimate joint posterior probability based on the dependency between the classes.

Further, $p(0|\boldsymbol{x}_\alpha(t))$ is a complementary event probability, and is defined as

$$p(0|\boldsymbol{x}_\alpha(t)) = \begin{cases} 1 & (\boldsymbol{x}_\alpha(t) \in \{\phi\}) \\ 0 & (\boldsymbol{x}_\alpha(t) \notin \{\phi\}) \end{cases}, \tag{13}$$

Generally, the PNN cannot calculate posterior probability without input (i.e., a vector with no input: $\boldsymbol{x}_\alpha(t) \in \{\phi\}$). The J-PNN outputs complementary event probability $p(0|\boldsymbol{x}_\alpha(t)) \in \{0, 1\}$ according to the state of input vector $\boldsymbol{x}_\alpha(t)$. If an input vector is empty , the complementary event probability becomes 1. Then, according to Eqs. (10)−(13), the J-PNN directly outputs the posterior probability for the other input. On the other hand, if an input vector is not empty ($\boldsymbol{x}_\alpha(t) \in \{\phi\}$), the complementary event probability become zero. The number of discriminated classes J is $J = (C_1 + 1)(C_2 + 1) - 1$.

B. Learning algorithm

The learning of the J-PNN can be done through the learning process for each network. Here, a set of vectors $\boldsymbol{x}_\alpha(t)^{(n)}$ ($n = 1, 2, \ldots, N_\alpha; t = 1, 2, \ldots, T_\alpha$) and teacher vectors $\boldsymbol{P}_\alpha^{(n)} = [P_1^{(n)}, \ldots, P_{c_\alpha}^{(n)}, \ldots, P_{C_\alpha}^{(n)}]^{\mathrm{T}}$ ($\alpha \in \{1, 2\}$) are given for the training of the J-PNN. If the vector $\boldsymbol{x}_\alpha(t)^{(n)}$ is set for class c_α, then $P_{c_\alpha}^{(n)} = 1$, and $P_{\hat{c}_\alpha}^{(n)} = 0$ ($\hat{c}_\alpha \neq c_\alpha$) for all other classes in this subset. The energy function for the network is defined as

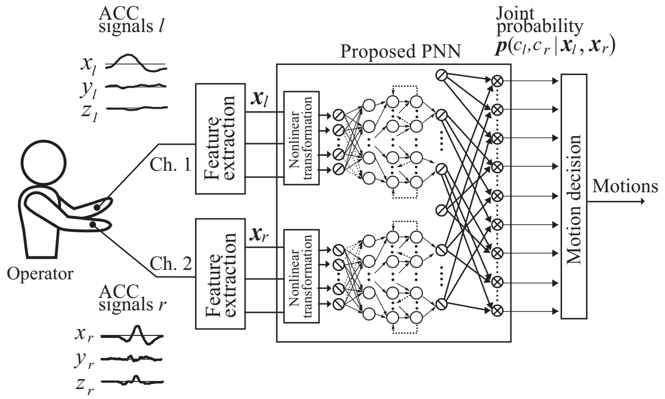


Fig. 2. Overview of the discrimination method for dual-arm motions

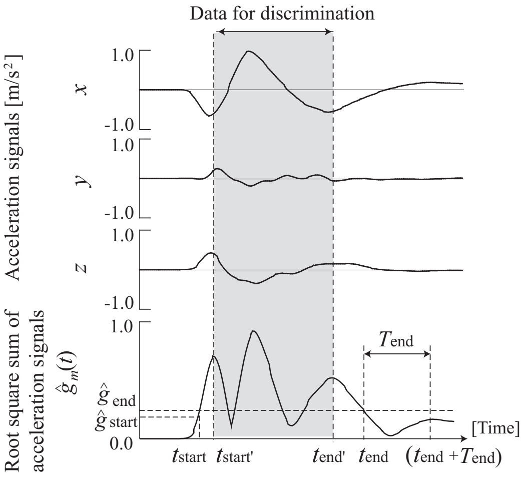


Fig. 3. Example of acceleration signals

$$E_\alpha = -\sum_{n=1}^{N_\alpha} \sum_{c_\alpha=1}^{C_\alpha} P_{c_\alpha}^{(n)} \log {}^{(5)}O^{(c_\alpha)}(T_\alpha)^{(n)}, \tag{14}$$

