NEURAL NETWORK ANALYSIS OF PILOT MANEUVER DURING LANDING PHASE

Ryota Mori*
***University of Tokyo**

Keywords: *Neural Network, Pilot Control, Human Control, Landing*

Abstract

Pilot control at the visual approach has been modeled using neural networks. The quality of the constructed pilot model depends on the given inputs and learning scheme because measured data includes noise and uncertain inputs. In order to cope with these uncertainties, a new learning scheme is proposed in this paper. Using the proposed scheme, pilot controls are analyzed for different flight conditions, such as day/night and no wind/gusty. The contribution ratios and sensitivity analysis results show clear differences between the flight conditions.

1 Introduction

Recent aircraft are equipped with a sophisticated autopilot system. While cruising, the autopilot system is fully used, and it is an indispensable part of the aircraft. On the other hand, during landing approach, the auto landing system is rarely used. First of all, this system can only be used if both aircraft and airport fulfill the requirements for auto landing. In addition, if the pilot relies on the auto landing system too often, it might lead to the pilot losing his skill. Furthermore, a computer can perform exact operations, and it does not has flexibility for uncertain conditions such as cross wind landings. Consequently, the pilot mostly performs a manual landing, and it is considered that the pilot control will remain important in the future.

Additionally, pilot control skills are mostly obtained through experience, and it is difficult to teach such skills to junior pilots. Some of the accidents seem to be due to a lack of skill, which is the reason to develop a tool that can examine a pilot's control skill quantitatively.

A method has been developed to evaluate control skill using neural networks (NN).[1] It is expected that this method will be used as a skill evaluation tool at airline training centers. One problem for airlines is to train new pilots, because it is very costly and difficult to make a guideline for control. The proposed method will help to clarify pilots' information processing flow and improve the efficiency of the pilot training.

The target of this study is the flare maneuver, which is a pitch-up control during the final landing phase, because it is said to be the most difficult maneuver in normal operations for airline pilots. During this phase, a pilot gets visual cues from the out-of-the-window view, because he or she has to obtain much information, and there is no time to watch instrument panels. The pilot has to estimate the aircraft state values, such as pitch angle and altitude, from visual cues. In our analysis, the relationship between visual cues and the pilot's control is modeled using a NN which is trained so as to simulate the recorded pilot maneuver for recorded visual cues. The obtained NN is analyzed mathematically.

In previous studies [1][2], the proposed NN modeling technique has been validated in several ways. However, in those studies, manual tuning of the NN parameters is necessary, which is an obstacle for making a pilot training tool. Thus, a new method with a fully automatic learning scheme is proposed in this paper. Additionally, using the proposed new method, pilot control under various flight conditions is analyzed and the differences are revealed.

2 Neural Network Modeling

2.1 Neural Network Modeling of the Landing Phase

2.1.1 Neural Network

Artificial NNs[3][4] are analogous to the biological nervous systems, and consist of smaller units called neurons which are grouped in layers and perform only a few simple operations. This neuron operates very simply like this expression:

$$y = f\left(\sum_{i=1}^{n}(w_i x_i + b/n)\right) \qquad (1)$$

where

$$f: \begin{cases} x \mapsto (e^x - e^{-x})/(e^x + e^{-x}) & \text{(hidden layer)} \\ x \mapsto x & \text{(output layer)} \end{cases},$$

and x_i, y, and w_i are the input, the output, and weights, respectively, and b is a bias of the activation function f. In this study, a normal three-layer hierarchical network is applied. This network is capable of very complex mappings due to the high interconnectivity of neurons in subsequent layers, and it is used as the pilot model and analyzed mathematically. In order to make the NN work as the pilot model, weights and biases have to be set. This process is called learning. For learning, the teaching data sets, which are model inputs and outputs, should be prepared. The learning minimizes the objective function F, which is normally the mean square error (MSE) between the teaching output data and the NN computed output data. The error back-propagation method is usually used as a learning scheme. In this paper, a scaled conjugate gradient algorithm[5] is applied, which uses an advanced conjugate gradient method.

2.1.2 Model Outline

In this study, a pilot model is constructed with a NN and the pilot's control characteristics are discovered with an analysis of the obtained NN. Thus, the NN inputs — which are the facts on which the pilot bases his control — should be chosen appropriately.

