Development of a Sensory Data Glove Using Neural-Network-Based Calibration

Chin-Shyurng Fahn and Herman Sun

Department of Electrical Engineering
National Taiwan University of Science and Technology
Taipei, Taiwan 106, Republic of China
csfahn@mouse.ee.ntust.edu.tw

Abstract

In this paper, we present the development of a sensory data glove using infrared receivers/transmitters as finger-bend measurement sensors. This data glove produces nonlinear outputs that must be calibrated before it is employed in a virtual environment. To make the glove easy for use, a four-stage calibration procedure together with the construction of the calibration device is realized.

In the software calibration process, we devise a neural-network-based function approximator trained with a modified robust backpropagation (BP) algorithm that has the ability of eliminating the effect of noises in the training data. In order to speed up the training process, we propose a "tentative-and-refined" training method that is combined with a robust BP algorithm to constitute the modified one. Many successful experiments are made on a concrete data glove to verify the effectiveness of the proposed algorithms. So far, the experimental results of the calibration process with our method are very satisfactory.

Key words: data glove, calibration device, neural-network-based function approximator, robust BP algorithm, tentative-and-refined training.

1. Introduction

In recent years, a new type of input devices, a sensory data glove, has been extensively applied along with the popularization of virtual reality (VR). The data glove is a multi-sensory device that generates a large amount of data and is more complex than other input devices. Nevertheless, most researchers still adopt this device because the natural interfacing characteristic of the data glove with the human being is the way to improve system manipulations that are applicable in many specific fields, particularly in immersive VR systems. At present, the data glove has been increasingly employed in the areas of teleoperations and robotic control [1]-[3], surgery training of medical applications [4],[5], entertainment sports of VR systems [6],[7], industrial manufacturing of CAD/CAM applications [8],[9], and so on.

Among the available input devices for VR, hand-tracking technology is the most popular one. Such glove-based input devices let VR users apply their manual dexterity to the VR activities. Hand-tracking gloves currently marketed include: Sayre Glove, MIT LED Glove, Digital Data-Entry Glove, DataGlove, Dexterous HandMaster, Power Glove, CyberGlove, VPL Glove, and Space Glove [10].

According to the outputs of sensors, the data gloves can be grouped into two classes: one produces *linear output*, and another produces *nonlinear output*. Either linear or nonlinear data gloves should be calibrated before they can be used in the applications. The calibration process of linear data gloves is directly executed by a linear mapping, but that of nonlinear data gloves is not so easy owing to lack of outputs' references of nonlinear sensors. In this paper, we present the development of a sensory data glove using infrared receivers/transmitters as the finger-bend measurement sensors. This data glove produces nonlinear outputs that must be calibrated before operation. To make the glove easy for use, the construction of a calibration device together with a four-stage calibration procedure is developed. The former creates a calibration device for a nonlinear data glove, and the latter performs an associated nonlinear mapping via a neural-network-based function approximator [11]-[13] that is trained by a modified robust backpropagation (BP) algorithm of noises elimination capabilities.

The rest of the paper is organized as follows. In Section 2 we describe the hardware construction of the data glove as well as the calibration device. In Section 3 we introduce the software calibration process. In Section 4 we present experimental results. Finally, we summarize our findings and conclude our paper in Section 5.

2. Hardware Construction

2.1 Finger-bend sensors

The finger-bend sensor is made of infrared transmitter and receiver components that are plugged into a small flexible pipe as shown in Fig.1. The flexible pipe functions as the infrared signal transmission space. When an operator's finger is bent, the finger-bend sensor located on the relative joint is also bent in the same shape that causes the decreasing of the radiation signal reaching the infrared receiver. This signal decrease will affect the output impedance of the receiver. Unfortunately, our experimental results of the relationship between the bend angle and the output impedance are nonlinear. Such nonlinear characteristic is affected by the bend position of the sensor. To overcome this problem, we implement a calibration device associated with a four-stage calibration procedure.

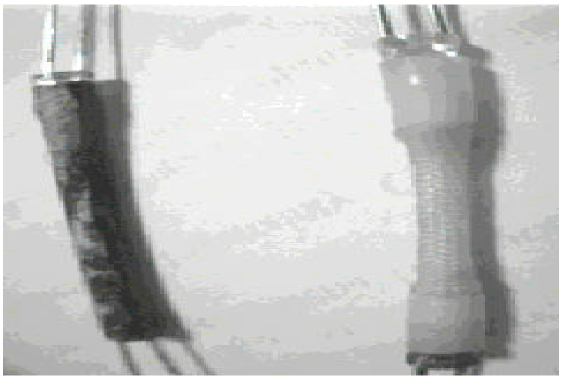


Fig. 1 The finger-bend sensors with two flexible pipes of different materials.