and learning is performed to minimize $E_\alpha$in order to maximize likelihood. Here, ${}^{(5)}O^{(c_\alpha)}(T_\alpha)^{(n)}$ is the posterior probability at given time T_α for time series patterns $\boldsymbol{x}_\alpha(t)^{(n)}$.

Due to the recurrent connection in the J-PNN, the backpropagation through time (BPTT) [10] algorithm is used, and the dynamic of the terminal attractor (TA) [11] is incorporated in the learning rule in order to regulate the convergence time of learning.

III. Dual-arm Motion Discrimination Method

The structure of the discrimination method for dual-arm motions is shown in Fig. 2. This approach involves first extracting the features of individual arm motions from ACC signals and estimating dual-arm motion from feature vectors using the J-PNN.

Three-axies ACC sensors are used to measure the acceleration signals of dual-arm motions.

The signals measured are digitized using an A/D converter and defined as $a_m^v(t) (m \in \{r, l\}, v \in \{x, y, z\})$. Here, r and l represent the indexes of the right and left hands, respectively, and x, y, and z are the axis indexes of the acceleration sensors. As the frequency components of motion acceleration signals $g_m^v(t)$ are low, they can be separated in the frequency region using a digital Butterworth low-pass filter (cut-off frequency: f_c [Hz]). The $g_m^v(t)$ values are then normalized to make the maximum value equal to 1 as follows:

$$\hat{g}_m^v(t) = \frac{g_m^v(t)}{\max g_m^v}, \tag{15}$$

where $\max g_m^v$ is the maximum value of $g_m^v(t)$. The magnitude of vector $\hat{\boldsymbol{g}}_m(t) = [g_m^x(t), g_m^y(t), g_m^z(t)]^{\mathrm{T}}$ is calculated and used as a movement detection index:

$$\hat{g}_m(t) = \sqrt{\hat{g}_m^x(t)^2 + \hat{g}_m^y(t)^2 + \hat{g}_m^z(t)^2}. \tag{16}$$

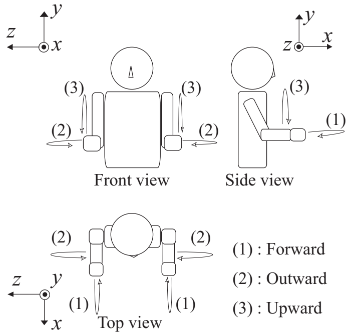
An example of the movement detection duration is shown in Fig. 3. Here, two thresholds - $\hat{g}_{\mathrm{start}}$ (the motion start threshold) and $\hat{g}_{\mathrm{end}}$ (the motion end threshold) - are used to extract the motion duration. The point at which $\hat{g}_m(t)$ first exceeds $\hat{g}_{\mathrm{start}}$ is defined as gesture start time t_{start}, and the initial instance at which it is less than $\hat{g}_{\mathrm{end}}$ during T_{end} [s] is defined as motion end time t_{end}. The time interval from the first to the last local maximum values in the duration of $[t_{\mathrm{start}}, t_{\mathrm{end}}]$ is defined as the movement duration $[t_{\mathrm{start}'}, t_{\mathrm{end}'}]$, and values of $\hat{\boldsymbol{g}}_m(t) \in \Re^3$ in this duration are extracted as discrimination data. The discrimination data are then resampled in such a way that the number of samples becomes S, and normalized such that the maximum value of $\hat{g}_m(t)$ in the duration $[t_{\mathrm{start}'}, t_{\mathrm{end}'}]$ is 1. The resulted vector $\boldsymbol{x}_m(t) (m \in \{r, l\})$ is finally defined as a feature vector. The feature pattern vectors for each arm motion $\boldsymbol{x}_m(t)$ and $\boldsymbol{x}_{m'}(t) (m' \in \{r, l\} \neq m)$ are input to the J-PNN, and the network then estimates the dual-arm motion posterior probabilities. However, since dual-arm motion involves a time lag between individual arm movements, $\boldsymbol{x}_{m'}(t)$ is set as $\boldsymbol{x}_{m'}(t) = \{\phi\}$ if $\boldsymbol{x}_{m'}(t)$ is not input within the T [s] interval after $\boldsymbol{x}_m(t)$ is input to the J-PNN.

