JOURNAL OF AIRCRAFT
Vol. 43, No. 3, May–June 2006

Analysis of Human-Pilot Control Inputs Using Neural Network

Shinji Suzuki* and Yuki Sakamoto†
University of Tokyo, Tokyo 113-8656, Japan
Youhei Sanematsu‡
Mitsubishi Electric Corporation, Kamakura 247-8520, Japan
and
Hiroshi Takahara§
All Nippon Airways, Tokyo 144-0041, Japan

A neural-network-modeling approach is applied to analyze human-pilot control inputs during the landing phase in the visual approach. Flight data that contain the aircraft state variables and pilot control inputs are recorded using a flight simulator. The time history of visual cues and control inputs is utilized as teaching data for neural networks that can emulate the movements of a human pilot. A genetic algorithms approach is proposed to improve the generalization ability of the network by determining the network structure and initial values of its parameters. Generalization capabilities are evaluated by analyzing the flight data of a personal-computer-based simulator. The contribution ratios of each visual cue and their sensitivities to the control inputs of a pilot are estimated by analyzing the obtained neural-network models using a training simulator. The obtained results reveal that the proposed method can be used for analyzing the skill of a pilot.

I. Introduction

A LANDING approach is perhaps the most difficult maneuver for airline pilots. With the exception of fully automated landing systems, such as CAT III C, pilots have to estimate airspeed, descent rate, altitude, and pitch angle of a plane by using visual cues from the cockpit during the final landing phase, because they do not have sufficient time to read them on the instruments. It is considered that the estimation skill of a pilot has an important effect on the smoothness of the landing. Because it is extremely difficult to analyze the cognitive process in a human brain, there is a strong demand to develop tools that can analyze the control inputs of a pilot and can be utilized for training pilots.

It has been widely reported that visual cues play a major role when pilots capture the states of an airplane during the landing phase. In the 1940s, Gibson investigated the landing skill of pilots from a psychological viewpoint and proposed that the optical flow of visual cues was utilized to perceive the states of the landing airplanes.[1] Recently, this hypothesis has been confirmed by the mathematical modeling of the human brain and nervous system[2]; that is, the states of a moving body could be estimated from the optical information by using simple neurological processes. Additionally, the optimal state estimation using the Kalman filter technique reveals that visual cues are more important than motion cues to minimize the effect of observation errors.[3,4]

In this paper we propose the application of a neural-network (NN) model for analyzing information process flows from visual cues to the control inputs of a pilot. Artificial NNs are mathematical models that emulate biological nervous systems and are composed of a large number of highly interconnected processing elements, similar to neurons.[5,6] The elements are tied together with weighted connections that are analogous to synapses. The synaptic connections are adjusted by the learning process in biological systems. In artificial NNs, the teaching or training data set of input/output data is used for adjusting several parameters. After the training, the NNs can present the input/output relationship. The artificial NN has been applied to automatic recognition systems and automatic control systems of complicated problems that have high nonlinearity.

Attempts have been made to analyze the control inputs of humans in automobile driving[7] and throttle operations of landing airplanes.[8] We have applied the NN model to analyze the movements of an airplane pilot during the landing phase.[9,10] In our analysis, flight data obtained by using a flight simulator were used for the NN learning. Input data for NNs are visual cues (e.g., runway geometries and the horizon) and control-stick input. The output data from NN are control-stick and throttle-lever deflections. The sensitivity from the input to output data as well as the contribution ratio of each input are computed to analyze the control inputs of a pilot.

It has been recognized that the most difficult problem in creating NN models is deciding whether the obtained model has generalization capability. Generalization refers to the NN producing reasonable outputs for inputs not encountered during training. A careful choice of network structures is necessary in determining redundant hidden nodes to improve the generalization capability of NNs.[11] There have been several approaches that modify training algorithms, such as minimizing the deviation from the desired output for several input data sets[12] and minimizing the difference between the normal and faulty networks.[13]

In this paper, genetic algorithms (GAs) are applied to obtain NN models with generalization capabilities. GA is one of the optimization methods and uses search procedures based on the mechanics of natural genetics, which combines a Darwinian survival-of-the-fittest strategy to eliminate unfit characteristics and random information exchange.[6] GAs can efficiently deal with large and complex problems with integers and real variables to find nearly global optima. In our study, GA is applied to optimize the number of nodes in hidden layers and initial values for weights and biases of the network. The real values of weights and biases are computed by a normal learning process such as back propagation. The objective function in GA is selected as the weighted sum of the training errors for the teaching data set and for a test data set to improve the generalization capability of the network. It should be noted that this method automatically

Presented as Paper 2005-910 at the AIAA 43rd Aerospace Science Meeting, Reno, NV, 10–13 January 2005; received 30 March 2005; revision received 24 July 2005; accepted for publication 25 July 2005.