2.2 Fitting up the data glove

The data glove we create consists of twelve bend sensors, ten of which are located in the finger joint positions of the glove, one of the remainder is in the thumb-index abduction angle position, and the last one is in the carpal position for measuring the wrist pitch rotation angle. Figure 2 illustrates the position of each sensor equipped on the data glove.

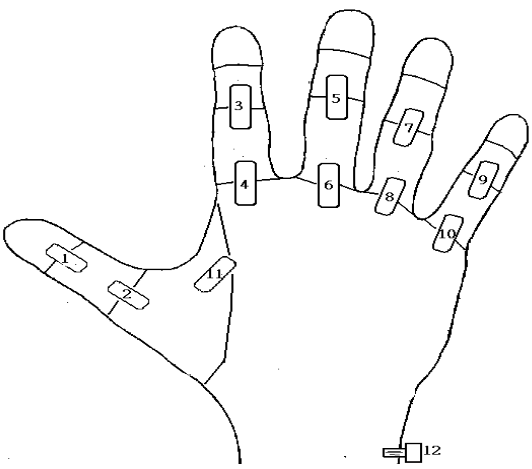


Fig. 2 Positions of the sensors on the data glove.

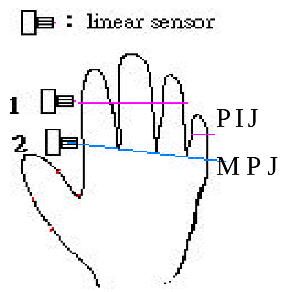
As shown in Fig.2, the sensor located in the carpal position is a linear one that produces linear outputs. In this case, the calibration is simply a linear mapping process. The name of each sensor related to its position in the glove is depicted in Table 1.

Table 1 The Names of the Sensors Related to Fig.2

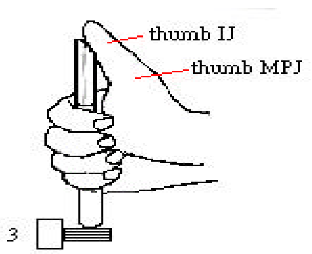
Position no.	Sensor name
1	Thumb IJ
2	Thumb MPJ
3	Index PIJ
4	Index MPJ
5	Middle PIJ
6	Middle MPJ
7	Ring PIJ
8	Ring MPJ
9	Pinkie PIJ
10	Pinkie MPJ
11	Thumb-index abduction
12	Wrist pitch

2.3 The calibration device

The calibration device of the data glove is composed of three linear sensors. The first linear sensor is fitted on the positions of proximal interphalangeal joints (PIJ), which provides the referenced values for the four PIJ sensors of the data glove. The second sensor is fitted on the positions of metacarpo-phalangeal joints (MPJ) to provide the referenced values for the four MPJ sensors of the data glove. The last sensor is attached to a moveable stick inside a pen-shaped tube to convert the bend angles of thumb IJ and MPJ joints into a linear motion. Figure 3 shows the positions of the linear sensors used for data calibration.



(a)



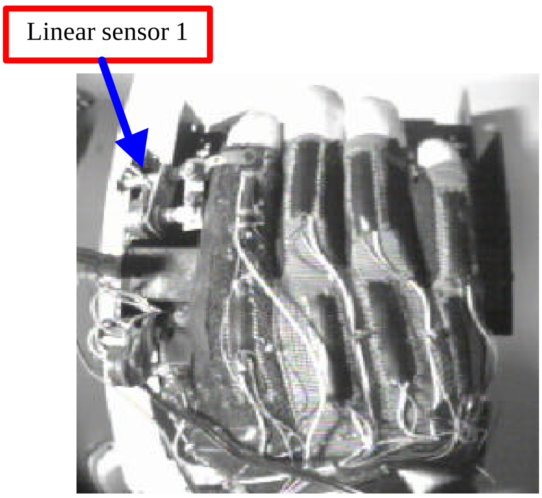
(b)

Fig. 3 Linear sensors of the calibration device positioned in: (a) four-finger PIJ and MPJ joints; (b) thumb IJ and MPJ joints.