As it is difficult to move both arms at the same time not in a point or line symmetric manner [8], feature vectors of asymmetric dual-arm motions may be skewed and become closer to be symmetric. In this paper, therefore, combinations of right and left arm motions $(c_r, c_l | c_r = c_l)$ are defined as the member of the set Ξ (the dependent class of dual-arm motions) in Eq. (11). The class with the maximum value of joint probability $p(c_r, c_l | \boldsymbol{x}_r(t), \boldsymbol{x}_l(t))$ (see Eq. (10)) is thus taken as the discrimination result.

IV. Experiments

Gesture discrimination experiments were performed to verify the effectiveness of the J-PNN.

NM



Motion number	Motions	
	Left	Right
M 1	1	0
M 2	2	0
M 3	3	0
M 4	0	1
M 5	0	2
M 6	0	3
M 7	1	1
M 8	1	2
M 9	1	3
M10	2	1
M11	2	2
M12	2	3
M13	3	1
M14	3	2
M15	3	3

Fig. 4. Dual-arm motions discriminated in the experiments

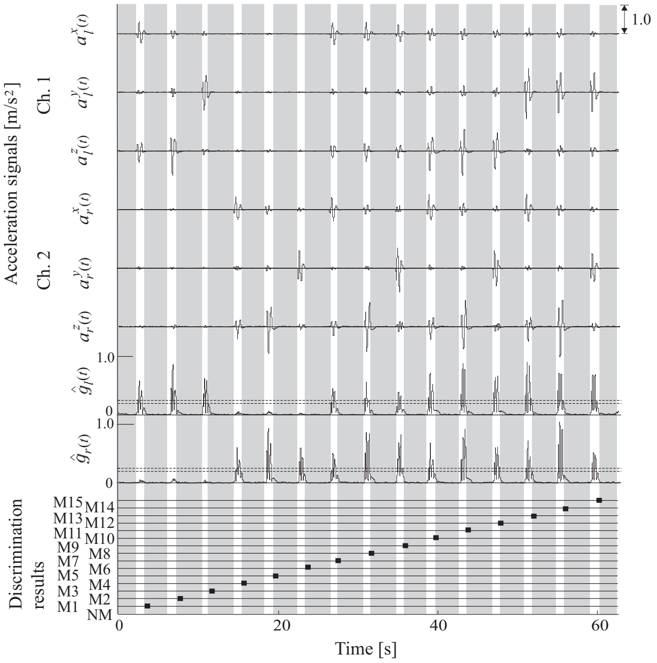


Fig. 5. Examples of measured ACC signals and discrimination results

Three-axis acceleration sensors (MA3-04Ac, Micro Stone Corporation) and wireless motion recorders (MVP-RF, Micro Stone Corporation) were utilized to measure acceleration signals. Each sensor was attached to both of the subjects' wrists. The sampling frequency was 200 [Hz], and the parameters for gesture discrimination were set as $f_c = 10.0$[Hz], $Th_{\mathrm{start}} = 0.2$, $Th_{\mathrm{end}} = 0.25$, $T_{\mathrm{end}} = 0.15$[s], $T = 0.4$[s], $S = 30$ and $H_{th} = 0.5$. The three healthy male subjects performed 15 dual-arm motions involving a combination of 3 single-arm motions (see Fig. 4) 50 times. In the experiments, 50 sets of each motion data were separated into 10 learning data sets and 40 discrimination data sets, and one of the 10 learning data sets was used for J-PNN training. It should be noted that the J-PNN can discriminate all 15 motions by learning just 3 motions for each arm (M1−M6).