*Professor, Department of Aeronautics and Astronautics, 3-7-1 Hongo, Bunkyo-ku; tshinji@mail.ecc.u-tokyo.ac.jp. Senior Member AIAA.

†Graduate Student, Department of Aeronautics and Astronautics, 3-7-1 Hongo, Bunkyo-ku.

‡Researcher, Aerodynamics, Guidance and Control Section, Kamakura Works, Kamakura, Kanagawa.

§767 Captain, Deputy Director, Flight Operation Administration, 3-3-2 Haneda Airport.

determines the number of hidden nodes and initial parameters of the network and does not require special training algorithms of the network.

The remainder of the paper is organized as follows. In Sec. II, NN modeling with GA is proposed as a method to improve the generalization ability of the network. The proposed method is applied to analyze the longitudinal control process of a pilot during the final approach and the flare maneuver. In Sec. III, experiments on the analysis of human-pilot control inputs for a personal computer (PC)-based flight simulator of a large airliner whose mathematical model is known are presented. In Sec. IV, the analysis data of several pilot control processes obtained by using a flight simulator for airline pilot training are presented. Finally, in Sec. V, the main contributions of this paper are summarized and several issues for future study are indicated.

II. Neural-Network Modeling with Genetic Algorithms

An artificial NN is applied to the modeling of the human control process. An artificial neuron can be referred to as a processing element, as shown in Fig. 1. Mathematically, the operation of the neuron can be represented as

$$y = f\left[\sum_{i=1}^{n}\left(w_i x_i + \frac{b}{n}\right)\right] \tag{1}$$

where x_i and w_i are the input and weight, respectively, and b is the threshold or bias of the activation function $f : x \mapsto 1/(1+e^{-x})$. A feedforward multiplayer network with three layers is used in this study, as shown in Fig. 2, where the system has n inputs and l outputs. Note that the hidden layers have the neuron model, as shown in Eq. (1), and the input and the output layers are using a linear function as the activation function. Therefore, the output of the network is generated as follows:

$$y_j^h = f\left[\sum_{i=1}^{n}\left(w_{j,i}^h x_i + \frac{b_j^h}{n}\right)\right] \tag{2}$$

$$y_k^o = \sum_{j=1}^{m}\left(w_{k,j}^o y_j^h + \frac{b_k^o}{m}\right) \tag{3}$$

where the superindice h and o represent hidden and output layers, respectively. In our study, the contribution ratio and the sensitivity from input $x_i (i = 1, \ldots, n)$ to output $y_k (k = 1, \ldots, l)$ are calculated in the following manner to analyze information flows in the network:

$$\begin{aligned} S_{k,i} &= \frac{\partial y_k^o}{\partial x_i} = \frac{\partial}{\partial x_i}\sum_{j=1}^{m}\left(w_{k,j}^o y_j^h + \frac{b_k^o}{m}\right) \\ &= \sum_{j=1}^{m} w_{k,j}^o \frac{\partial y_j^h}{\partial x_i} = \sum_{j=1}^{m} w_{k,j}^o w_{j,i}^h \dot{f} \end{aligned} \tag{4}$$

$$C_{k,j} = \sum_{j=1}^{m}\left|w_{j,k}^h x_i + \frac{b_j^h}{n}\right| \cdot \left|w_{k,j}^o y_j^h + \frac{b_k^o}{m}\right| \tag{5}$$

The learning process of the network is carried out to minimize the mean-square error between the teaching or training data set $\boldsymbol{t}$ and the computed output data set $\boldsymbol{a}$ for the same input data in the teaching data as follows:

$$F_{NN}(\boldsymbol{w}, \boldsymbol{b}) = E[(\boldsymbol{t} - \boldsymbol{a})^T(\boldsymbol{t} - \boldsymbol{a})] \tag{6}$$

The weights $\boldsymbol{w}$ and biases $\boldsymbol{b}$ of neurons are obtained by the learning process using the error back-propagation method. Although there are many learning algorithms, we are using the Levenberg–Marquardt method described by Ham and Kostanic,[14] which combines the steepest descent method, and the Gauss–Newton method.