During the landing, the pilot receives the visual cues as main information, because the pilot generally cannot afford to watch the instrument panels. The optical flow of visual cues, such as the position of the horizon or the runway shape, can help the pilot estimate the current aircraft state values.[6] In the current work, some visual cues are quantified as NN inputs to construct a pilot model, and the column movement and throttle setting are used as outputs, and the relationship between inputs and outputs has been analyzed. Fig. 1 shows the quantified visual cues in this study, and Table 1 shows mathematical relationships between quantified visual cues and aircraft state values. Note that only longitudinal motion is considered. The neural network structures for each output are shown in Fig. 2. It should be noted that not only the visual cues but also their time derivatives and time lagged data are used as NN inputs, because data flow and time-delayed information also affects the pilot control and human response has a time delay.

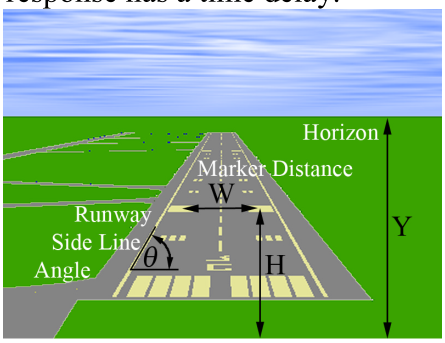

Fig. 1 Visual Cues.

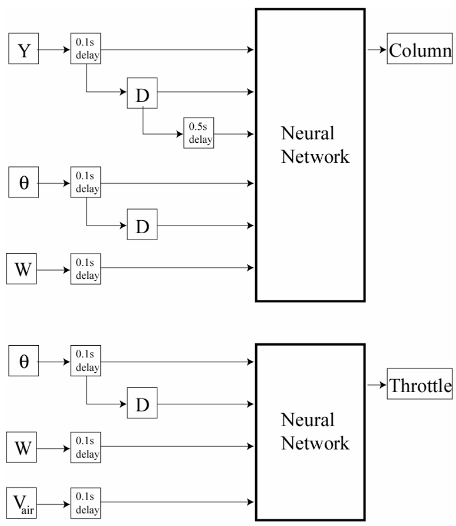

Fig. 2 Network Structure.

Table 1 Relationship between Visual Cues and Aircraft States

Visual Cue	Related Main Aircraft State
Y	Pitch angle
θ	Altitude
W	Distance to Runway

2.2 Neural Network Construction Problems

2.2.1 Generalization

When constructing a neural network model, one problem is how to obtain a good generalization. Generalization means that the NN model can generate reasonable outputs in the various situations, which were not included in the teaching data sets. It is obviously true that the NN cannot deal with every situation, because the NN learns just a few typical cases of a pilot's control. However, a NN sometimes imitates the noise of human control, and it does not act like the pilot when the inputs go off the teaching data sets. Many methods with good generalization capability have been proposed for constructing or training NN, but there is no well-understood method at present. However, it is considered that this problem can be solved by choosing the right learning scheme, and some other methods had been considered in a previous study.[7]

2.2.2 Regularization

One method to improve generalization is to use regularization. As previously noted, the normal objective function consists of only the MSE between the NN output and the teaching output data. The teaching output data is the pilot control which includes much human noise. The human noise consists of hand tremor, misconception, non-typical environmental conditions, and confirmative control of aircraft reaction, which are difficult to model and do not have to be modeled.

It is said that the absolute values of the network weights become larger, when fitting more detailed. With the use of this characteristic, the weight values are added to the objective function F as follows.

$$\begin{aligned}\alpha &= \gamma \sum (x_{NN} - x_{teach})^2 / N \\ \beta &= (1-\gamma) \sum w_{ij}^2 / 2n \\ F &= \alpha + \beta\end{aligned} \qquad (2)$$

where γ is a regularization parameter, and N is the number of teaching data sets, and n is the number of all weights. If γ is equal to 1, regularization does not work. The smaller γ is, the more noise is removed because of the latter term. However, if γ gets too small, necessary output data is also treated as noise, and the NN model loses the ability to reproduce the teaching data. Thus, γ should be decided properly. There is a method (Bayesian Regularization method[8]) to decide a proper value for γ. However, this method is based on the Bayesian probabilistic framework, which assumes the noise is white noise. The human noise is very different from the white noise, so another method should be considered.