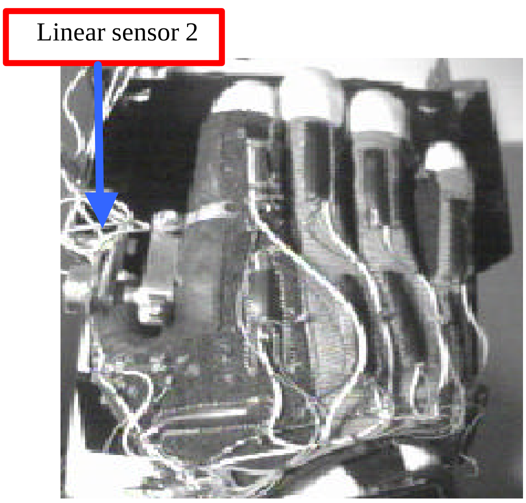
Fig. 4 Illustration of the data glove calibration process: (a) four-finger PIJ joints calibration; (b) four finger MPJ joints calibration.

The calibration process is executed before the data glove is employed in the virtual environment. To make it easy for use, we develop the calibration technique of four stages as follows:

1) Use the first linear sensor to calibrate the four-finger PIJ joints. This stage begins with placing the hand on the calibration device whose first sensor attaches to the middle phalange position of the index as shown in Fig.4(a). As the calibration process is started, users bend the four-finger PIJ joints to the maximum angle and then stretch the PIJ joints back to their original positions at a constant velocity.
2) Use the second linear sensor to calibrate the four-finger MPJ joints. At the beginning of this stage, the hand wearing the data glove is placed on the calibration device with the second sensor attaching to the proximal phalange position of the index as shown in Fig.4(b). When the calibration process is started, users flex the four-finger MPJ joints to the maximum angle and restore the MPJ joints to the original positions at a constant velocity.

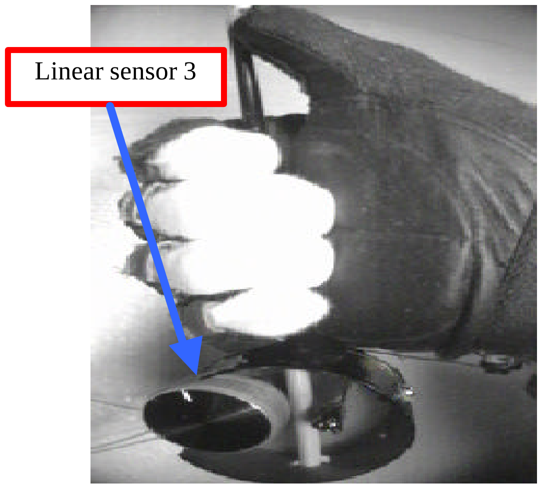


(a)

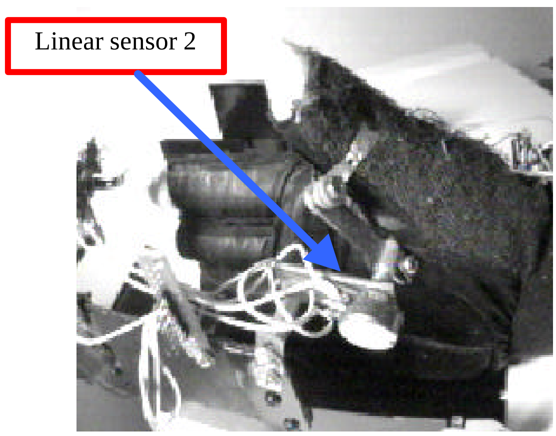


(b)

3) Use the third linear sensor to calibrate the thumb IJ and MPJ joints. At this stage, users wearing the data glove grasp the pen-shaped tube and push the movable stick downwards as shown in Fig.5(a). The motion of the stick is connected with the linear sensor to produce the referenced outputs for the thumb IJ and MPJ sensors.
4) Use the second linear sensor to calibrate the thumb-index abduction angle. At this stage, the hand is placed on the calibration device with the palm facing to the left as shown in Fig.5(b). The second linear sensor is attached to the thumb distal phalange position for measuring the movement of the thumb-index abduction angle.



(a)



(b)

Fig. 5 Illustration of the data glove calibration process: (a) thumb IJ and MPJ joints calibration; (b) the thumb-index abduction angle calibration.