The most difficult problem in preparing NN models is the determination of their structures. Because a three-layer network was used in this study, it is essential to specify the number of nodes in hidden layers. An inappropriate choice for this number leads to a lack in the generalization ability of the network, which means that input data that are slightly different from the teaching data produce a poor estimate of the actual output. Additionally, an inappropriate choice of initial values of parameters may lead to local minimum solutions in the learning process of the NN. This paper proposes the use of GAs to determine the NN structure and the initial values of the parameters for the learning process of each NN.

GA is one of the optimization methods that use search procedures based on the mechanics of natural genetics and combines a Darwinian survival-of-the-fittest strategy to eliminate unfit characteristics and random information exchange.[6] First, the number of nodes in the hidden layer as well as the initial values of the weights and biases of each neuron is coded into a binary string. The initial population is randomly selected, and the population is successively transformed by the use of probabilistic rules from generation to generation. Second, the NNs are generated for each string, and the mean-square error defined in Eq. (6) is computed. It should be noted that the weights and the biases of each NN are optimized by the usual error back-propagation process using the Levenberg–Marquardt method, because the back-propagation method is more efficient than the GA optimization process. The generalization capability of the obtained NN is examined for the test data set $\boldsymbol{t}^*$, which is slightly different from the teaching data. The output data set $\boldsymbol{a}^*$ is computed when the test data set $\boldsymbol{t}^*$ is applied as input data to each NN obtained after the learning process for the data set $\boldsymbol{t}$. Furthermore, the mean-square error between the test data set $\boldsymbol{t}^*$ and the output data set $\boldsymbol{a}^*$ is computed. The cost function to be minimized in the GA is the weighted sum of the mean-square error for the teaching data set and the error for the test data set, as expressed in Eq. (7):

$$F_{\mathrm{GA}}(\boldsymbol{w}_{\mathrm{ini}}\boldsymbol{b}_{\mathrm{ini}}n_{\mathrm{nid}}) = E[(\boldsymbol{t}-\boldsymbol{a})^T(\boldsymbol{t}-\boldsymbol{a})] + \alpha E[(\boldsymbol{t}^*-\boldsymbol{a}^*)^T(\boldsymbol{t}^*-\boldsymbol{a}^*)] \tag{7}$$

where $\boldsymbol{w}_{\mathrm{ini}}$, $\boldsymbol{b}_{\mathrm{ini}}$, and n_{ini} are the initial values of weights, biases, and the number of nodes in the hidden layer to be optimized in the GA, respectively. Note that α in Eq. (7) is a scaling factor.

Third, a reproduction process is carried out to select good strings with low cost function by using the survival-of-the-fittest concept. Although several methods have been proposed for this process, we used a roulette method, which produces parent strings. Crossover and mutation following reproduction are introduced to impart probabilistic search characteristics to the GA. As a crossover process, a uniform crossover is selected, in which paired strings merge at random. Mutation is a simple alternation of a bit in a string based on the probability of mutation. When mutation is used sparingly with reproduction and crossover, it helps avoid a local minimum through search iterations. Note that the best string is transferred to the next generation; this is referred to as the elite strategy. This process is repeated until the convergence is obtained, as shown in Fig. 3.

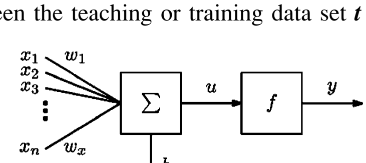


Fig. 1 Neuron model.

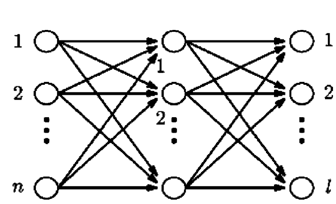


Fig. 2 Neural network.

III. Experimental Results (PC-Based Simulator)

The proposed method is applied to analyze the longitudinal control process of a pilot during the final approach and flare maneuver. PC-based simulator experiments were carried out prior to application in experiments using a flight simulator for airline pilot training. Because the dynamic equations of an airplane are perfectly known

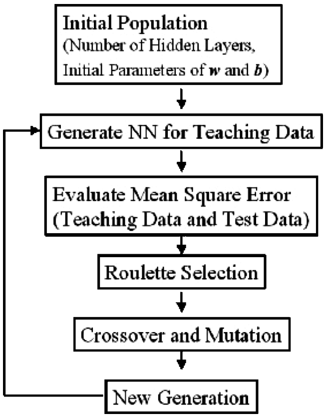


Fig. 3 Flow chart of GA.