2.2.3 General Neural Network Problems

The proposed analysis method is to be used as a pilot training tool at airlines, so the learning has to be completely automated. However, when learning, the initial weights and biases should be decided randomly, and this randomness sometimes causes the random convergence, which means the NN converges to a local solution. If they are decided manually every time, it is very time consuming, and it contains subjectivities, which means that the pilot control cannot be analyzed objectively. This problem can be solved by decreasing local minima to some extent by refining the inputs of the network, but this is not enough. In addition, it is widely said that the number of hidden neurons is also important to keep the generalization of the network.

3 Automation for Pilot Modeling

3.1 Proposed Learning Method

In order to solve problems of overfitting, bad generalization, and convergence to local minima, a new training method is proposed. The concept is that the network is trained including noise with high gamma first, and then the noise is removed from the network decreasing gamma gradually. It is known that the proper value of gamma is 0.4 to 0.9 by experience. The detailed flow of learning is written as follows. Fig. 3 also shows the flow of learning a network.

1) The initial weights and thresholds are decided randomly. Gamma is set to a value a little higher than 0.9 (e.g. 0.95). The network is trained to minimize α+β as defined in Eq. 2 until α+β is converged (the gradient reached sufficiently small value).
2) Gamma is set to a slightly lower value (e.g. 0.949), and the network is trained again. This is repeated until the gamma is 0.4. (e.g., 0.948, 0.947, …, 0.401, 0.400)
3) The network with lowest α/(α+β) is chosen.

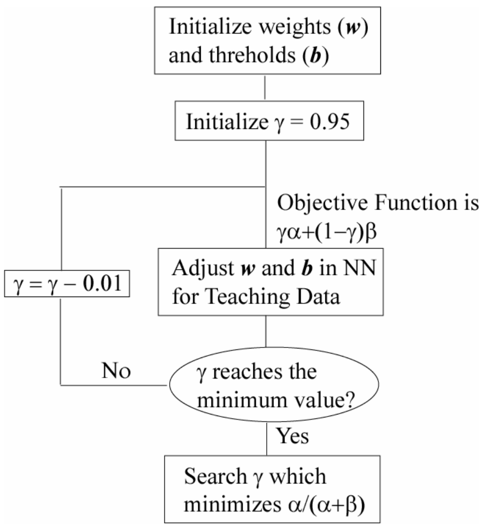

Fig. 3 A flow of the proposed learning method.

This proposed method can solve many problems discussed before. With the old method, firstly, even if the gamma is fixed to one value, there are local minima, and the network does not always converge to the same structure as it depends on the initial network values. However, using the new method, the gamma and the network have a one-to-one relation, which means that the network does not depend on the initial network values but only depends on the gamma. Fig. 4 shows one example of time histories of learning. The red points mean that the gamma is fixed to some values using the old method. The black points (line) mean that the new method is used. The learning in each method is carried out 20 times. The figure indicates that the MSE depends on only the gamma using the new method when the gamma is between 0.4 and 0.9. On the other hand, the MSE has several discrete values using the old method.

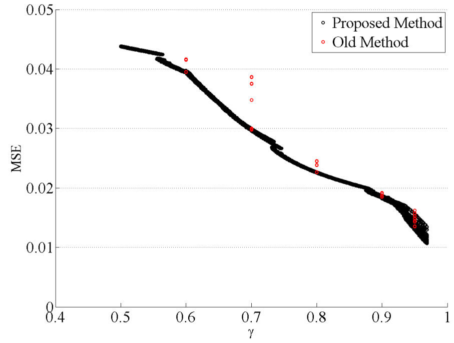

Fig. 4 MSE vs. gamma between the proposed method and the old method.