A. Experimental results

Examples of discrimination results are given in Fig. 5, which shows acceleration signals $a_m^n(t)$, the square root of the sum of the acceleration signals $\hat{g}_m(t)$, and discrimination

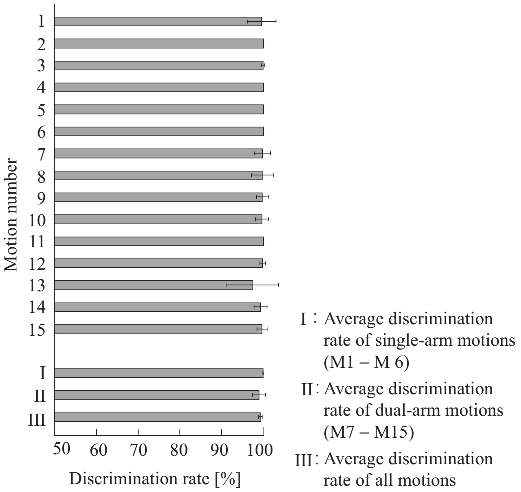


Fig. 6. Discrimination results of each motion (Subject A)

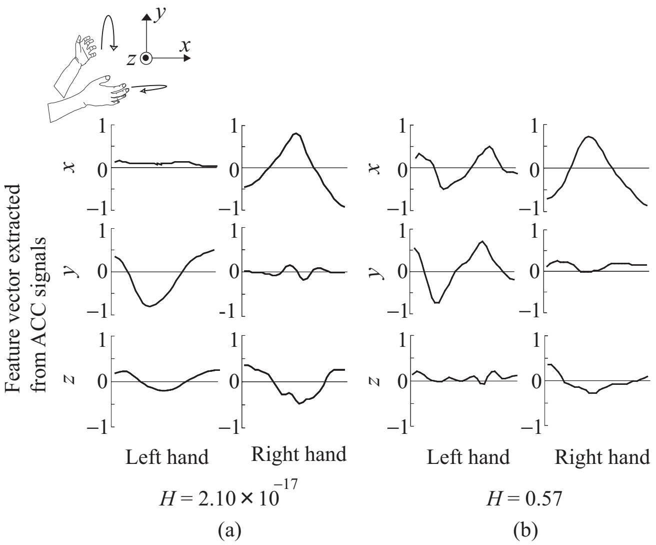


Fig. 7. Examples of ACC signals in a dual-arm motion (M13)

results. It can be seen that the features of acceleration signals are different individually according to each motion, and that they were discriminated accurately using the J-PNN. In addition, dual-arm motions (M7−M15) not learned by the J-PNN were also discriminated correctly, indicating that the network can be used to estimate the joint posterior probability of dual-arm motions with learning for single-arm motions only.

The accuracy of discrimination for each dual-arm motion is shown in Fig. 6, which plots the average discrimination rates for each motion while the set of learning data and discrimination data were changed and discriminated at each data set for 200 times. It indicates that the J-PNN can discriminate motions accurately (average discrimination rate for all motions: 99.34 ± 0.56 %). It is also confirmed that dual-arm motions (M7−M15) can be discriminated with high accuracy (98.96 ± 1.60 %) using the J-PNN.

Here, the motion with the lowest discrimination rate was M13 (i,e., moving the left arm in the $y-$axis direction and the right arm in the $x-$axis direction). Examples of feature vectors in motion M13 are shown in Fig. 7, which illustrates that waveforms were large according to the direction in which each arm was moved (left arm: $y-$axis; right arm: $x-$axis; see Fig. 7 (a)). It is also seen that the waveform of the $x-$axis is large for the left arm, even though the subjects moved their left arm in the $y-$axis direction (see Fig. 7 (b)). As dual-arm movement in different directions at the same time is difficult for humans [8], left arm motion may have been influenced by right arm motion because M13 was not a symmetric motion. In fact, the feature vector shown in Fig. 7 (b) was misdiscriminated as M7 using Eq. (17):