3. Software Calibration Process

After the four-stage calibration procedure is finished, a function approximator implemented by a feedforward neural network is developed to each sensor of the data glove. The structure of the neural network is designed in the following way to provide the function approximating capability. The hidden layers of the network contain up to twenty-five nodes. It was determined experimentally for obtaining the best approximation result. In our experiments, each network consists of five layers.

The neural-network-based function approximator in the calibration process is normally trained by the BP algorithm, which acts as a nonlinear converter to map data glove sensors' outputs into calibration sensors' values. The outputs of these nonlinear converters are then transformed into the finger-bend angles by a linear mapping function.

Some factors that slow down the execution time of the BP algorithm, especially when using a large amount of training pairs, are summarized as follows:

1) *The correlation between training pairs*. It means that on an average, the sampling signals do not change rapidly so that the difference between adjacent samples should have a lower variance than the variance of the whole signals. When applying the BP algorithm to train the network, we treat each training pair as an independent one that will generate conflict in the weight adjustment of the training process.
2) *The number of floating-point multiplications*. Assume that the number of floating-point multiplications needed to train one training pair is n, the total number of floating-point multiplications required for one iteration in the training process yields nm or more when the conflict is occurred for m training pairs in the training set.
3) *Small learning rates*. When a large amount of training pairs is adopted in the training process, a small learning rate is usually selected to prevent the conflict in the weight adjustment among training pairs.
4) *Undesired initial weights of the network*. The initial weights selected at random normally generate the outputs that are deviated from the approximated function.

To speed up the BP algorithm, we propose a "tentative-and-refined" training method. This method includes a tentative training procedure, followed by a refined training one. In the tentative training procedure, part of the original training set is chosen to train the network. After this tentative training, the entire original training set is employed to refinedly train the network. The motivation is based on the fact that the training speed will perform rapidly when the training set is not too large. Additionally, it can provide good initial weights for the subsequent process.

When noises exist, the approximated function behaves like a highly nonlinear one. Consequently, the number of neurons in the network should be large enough to approximate the nonlinear function. Furthermore, as the nonlinearity increases, more number of iterations is needed for the network to reach the desired error that causes the performance of the BP algorithm becoming too slow for practical uses. In most applications, it is difficult to guarantee that noises do not present in the training set. In order to eliminate the effect of noises in the training data, we devise a neural-network-based function approximator trained by a modified robust BP algorithm. This training approach combines the "tentative-and-refined" training method and a robust BP algorithm [14]. The following describes this combination that results in a modified robust BP algorithm [15]:

Step 1: Use the first procedure of the tentative-and-refined training method to train the network until the value of its energy function $E_R ? \overset{K}{\underset{k?1}{?}} ?_t(r_k)$ reaches $?$, where $?_t ?r_k ?$ is the integration of the Hampel's tanh estimator, r_k is the error residual, K is the number of training sets, and $?$ is the threshold employed to detect the time when the energy function has a sharp drop during the initial estimation.

Step 2: Reset a counter k that is used for updating $?_t ?r_k ?$.

Step 3: Compute the robust energy function: if $E_R ? ?$ or the energy difference between the current and the previous iterations is less than $?_d$, then terminate the learning process.

Step 4: If the counter k is a multiple of the time duration $?t$ between successive updates, then alter $a?t?$ and $b?t?$ which are the time-various cut off points used for obtaining the derivative of the optimal $?_t ?r_k ?$.

Step 5: Compute the error signals for the output layer and hidden layers by using the robust BP algorithm, and update the weights of the network.

Step 6: Increase the counter k by one and go to Step 3.

4. Experimental Results

To demonstrate the performance of our training method, we construct a feedforward neural network consisting of 4 layers with 2 input neurons, 1 output neuron, and 8 neurons in the first and the second hidden layers. The learning rate is 0.002 , the parameter $?$ of the activation function is 15, and the expected error is 0.000005. Firstly, the network is trained with a traditional BP algorithm. The number of iterations and the execution time required in each training process are recorded, and then compared to the tentative-and-refined training method with the learning rate of 0.005 and the expected error of 0.0005 in the tentative training procedure. In this initialization process, the training pairs are selected from the original ones with the interval of 20 samples, including stationary points. The number of iterations and the execution time required for the above two techniques are listed in Table 2 and Table 3 with respect to 14 different experiments.