Secondly, the gamma can be decided automatically. One indicator — minimum of α/(α+β) — is proposed. When gamma decreases, the learning can be separated into two stages — “deleting noise stage” and “over deleting stage”. “Deleting noise stage” means that the network is deleting noise, which is a proper stage. “Over deleting stage” means that the network is deleting necessary parts of the network, which is an improper stage. The noise accounts for a relatively small part of the network, which means that α increases steeper in the over deleting stage. On the other hand, in the over deleting stage, the necessary part of the network accounts for relatively bigger part of the network, which means the necessary weights also account for the most part. The β fundamentally increases with regard to the decrease of the gamma, but in the over deleting stage, the main parts of weights are deleted, and β increases slowly or decreases. What it comes down to is that the turning point between the deleting noise stage and the over deleting stage gives the minimum value of α/(α+β). Fig. 5 shows one example of time histories of α, β, and α/(α+β). It shows that γ = 0.764 gives the minimum of α/(α+β). The effectiveness of this way of deciding the gamma is verified in the next section.

Thirdly, using the proposed method, the number of hidden neurons does not have to be considered. The term of β works to decrease the magnitude of weights, so it is expected that the unnecessary weights are deleted automatically. This means that the excess of hidden neurons does not matter. Fig. 6 shows one example with 15, 20, 25, and 30 hidden nodes. In each case, the minimum value of α/(α+β) is determined, and the corresponding values of gamma are 0.764, 0.721, 0.661, and 0.618, respectively. The corresponding MSE are 0.0244, 0.0237, 0.0243, and 0.0243, respectively, which have almost the same value with different number of hidden nodes. This means that the network structures also seem to be the same. This fact is also verified with Monte Carlo simulations in the next section.

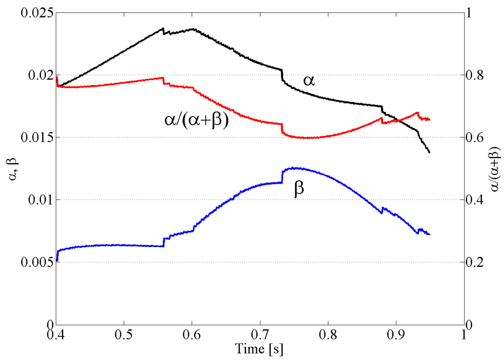

Fig. 5 α, β, α/(α+β) vs. γ.

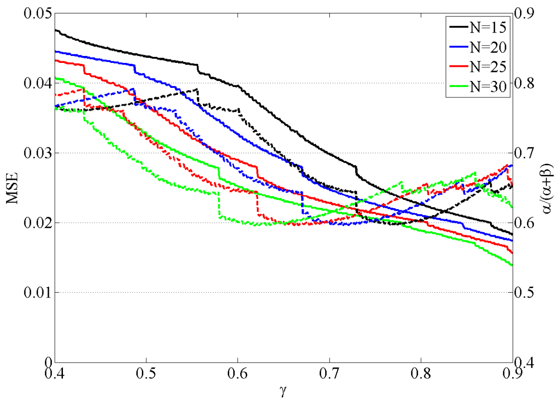

Fig. 6 α/(α+β)[dotted line], MSE[solid line] vs. gamma with different number of hidden nodes.

3.2 Experiment and Simulation Conditions

The evaluation of the generalization of the network by only the MSE and the magnitude of weights is difficult. Thus, Monte Carlo simulations are carried out to evaluate it. Firstly, the flight data should be obtained. A PC-based simulator (Fig. 7) using a mathematical model of a big jet airliner was used, and a B747 airline captain pilot was asked to land the aircraft three times to generate training data for the NNs (for initial settings see Table 2). The visual cues were computed from the recorded flight data. Of this data, the data under the altitude of 200 ft is extracted as teaching data, because the pilot commented that at 200 ft or a little less is the turning point to the visual approach completely. Note that only longitudinal motion was simulated, and the data sampling ratio was 10 Hz.

Fig. 7 PC-based Simulator.

In order to check how much the generalization capability of the proposed methods can improve, Monte Carlo flight simulations are carried out. The obtained NN model is used as an automatic controller, and flight simulations are carried out many times with a random variation of initial flight conditions. The range of initial flight conditions are configured as shown in Table 2. The Monte Carlo simulations are carried out 200 times each, and the criterions of the simulation results are sink rate, pitch angle, and the position of the landing.