$$p'(c_1, c_2 | \boldsymbol{x}_1(t), \boldsymbol{x}_2(t)) = \prod_{\alpha \in \{1,2\}} p(c_\alpha | \boldsymbol{x}_\alpha(t)), \qquad (17)$$

where the modifying coefficient of posterior probability w_{α,c_α} was set as $w_{\alpha,c_\alpha} = 1$ in Eq. (10), which means the motions of each arm might be independent. In contrast, the J-PNN was able to discriminate it correctly with Eqs. (10) and (11), indicating its ability to discriminate based on motion dependency for asymmetric dual-arm motions.

TABLE I
RELATIONSHIPS BETWEEN THE NUMBER OF CORRECT AND INCORRECT CLASSIFICATIONS FOR THE DATA BELONGING TO THE SET OF DEPENDENT CLASSES $\boldsymbol{\Xi}$ WITH HIGH ENTROPY.

Subject	A	B	C
Number of misdiscrimination data belonging to Ξ with high entropy using Eq. (17) [trial]	113	200	227
Number of discriminated data for dependent data using Eq. (10) [trial]	70	149	136

Table I shows the number of correct classifications achieved using Eq. (10) and incorrect classifications using Eq. (17) for the data belonging to the set of dependent class $\boldsymbol{\Xi}$ with high entropy. The average number of discriminated data using Eqs. (10) and (11) was 127.6 ± 33.9 for the misdiscriminated 183.6 ± 43.6 data using Eq. (17). These results indicate that discrimination based on the dependency between the arms is effective for dual-arm motion discrimination.

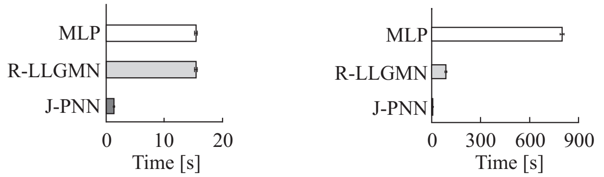
B. Comparison with previous methods

This section reports on discrimination performance verification for the J-PNN, the multi-layered persceptron (MLP) and the single R-LLGMN. The MLP had 12 layers (10 of them hidden), and the units of the input layer, each hidden layer and the output layer were 180, 200 and 15, respectively. Sigmoid functions were employed to express the input-output relationship of a unit. The back propagation algorithm was introduced, its learning rate was set as 0.01, and the error threshold was set as 0.01 for learning. The R-LLGMN consists of an input layer including 6 units and an output layer including 15 units. The other parameters were

TABLE II
AVERAGE DISCRIMINATION RATES OF MLP, R-LLGMN[9] AND J-PNN

Subject	A	B	C	Average
MLP	60.68±29.39	56.34±20.39	62.04±15.85	62.67±6.67
R-LLGMN	93.90±6.61	94.33±5.34	87.97±9.13	92.07±3.56
J-PNN	99.64±0.56	99.03±2.91	98.03±4.38	98.80±0.68

Ave. ± S.D. [%]



(a) Measurement time (b) Learning time

Fig. 8. ACC measurement time and learning time of each neural network for all motions

the same as those of the J-PNN. The specifications of the computer used for the experiment had an Intel Core (TM) i7 processor, a clock frequency of 3.3 [GHz] and 3.0 [GB] of memory.

The discrimination results for each subject are shown in Table II. It can be seen that the MLP discrimination rate was low (62.67 ± 6.67 %). Conversely, the R-LLGMN and the J-PNN achieved high levels of classification performance (92.07±3.56 % and 98.80±0.68 %, respectively) because the statistical distribution of time series data could be estimated based on the GMM and the HMM using the R-LLGMN and the J-PNN. The J-PNN can also discriminate 15 motions by learning only 3 motions for each arm.