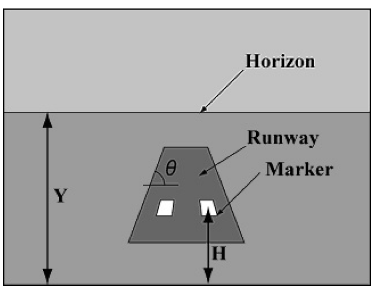


Fig. 4 Visual cues.

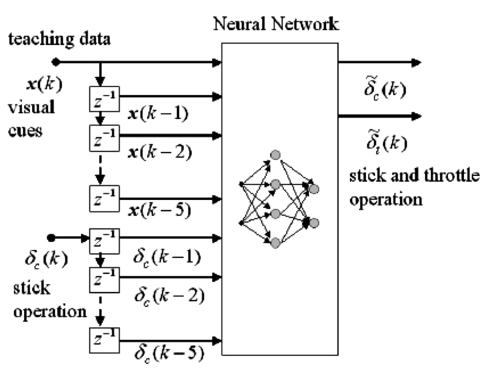


Fig. 5 Neural network model of pilot movements.

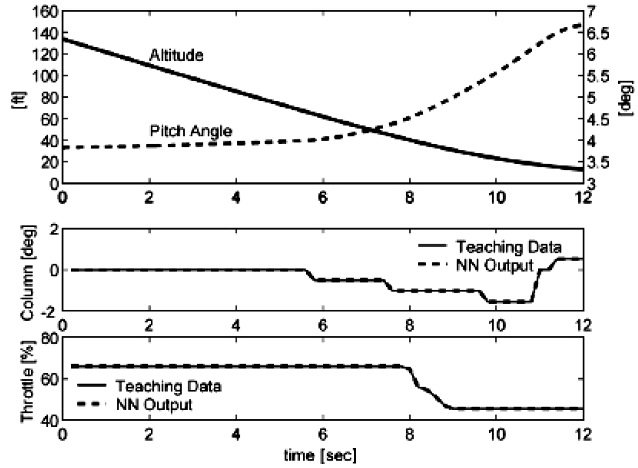


Fig. 6 Teaching data and output of neural networks.

in the PC-based simulator, it is possible to evaluate the characteristics of the obtained NN. An airplane model was constructed from the published B747 data sheet.[15]

In the first step, one of the authors operated the flight simulator using a control stick. The operator observed the screen that showed the visual image of the runway during the final landing phase. The flight conditions are weight (lb): 564,000; CG position: 24% MAC; flap position: 30 deg; initial altitude: 200 ft; initial path angle: −3 deg; and initial velocity, ft/s: 239.4. The operator was using the visual cues on the screen and could obtain information of the stick position during the experiment. The visual cues are represented as shown in Fig. 4, where Y and H are the height of the horizon and the marker on the runway from the bottom of the screen, respectively, and θ is the angle of the runway sideline. Figure 5 illustrates the input and output data of the network. Note that not only the present data but also the time-lagged data were used as the input data of the network, because the memory of visual cues and stick position influence present pilot-control inputs. The sample rate of data acquisition was 10 Hz, and the time-lagged data was selected as follows: current time, −0.4 s, −0.8 s, −1.2 s, −1.6 s, and −2.0 s. This is because that both the present and past information are used to generate the pilot-control movements: control stick and throttle deflections. The flight data of 12 s before touchdown was used to teach the NN that minimizes the mean-square error in Eq. (6), and the sample rate of the teaching data was 5 Hz.

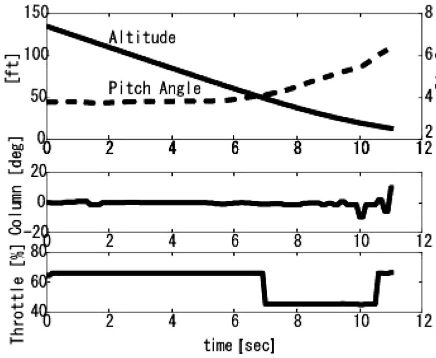
The upper figure in Fig. 6 shows an example of the measured flight data (e.g., the altitude and the pitch angle). The lower figures in Fig. 6 compare the measured pilot-control inputs and the computed data. Figure 6 indicates that the computed output data from the generated network coincide with the teaching data. It should be noted that the proposed GA process was not applied to obtain the network; that is, the initial values of weights/biases and the number of nodes in the hidden layer were appropriately set prior to the training. Although there is an excellent agreement between the two results, the obtained network may have a problem in its generalization capability.