Table 2 Initial Condition for Monte Carlo Simulation

Initial Condition Case	Initial Velocity	Initial Path Angle	Initial Pitch Angle	Initial Position from Markers
Monte Carlo Simulation	258±5 ft/s	–3±1 deg	1.72±1 deg	4016±500 ft
Teaching Data 1	259.9 ft/s	–3.10 deg	1.73 deg	3715 ft
Teaching Data 2	257.8 ft/s	–2.81 deg	2.23 deg	4535 ft
Teaching Data 3	256.2 ft/s	–2.77 deg	2.11 deg	4129 ft

3.3 Simulation Results

3.3.1 Gamma Optimization

In the previous section, the gamma optimization method was explained. Thus, in this section, it is explained how the gamma is optimized. Two separate networks are trained with column deflection and throttle deflection as their respective outputs, so both gammas are optimized. The optimized gammas are 0.764 and 0.891, respectively. For comparison, the results with gamma (0.6, 0.7, 0.75, 0.8, 0.85, and 0.9) are also shown in Fig. 8. Three criterions are separated into several stages; good (blue), normal (sky blue), acceptable (yellow), bad (red), and it shows the percentage of each stage for 200 Monte Carlo simulations. If the blue part is big, it means that the landing performance is good. For sink rate, with high gamma, there are both good and bad landings, while with low gamma, most landing are normal. With optimized gamma, there are many good landings and few bad landings. For pitch angle, higher gamma shows the bad landings. For landing position, lower gamma shows the relatively few good landings. From these results, it can be said that the optimized network shows good performance for every criterion.

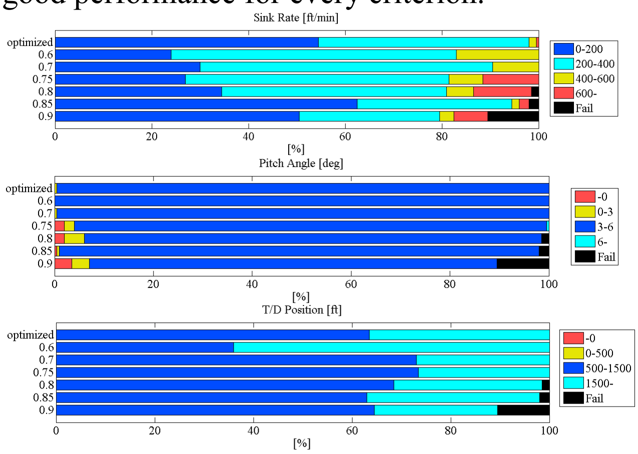

Fig. 8 NN Performance for several gamma settings in Monte Carlo simulations.

Trajectories of simulations show more clear characteristics of each gamma. Fig. 9 shows the trajectories of Monte Carlo simulations with different gammas. The black lines indicate the Monte Carlo simulation results, and the red lines indicate the obtained flight trajectories. When gamma is 0.9, there are good landings, but sometimes the NN controller makes the aircraft climb instead of land. When gamma is 0.4, the simulation results deviate much from the real flight trajectories, especially just before the landing. It means that the network cannot imitate the real pilot characteristic. With the optimized gamma, the result shows the good performance for almost every landing.

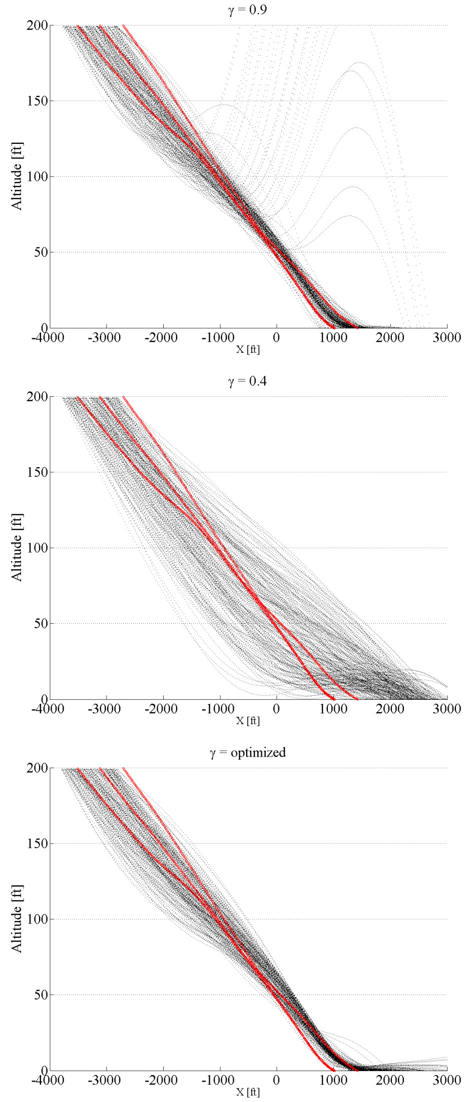

Fig. 9 Trajectories for Monte Carlo simulation for several gamma settings.