Table 3 shows the total execution time of the tentative-and-refined training method is less than that of a traditional BP algorithm, even though the number of iterations of the weight initialization procedure is larger than that of the traditional one, because of fewer training pairs participating in the tentative training stage. Figure 6 shows the output of the ring MPJ sensor of the data glove, and the output of the network trained by the modified robust BP algorithm is shown in Fig.7.

Table 2 The Number of Iterations and the Execution Time of the Traditional BP Algorithm

Exp. no.	The traditional BP algorithm	
	Iterations	Time in sec.
1	1,347	146
2	573	63
3	1,316	144
4	1,772	194
5	751	83
6	3,472	376
7	1,654	181
8	1,771	193
9	4,500	491
10	600	67
11	512	56
12	2,036	221
13	4,500	489
14	400	45
Total execution time in sec.		2,798

Table 3 The Number of Iterations and the Execution Time of the Tentative-and-Refined Training Method

Exp. No.	The tentative-and-refined training method				
	Procedure 1		Procedure 2		Total time in sec.
	Iterations	Time	Iterations	Time	
1	116	0.7	73	8	8.7
2	3,784	24	5	0.5	24.5
3	23,789	144	6	0.7	144.7
4	1,480	9	391	42	51
5	4,806	30	1	0.1	30.1
6	1,881	12	822	90	102
7	5,397	33	1	0.1	33.1
8	7,202	44	6	0.7	44.7
9	3,092	19	7	0.8	19.8
10	418	3	1	0.1	3.1
11	6,908	43	7	0.8	43.8
12	9,010	55	1	0.1	55.1
13	6,446	39	885	95	134
14	578	3	273	30	33
Total execution time in sec.					822.6

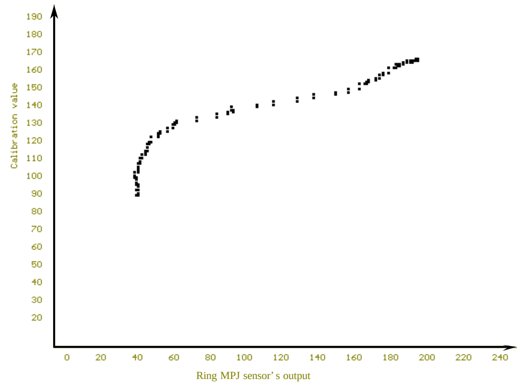


Fig. 6 The output of the ring MPJ sensor on the data glove.

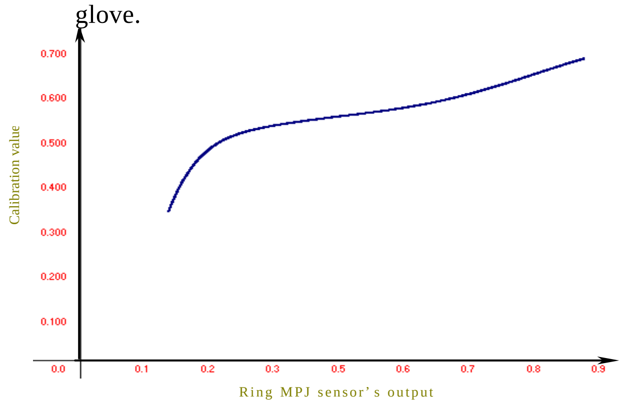
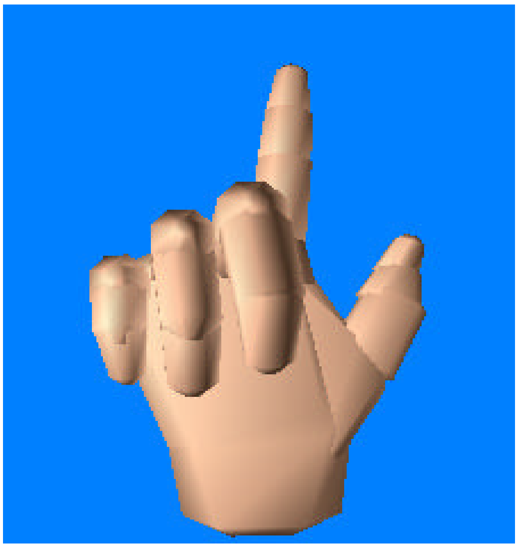


Fig. 7 The output of the network trained by the modified robust BP algorithm.

The performance of the data glove after completing the calibration is illustrated in Fig. 8.

(a)



(b)

Fig. 8 A hand gesture and the corresponding virtual hand: (a) a user's hand wearing the data glove; (b) the virtual hand in a virtual environment.