To check the generalization capability, the obtained network was used as an automatic controller in the flight simulation. In many cases, the obtained network cannot perform a successful landing when the network was used as an automatic controller. It is due to a difference in computation between the teaching data obtained from a pilot operation and the simulation data obtained from the NN automatic controller even when the flight condition was the same.

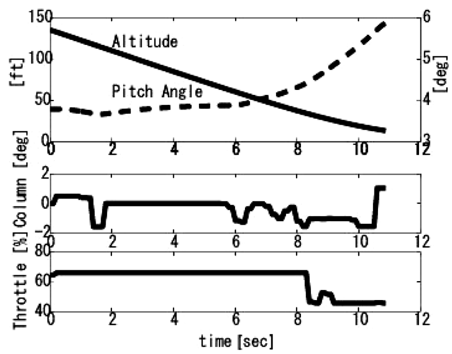
Figure 7 shows the flight data controlled by the best network in some generations. The number of string population was 20, and the number of optimized variables was 211. These variables include the number of hidden-layer nodes and the number of initial values for the weights and biases. Note that the number of hidden-layer nodes was selected from the range of one to eight, the number of hidden layer nodes was represented by three-bit gray code, and the initial values for weights and biases were represented by eight-bit gray code. The upper part of Fig. 7 indicates the initial generation network, which is equivalent to a randomly generated network. Although the zeroth network can successfully land aircraft, the time history of the column and throttle input was quite different from the human-pilot inputs as shown in Fig. 6. Consequently, the zeroth network can mimic the human movements when the teaching data flight was provided; however, the network cannot control the airplane in the exact manner as the human pilot. The zeroth network model does not have the generalization capability. Figure 7 indicates the improvement of the network through the GA process. Although the 10th network is not sufficient, the 30th-generation network can capture the movement characteristics of a human pilot. Figure 8 illustrates the best-cost function between each generation as given in Eq. (7). To evaluate Eq. (7), test data that is slightly different from the teaching data is necessary. Because even the same pilot cannot operate in the exact same manner, the test data that evaluates the generalization capability of an obtained NN model should be chosen from the same flight data. As shown in Fig. 9, the teaching data and the test data were selected alternately from the same flight data. The computation time required to obtain the 30th generation network from 20 strings is about 5 h when a Pentium 4 with a 2.0-GHz central processing unit was used. Furthermore, note that the mutation ratio is selected as the inverse of the string length.

IV. Experimental Results (Training Simulator)

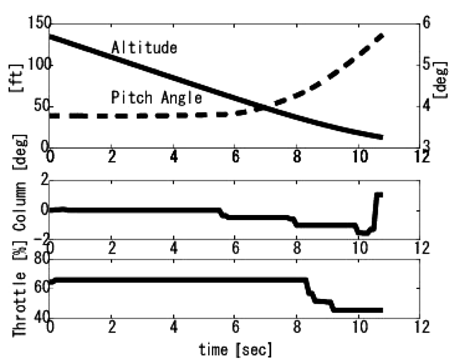
Finally, veteran-pilot and freshman-pilot control inputs were compared using a flight simulator for airline pilot training, as shown



a) 0th generation



b) 10th generation



c) 30th generation

Fig. 7 Simulation results operated by obtained neural networks.

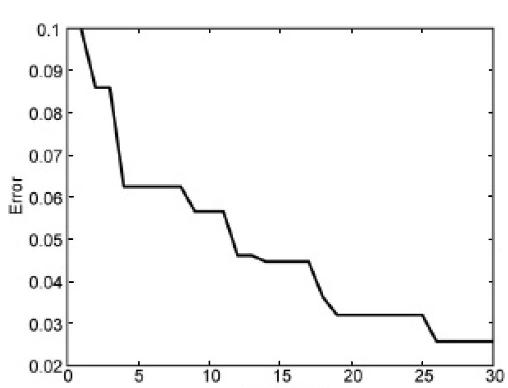


Fig. 8 Cost function vs generations in GA.

in Figs. 10 and 11. The veteran pilot had more than 13 years of experience with B767 aircraft, including more than 5 years as a captain. He had sufficient flight time in jet aircraft, exceeding 10,000 h. The freshman pilot had flight times of around 300 h and 100 h in B767. The visual cues used in the previous section were estimated from the recorded flight data, and these were combined with the control input data as the teaching data set. The NNs improved by the proposed GA process were obtained for the veteran- and freshman-pilot flight data.