3.3.2 The Number of Hidden Nodes

As explained in section 3.1, the network performance does not seem to depend on the number of hidden nodes, which is verified in this section. The landing results with Monte Carlo simulations are shown in the same manner as Fig. 8 (discussed in section 3.3.1). The number of hidden nodes is chosen as 15, 20, 25, and 30 with the optimized gammas 0.764, 0.721, 0.661, and 0.618, respectively. Fig. 10 shows that no significant difference can be found between all cases, which means that the number of hidden nodes does indeed not affect the construction of the network. Of course, the excess of hidden nodes causes slower training, so a too high number of hidden nodes is not appropriate for learning.

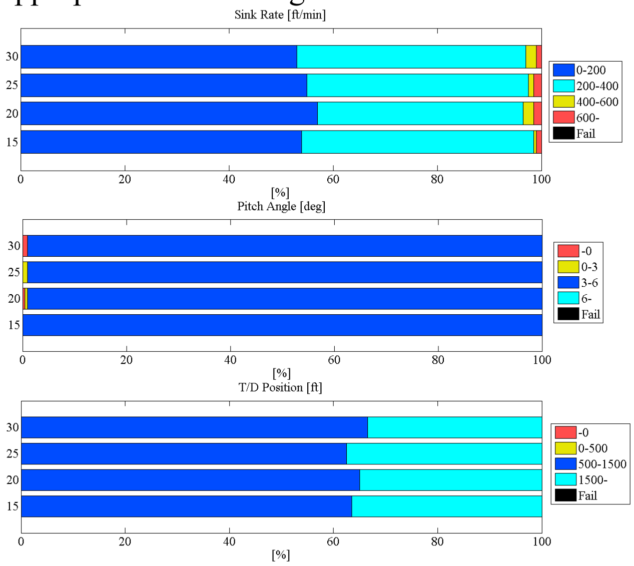

Fig. 10 Simulation Results with different number of hidden nodes.

4. Pilot Analysis Results for Different Flight Condition

4.1 Background of Analysis

4.1.1 Experiment Conditions

Using the proposed method explained in the last section, pilot controls under different conditions are analyzed. The experiments were carried out with a B767 flight simulator owned by All Nippon Airways (Fig. 11). This flight simulator is used for pilot training. A B767 captain pilot operated this simulator three times in each condition, and the flight conditions are summarized in Table 3. The obtained flight and control data are used to construct a neural network model in each condition. The initial altitude is 200ft, the same as in the last section.

Fig. 11 ANA Flight Simulator.

Table 3 Experiment Condition.

Case	Wind	Day/Night
Normal	No Wind	Day
Gusty	Cross wind & Gust	Day
Night	No Wind	Night

4.1.2 Analysis Methods

In order to analyze a pilot's control, a contribution ratio analysis and a sensitivity analysis are proposed.

In the contribution ratio analysis, the influence of each visual cue can be recognized. This method is based on the weights method. The weights method was first proposed by Garson[9], and the procedure can be used for partitioning the connection weights of the NN's neurons to determine the relative importance of the various inputs. The contribution with the improved weights method can be calculated as follows:

$$R_{k,i}(t) = \frac{\sum_{j=1}^{m} \mathrm{std}(w_{j,i}^{h} x_i(t)) \cdot \mathrm{std}(w_{k,j}^{o} y_j^{h}(t))}{\sum_{i=1}^{n} \sum_{j=1}^{m} \mathrm{std}(w_{j,i}^{h} x_i(t)) \cdot \mathrm{std}(w_{k,j}^{o} y_j^{h}(t))} \quad (3)$$

where std means the standard deviation. In some researches, it is concluded that this weights based method is not a good indicator[10], because the unnecessary neurons also contribute to the calculation. However, in this research, the unnecessary neurons are deleted with the regularization method, and it is confirmed that this method works well with a linear control model.