Figure 8 shows the time taken for measuring learning data (referred to as *the measurement time*; Fig. 8 (a)) and the learning time (Fig. 8 (b)) for all subjects using each method. It can be seen that the measurement time and learning time using the J-PNN was much shorter than that of the other methods. The measurement time using the J-PNN was reduced because the J-PNN needs only single arm motions for learning. Moreover, the J-PNN can reduce the learning time because a TA is incorporated into the learning rule of the R-LLGMN and the J-PNN in order to regulate the learning convergence time, and the number of classes in the J-PNN learning became smaller by combination of the single-arm motions for dual-arm motion discrimination. The learning times for the MLP, the R-LLGMN and the J-PNN were 817.44 ± 7.37 [s], 104.13 ± 0.19 [s] and 5.12 ± 0.18 [s], respectively. The experimental results indicate that the J-PNN can drastically reduce learning time.

V. CONCLUSION

This paper proposes a joint probabilistic neural network (J-PNN) involving time series pattern discrimination and joint probability estimation, and outlines the development of a method for dual-arm motion discrimination. The method can discriminate dual-arm motions consisting of single-arm motions based on the dependencies between individual arm movements. The experimental results demonstrated that the J-PNN could be used to discriminate 15 dual-arm motions with $98.80 \pm 0.68\%$ accuracy based on learning of only three motions for each arm.

In future research, we plan to investigate the optimal entropy threshold H_{th} for individual subjects, and to discuss the applicability of the J-PNN to sign language recognition.

REFERENCES

[1] M. Knapp and J. Hall: Nonverbal Communication in Human Interaction. New York: Holt, Rinehart, and Winston (1972).

[2] M. Okamoto, K. shima, Y. Matsubara and T. Tsuji: Pattern Discrimination method with a boosting approach using hierarchical neural trees, Proceedings of the Institution of Mechanical Engineers, Part I: Joumal of Systems and Control Engineering, **222**-1-2, 701/710, (2008).

[3] R. Bowden and M. Sarhadi: A non-linear model of shape and motion for tracking finger spelt American sign language, Image and Vision Computing 20, 597/607 (2002).

[4] B. Gonzalo, R. Daniel, T. Gerhard and T. Gracian: Real time gesture recognition using Continuous Time Recurrent Neural Networks, ICST, 978-963-06-2193-9 (2007).

[5] H. Sawada and S. Hashimoto: Gesture Recognition Using an Acceleration Sensor and Its Application to Musical Performance Control, Electronics and Communications in Japan, Part 3, **80**-5, 9/17 (1997).

[6] C. Vogler and D. Metaxas: Adapting Hidden Markov Models for American Sign Language Recognition by using Three-dimentional Computer Vision Methods, IEEE International Conference on Systems, Man and Cybernetics, 156/161, Orlando, FL, Octorber, 12-15 (1997).

[7] J. Kim, W. Jang and Z. Bien: A Dynamic Gesture Recognition System for the Korean Sign Language (KSL), IEEE Trans. on Systems, Man and Cybernetics-Part B, Cybernetics, **26**-2, 354/359 (1996).

[8] Y. Guiard: Asymmetric Division of Laber in Human Skilled Bimanual Action: The Kinematic Chain as a Model, Slightly edited version of an article originally published in *Journal of Motor Behavior*, **19**, 486-517 (1987).

[9] T. Tsuji, N. Bu, O. Fukuda and M. Kaneko: A Recurrent Log-Linearized Gaussian Mixture Network, IEEE Trans. on Neural Networks, **14**-2, 304/316 (2003).

[10] P. J. Werbos: Backpropagation through time:What it does and how to do it, Proc. IEEE, **78**-10, 1550/1560 (1990).

[11] M. Zak: Terminal Attractors for Addressable Memory in Neural Networks, Physics Letters A, **133**-1-2, 18/22 (1988).

[12] K. Shima, M. Okamoto, N. Bu, and T. Tsuji: Novel Human Interface for Game Control Using Voluntarily Generated Biological Signals, Journal of Robotics and Mechatronics, **18**-5, 626/633 (2006).