Fig. 9 Teaching data and test data.

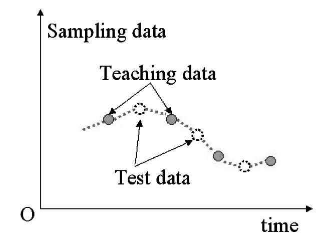
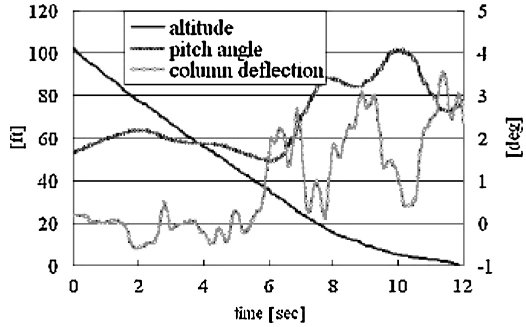


Fig. 10 B767 flight simulator.

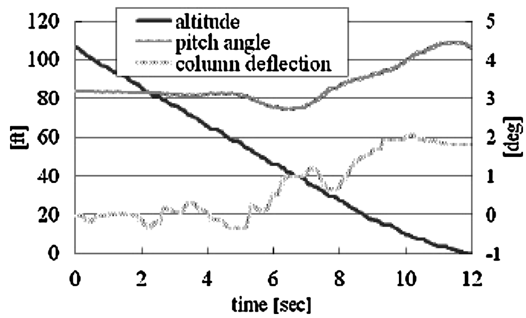
Fig. 11 Landing view from cockpit.

Flight data of altitude, pitch angle, and column deflections indicate that the veteran pilot moved the column quickly and the freshman pilot moved it relatively slowly, as shown in Fig. 12. The contribution ratios of each pilot's data are compared in Fig. 13, which are computed by using the NNs obtained for each landing. It shows that the veteran-pilot quickly changes the contribution ratios according to the situations. The veteran-pilot increased the attentiveness to the horizontal-line information just before touchdown, whereas the freshman pilot did not change the contribution ratios so dynamically. These tendencies coincide well with the pilot's impressions obtained after the experiments.

The sensitivity analysis clearly reveals the difference between the two pilots' control inputs. Figure 14 indicates the sensitivities to the column control for three input data, that is, the column movement to a change in each visual cue. The sensitivity data are categorized as new (present to 0.27 s), intermediate (0.27 to 1.1 s), and old (1.1 to 1.9 s) input data. It is presumed that the veteran-pilot was relaxed in the approach phase and increased his sensitivity just before the flare phase. In contrast, although the freshman pilot displayed high sensitivity during the approach phase but lost sensitivity during the flare phase. This implies that the freshman pilot could not maintain his concentration during the final phase. Three landing data were obtained for each pilot, and the analyzed data indicates a similar trend. Furthermore, the impressions of the two pilots support the analyzed results.