In the sensitivity analysis, the degree of change of the output is calculated for small changes of inputs. The sensitivity can be calculated with the following expression:

$$S_{k,i} = \frac{\partial y_k^o}{\partial x_i} = \frac{\partial}{\partial x_i} \sum_{j=1}^{m} (w_{k,j}^o y_j^h + b_k^o / m) \qquad (4)$$

When constructing the network, the inputs and outputs are scaled to the interval [–1, 1], but the sensitivity is scaled back to the absolute values (which have a unit), so they can be compared between different experiments. Moreover, if the sensitivity is assumed to be like a feedback gain in a linear model, it can be seen whether the direction of the sensitivity is stable or unstable. For example, if the pitch angle gets lower, the stable direction of the column is to pull up. In this paper, the direction of the sensitivity is also discussed.

4.2 Pilot Analysis Results

4.2.1 Contribution Ratios Analysis

Table 4 shows the contribution ratios. This result shows some interesting characteristics. In the normal case, the ratio of dθ/dt is higher, while the ratio of θ is lower than in other cases. Note that θ is an inclination of the runway sidelines, and it depends on the altitude information. According to the pilot comment, θ is the most important cue for learning. The derivative information is normally used to estimate the aircraft movement in the near future. In the gusty wind case, it is difficult to estimate the near future condition because of the gusty wind, and as a result, the cue is shifted from the derivative of θ to θ itself.

Pilot comments were also obtained about night landing. He says that it is difficult to recognize the necessary information compared to a day time landing, especially for altitude information. A false sense is sometimes caused, so he gives extra attention to the altitude information. This comment reflects on the higher ratio of θ than normal case. The ratio of dY/dt is also shifted to the ratio of Y in the night case. Note that Y indicates the position of the horizon, and it corresponds to the pitch angle. He maybe cannot afford to pay attention to the change (derivative) of the information, and the ratio of dY/dt and dθ/dt decreases.

4.2.2 Sensitivity Analysis

For the sensitivity analysis, only characteristic results are shown. Figs. 12, 13, and 14 show the sensitivity to Y, θ, and dθ/dt, respectively. Three landing data sets were obtained in each case, but only one set is shown because all three data have similar trends. The sensitivity to θ has a similar trend between cases; it is first close to zero and it increases at almost the same time. This increase corresponds to the flare maneuver. However, in the night case, the magnitude of increase is the biggest, which implies the pilot's close attention to the altitude information at night. In addition, the sensitivity to θ increases monotonously in the normal and night cases, while it decreases at about –8 s once. It is considered an influence of a wind gust, and he tried to take care of the windy situation.

In the sensitivity to Y and dθ/dt, the sensitivity in case normal is varying more widely than other cases, which implies that the pilot tries to control daringly and attempt a soft landing because the flight condition is good.

Moreover, in the gusty case, the characteristic wind at the airport is simulated, which is extremely gusty between 100 and 200 ft of altitude. It corresponds to the high ratio of dθ/dt between –15 s and –8 s, while this type of increase cannot be seen in the other cases. It is considered that the gusty wind is dealt with according to the change of the altitude.

Table 4 Contribution Ratios Analysis Reults

Case	Y	dY/dt	θ	dθ/dt	W
Normal	17.3%	23.3%	18.9%	16.1%	24.4%
Gusty	13.9%	23.0%	24.4%	11.2%	27.6%
Night	19.6%	18.3%	22.5%	10.9%	28.7%

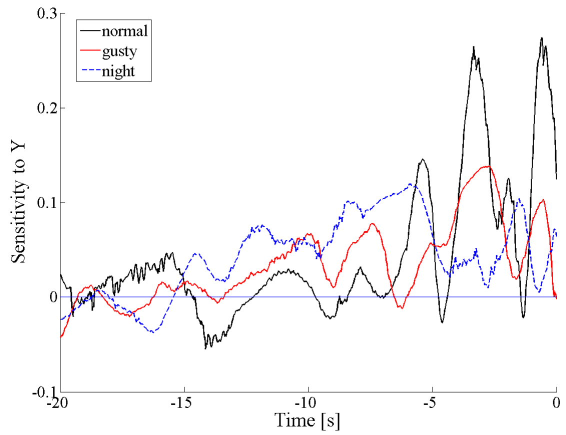

Fig. 12 Sensitivity Analysis to Y.