5. Conclusions

In this paper, we have presented the construction of a nonlinear data glove associated with a four-stage calibration procedure. In the software calibration process, we propose a new method of accelerating the BP algorithm by repeatedly training the network with different sizes of training sets that are produced by resampling the ones. We call it the tentative-and-refined training method. It can work well in the application of function approximation because the training pair generated by the sampling mechanism is usually correlated to the adjacent one. To increase the robustness of the algorithm, we devise a modified robust BP algorithm that combines the "tentative-and-refined" training method and the robust BP algorithm.

Although the data glove provides a natural way of performing a human-machine interface, it is not so convenient for the operator to use in the virtual environment owing to the presence of electric wires connecting the glove with the control device. More researches that should be accomplished in future involve:

1) *The development of a force-feedback device*. This device is attached to the data glove to feed the force back to the operator from a virtual environment. When a virtual hand touches a virtual object in the virtual environment, the force generated from the object is calculated according to the physical modeling used, and then sent out to the force-feedback device.
2) *The natural way of object grasping in a virtual environment*. In the real-time virtual reality application, the user wearing a data glove manipulates virtual objects via the virtual hand. To provide more realistic object grasping, the force generated from the hand making contact with the object should be modeled in the virtual environment.
3) *The development of a motion constraint device*. This device is employed to restrict the fingers' movements of an operator's hand when he grasps an object in the virtual environment.
4) *The development of a portable data glove*. In this research, we attempt to increase the efficiency of the data glove acting as a human-machine interface, and to enhance its performance.

References

[1] C. P. Tung and A. C. Kak, "Automatic learning of assembly tasks using a DataGlove system," in *Proc. of IEEE/RSJ Int. Conf. on Intell. Robots and Syst., 1995*, pp.1-8.

[2] H. Hashimoto et al., "An unilateral master-slave hand system with a force-controlled slave hand," in *Proc. of IEEE Int. Conf. on Robotics and Automat., 1995*, pp.956-961.

[3] B. K. Sing and K. Ikeuchi, "A robot system that observes and replicates grasping tasks," in *Proc. of the 5th Int. Conf. on Comput. Vision, 1995*, pp.1093-1099.

[4] W. J. Greenleaf, "Rehabilitation, ergonomics, and disability solutions using virtual reality technology," in *Proc. of Interactive Tech. and the New Paradigm for Healthcare Conf., 1995*, pp.415-422.

[5] R. M. Satava, "Medicine 2001: the king is dead," in *Proc. of Interactive Tech. and the New Paradigm for Healthcare Conf., 1995*, pp.334-339.

[6] S. Kawamura et al., "Development of a virtual sports machine using a wire drive system- a trial of virtual tennis," in *Proc. of IEEE/RSJ Int. Conf. on Intell. Robots and Syst., 1995*, pp.111-116.

[7] D. Gromala and Y. Sharir, "Dancing with the virtual dervish: virtual bodies," in *Proc. of Virtual Reality Software and Tech. Conf., 1994*, pp.321-328.

[8] M. W. S. Jaques, P. Strickland, and T. J. Oliver, "Design by manufacturing simulation: concurrent engineering meets virtual reality," in *Proc. of Mechatronics Conf., 1994*, pp.637-642.

[9] M. W. S. Jaques and D. J. Harrison, "Using gestures to interface with a 'virtual manufacturing' package," in *Proc. of the 3rd Int. Conf. on Interface to Real and Virtual Worlds, 1994*, pp.231-240.

[10] D. J. Sturman and D. Zeltzer, "A survey of glove-based input," *IEEE Comput. Graphics and Appli*., vol.14, no.1, pp.30-39, 1994.

[11] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Net*., vol.2, pp.359-366, 1989.

[12] G. Cybenko. "Approximation by superposition of a sigmoidal function," *Math. of Contr., Signals, and Syst*., vol.2, pp.303-314, 1989.

[13] K. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Net*., vol.2, pp.183-192, 1989.

[14] D. S. Chen and R. C. Jain, "A robust back propagation learning algorithm for function approximation," *IEEE Trans. Neural Net*., vol.5, no.3, pp.467-479, 1994.

[15] S. C. Sun, *Development of a Sensory Data Glove with Neural-Network-Based Calibration*, Master Thesis, Dep. of Elect. Eng., Nat. Taiwan Univ. of Sci. and Tech., Taipei, Taiwan, 1998.