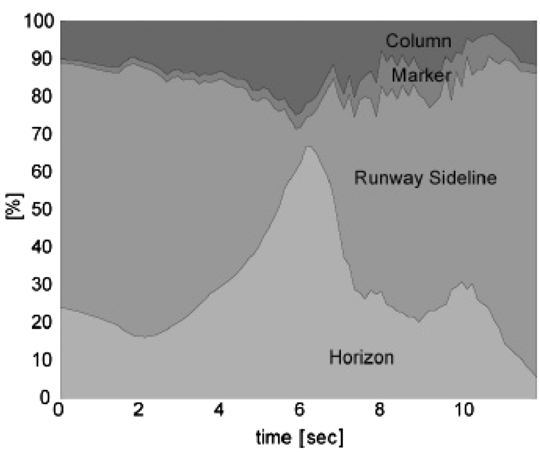


a) Veteran pilot

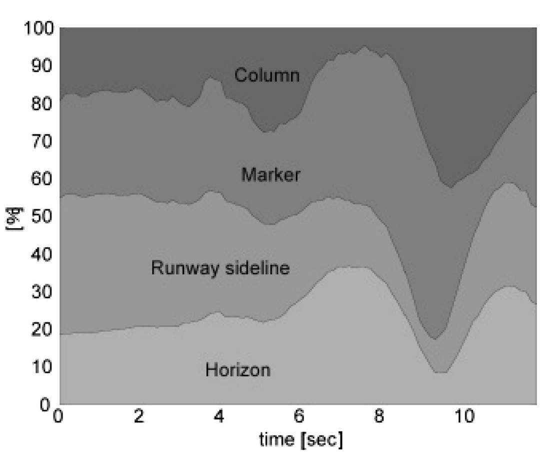


b) Freshman pilot

Fig. 12 Flight data obtained by B767 flight simulator.

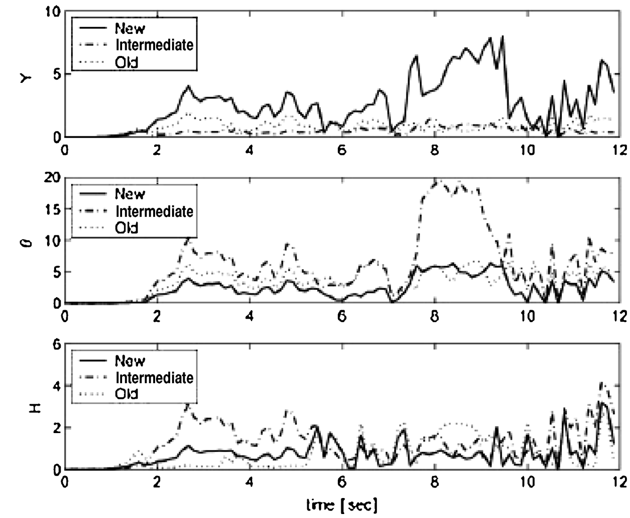


a) Veteran pilot

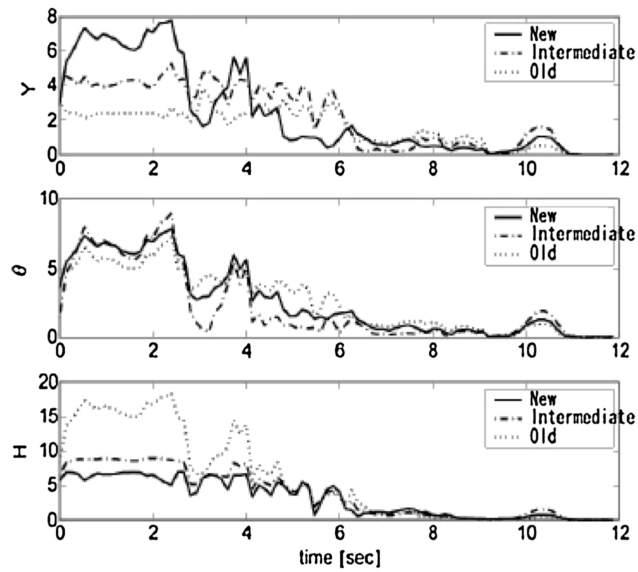


b) Freshman pilot

Fig. 13 Contribution analysis of column input.



a) Veteran pilot



b) Freshman pilot

Fig. 14 Sensitivity analysis of column input.

Because this paper shows the flight data of only one freshman pilot and one veteran pilot, it is impossible to derive generalized results. However, the proposed approach could reveal the difference in the information process flow of each pilot during the visual-landing phase. Upon collecting and analyzing flight data of many pilots, the differences can be used to improve the skill of freshman pilots. In the visual-landing phase, the pilot should recognize the error of approach path based on the visual and motion cues and adjust the flight paths. Although the overall performance can be measured from the flight data, it is difficult to evaluate the cognitive skill of a pilot quantitatively. The proposed contribution and sensitivity analyses can quantify the data information flow from visual cues to control inputs of each pilot. Instructors can thus use these data to recognize the cognitive skill of a pilot. This enables the instructor to advise as well as assist pilots in gaining quantitative understanding of their own control characteristics. Furthermore, the effects of winds or visibility in the control performance of each pilot will be analyzed. Consequently, the proposed analysis method will be used as a training tool for various purposes.[16]