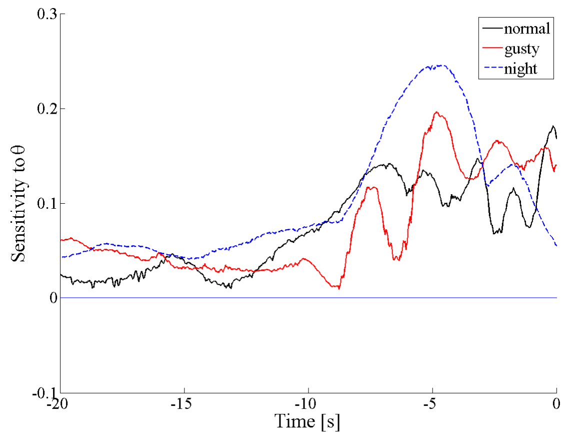

Fig. 13 Sensitivity Analysis to θ.

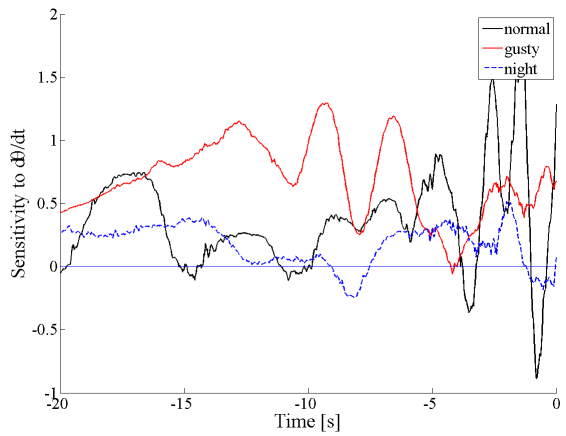

Fig. 14 Sensitivity Analysis to dθ/dt.

5. Conclusions

The NN analysis helps us to know a pilot's information processing flow from visual cues to control operations for a visual approach. In this paper, a new learning scheme was proposed, and it enables to construct a NN pilot model with good generalization capability without careful manual tuning. Using this proposed method, the pilot controls under different flight conditions, such as gusty and night condition, are analyzed with contribution ratios and sensitivity analysis. These results reveal the control strategy of the pilot, and clarify the differences between flight conditions. It is expected that this analysis can be used for an airline pilot training, and it will help the efficient pilot training of pilots. In a future study, this analysis should be extended to lateral control, and characteristic landings with cross wind condition will be analyzed.

Acknowledgement

It is noted that this research is supported by Grant-in-Aid for "Scientific Research A" from Japan Society for the Promotion of Science (JSPS). Additionally, the simulator experiments were collaborative with All Nippon Airways as a joint research.

References

[1] Suzuki, S., Sakamoto, Y., Sanematsu, Y., and Takahara, H., "Analysis of Human Pilot Control Inputs using Neural Network," *Journal of Aircraft*, Vol. 43, No. 3, pp 793-796, 2006.

[2] Mori, R., Suzuki, S., Sakamoto, Y., and Takahara, H., "Analysis of Visual Cues during Landing Phase by using Neural Network Modeling," *Journal of Aircraft*, Vol. 44, No. 4, pp 2006-2011, 2007.

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

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

[5] Moller, M. F., "A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning," *Neural Networks*, Vol. 6, pp. 525-533, 1993.

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

[7] Mori, R., Suzuki, S., and Takahara, H., "Optimization of Neural Network Modeling for Human Landing Control Analysis," *AIAA Infotech@Aerospace 2007 Conference and Exhibit*, 7-10 May. 2007, Rohnert Park, AIAA-2007-2840.

[8] David J. C. MacKay, "A Practical Bayesian Framework for Backpropagation Networks," *Neural Computation*, Vol. 4, No. 3, pp. 448-472, 1992.

[9] Garson, G. D., "Interpreting neural network connection weights," *Artificial Intelligence Expert 6*, pp. 47–51, 1991.

[10] Gevrey, M., Dimopoulos, I., and Lek, S., "Review and comparison of methods to study the contribution of variables in artificial neural network models," *Ecological Modeling*, Vol. 160, pp. 249-264, 2003.

Copyright Statement