V. Conclusions

The proposed NN-analysis method has the potential to reveal the information-processing flow of human-pilot control inputs. The GA improved the generalization capability of the network. The present study analyzes the movements of human pilots during the landing phase of a large jet transport. The NN models were obtained by learning the visual cues and the control data. The information flow inside the network was analyzed by using the contribution ratio and sensitivity analysis. The generalization capability of the obtained network was confirmed by analyzing the flight data of the PC-based simulator, whose mathematical model was known. Finally, this method was applied to the veteran- and freshman-pilot control inputs of the training simulator. The analysis results indicate that our method can reveal the differences in the characteristics of each pilot. In the near future, the influence of flight conditions, such as visibility and wind effect, will be analyzed, and real flight test data will be considered. Finally, it should be mentioned that this approach is currently under consideration in airline-pilot-training applications.

References

[1]Gibson, J. J., *The Perception of the Visual World*, Houghton Mifflin, Boston, 1950.

[2]Okamoto, H., Kawakami, S., Saito, H., Hida, E., Odajima, K., Tamanoi, D., and Ohno, H., "MT Neurons in the Macaque Exhibited Two Types of Bimodal Direction Tuning as Predicted by a Model for Visual Motion Detection," *Vision Research*, Vol. 39, No. 20, 1999, pp. 3465–3479.

[3]Schmidt, D. K., and Silk, A. B., "Modeling Human Perception and Estimation of Kinematic Responses During Aircraft Landing," *Proceedings of the Guidance, Navigation, and Control Conference* (Minneapolis, MN), AIAA, Washington, DC, 1988, pp. 1117–1126.

[4]Kuriki, Y., Suzuki, S., Uemura, T., and Kitaki, Y., "Research on State Estimation of Pilot in Landing Aircraft," *Journal of the Japan Society for Aeronautical and Space Sciences*, Vol. 49, No. 571, 2001, pp. 246–254 (in Japanese).

[5]Arbib, M. A., *Brains, Machines and Mathematics*, 2nd ed., Springer Verlag, New York, 1987.

[6]Holland, J., *Adaptation and Natural and Artificial System*, Univ. of Michigan Press, Ann Arbor, MI, 1975.

[7]Kageyama, I., and Arai, A., "Research on Driver's Information Process at Accident-Prone Area," *The 7th Transportation and Logistics Conference 2003*, Japanese Society of Mechanical Engineers, Tokyo, pp. 227–230 (in Japanese).

[8]Martens, D., "Neural Networks as a Tool for Assessment of Human Pilot Behavior in Wind Shear," *Aerospace Science and Technology*, No. 1, 1999, pp. 39–48.

[9]Hayashi, K., Suzuki, S., and Uemura, T., "Analysis of Human Pilot Behavior at Landing with Neural Network," *Journal of the Japan Society for Aeronautical and Space Sciences*, Vol. 49, No. 564, 2001, pp. 21–26 (in Japanese).

[10]Suzuki, S., Hayashi, K., and Uemura, T., "Analysis of Human Pilot Behavior at Landing with Neural Network 2nd Report," *Journal of the Japan Society for Aeronautical and Space Sciences*, Vol. 51, No. 590, 2003, pp. 109–113 (in Japanese).

[11]Emmerson, M. D., and Damper, R. I., "Determining and Improving the Fault Tolerance of Mutilayer Perception in a Pattern-Recognition Application," *IEEE Transactions on Neural Networks*, Vol. 4, No. 5, 1993, pp. 788–793.

[12]Neti, C., Schneider, M. H., and Young, E. D., "Maximally Fault-Tolerant Neural Networks," *IEEE Transactions on Neural Networks*, Vol. 3, No. 1, 1992, pp. 14–23.

[13]Tan, Y., and Nanya, T., "Fault-Tolerant Back-Propagation Model and Its Generalization Ability," *Proceedings of International Joint Conference Neural Networks*, IEEE Publications, Piscataway, NJ, 1993, pp. 2516–2519.

[14]Ham, F. M., and Kostanic, I., *Principles of Neurocomputing for Science and Engineering*, McGraw–Hill, New York, 2001.

[15]Heffley, R. K., and Jewell, W. F., "Aircraft Handling Qualities Data," NASA CR-2144, 1972.

[16]Takahara, H., Kondo, T., and Suzuki, S., "Study of Landing Technique During Visual Approach," ICAS2004-7.11.2, ICAS, Stockholm, Aug.–Sept. 2004.