Acquisition of Human Operation Characteristics for Kite-based Tethered Flying Robot using Human Operation Data

Acquisition of Human Operation Characteristics for Kite-based Tethered Flying Robot using Human Operation Data

Chiaki Todoroki and Yasutake Takahashi
Dept. of Human & Artificial Intelligent Systems,
Graduate School of Engineering, University of Fukui
3-9-1, Bunkyo, Fukui, Fukui, 910-8507, Japan
Email: {ctodoroki,yasutake}@ir.his.u-fukui.ac.jp

Takayuki Nakamura
Department of Computer and Communication Sciences,
Faculty of Systems Engineering, Wakayama University
Sakaetani 930, Wakayama 640-8510, Japan
Email: ntakayuk@sys.wakayama-u.ac.jp

***Abstract*—This paper shows human skill acquisition systems to control the kite-based tethered flying robot. The kite-based tethered flying robot has been proposed as a flying observation system with long-term activity capability[1]. It is a relatively new system and aimed to complement other information gathering systems using a balloon or an air vehicle.**

This paper shows some approaches of human operation characteristics acquisition based on fuzzy learning controller, k-nearest neighbor algorithm, and artificial neural network for the kite-based tethered flying robot using human operation data and their validity through computational simulation which we developed[2].

I. Introduction

Autonomous observation systems using a balloon[3], [4] or an airplane[5], [6], [7] have been studied as a solution of information gathering systems from the sky. The balloon system is able to stay in the sky for a long time. However, the helium gas reservation is necessary and it needs a relatively long time and specialists of gas maintenance for the flight preparations. Furthermore, the balloon system is not available under strong wind condition. On the other hand, an airplane system needs less time for flight preparations, but long-term activity is difficult due to limitation of fuel.

In order to complement those information gathering system, we have proposed a tethered flying robot based on a kite that flies with wind power as one of the natural power sources and conducted some experiments with a real robot we designed and built[1]. We also proposed a fuzzy controller for the robot inspired by how to fly a kite by human[8]. It is hard to ensure that the fuzzy rule table is correct in various wind situations and to define the table if the number of state variables or membership functions is big. Furthermore, the proposed 3-inputs 1-output fuzzy controller in the previous paper, however, has some problems to control the kite stably in condition of various change of wind because of lack of information for state description.

This paper aims at acquiring human operation characteristics for the kite-based flying robot using human operation data. Besides learning the fuzzy control parameters, we investigate k-nearest neighbor algorithm and artificial neural network for the purpose on the computational simulation developed based on the real robot experiment data[8].

This paper gives two contributions. First, we show that the improved fuzzy control system has characteristics similar to human operation compared to the previous system. Second, we propose other human skill acquisition methods based on k-nearest neighbor algorithm and artificial neural network, and show their validity with the computational simulations.

II. Kite-based Tethered Flying Robot

Our tethered flying robot consists of a kite, a flight unit, a tether line, and a ground control unit. The concept of the kite-based tethered flying robot is shown in Fig.1. The flight unit carries sensors that measure the surrounding wind state and the position of the flight unit itself, and also transmits to the ground control unit wirelessly. The flight unit is lifted from the ground by a kite. The ground control unit controls the tether line attached to the flight unit according to the data sent by the flight unit. A ZigBee module is used for wireless communication between the flight unit and the ground control unit. More details of the real robot is described in [1], [8].

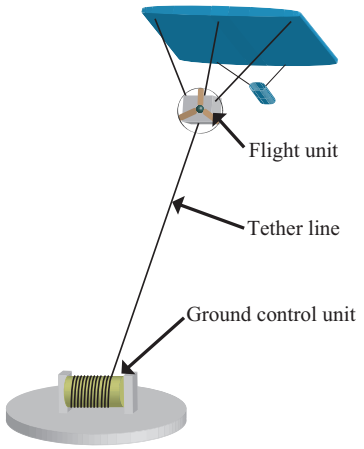


Fig. 1. Concept of kite-based tethered flying robot

Figure 2 shows the 2D model of our kite-based tethered

flying robot. It also illustrates lift and drag force of the kite, force to the drag tail, gravities of materials, tension of lines, and so on. A kite might change its shape according to the

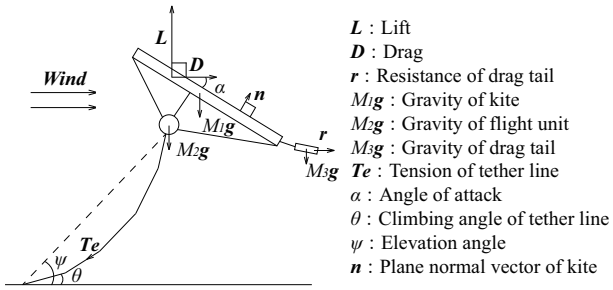


Fig. 2. Computational model of kite-based tethered flying robot

wind situation around the kite. We assume that the kite does not change shape too much during the flight because the wind power is strong enough. Therefore, the kite is modeled as a board in the dynamics engine simulator. Open Dynamics Engine (ODE)[9] is utilized for our dynamics simulator of the kite-based tethered flying robot. The drag force of the drag tail of the kite is measured through real experiments with the drag tail of the kite and modeled to use them in the dynamics simulator. The tether line is modeled with a set of small pieces of rigid sticks because the ODE does not have a flexible line model. The tether line is composed of 20 small rigid stick-type bodies in this paper. The line between the kite and drag tail is also modeled in a same way. This line is composed with 3 small rigid stick-type bodies. Each rigid stick-type body takes gravity into account. The line-winding behavior of the ground control unit is modeled as dragging the tail of the tether line. The ODE itself does not take the air friction into consideration. The real kite-based tethered flying robot has influence over air friction. Therefore, air viscous friction is added to each rigid body on the ODE in the direction opposite to the object relative velocity to the air. In real experiments, a safety line is attached to the flight unit in order to avoid releasing the flight unit just in case the main tether line is cut off. The simulator takes it into consideration. The gravity force and the wind viscous friction are added to the dynamics simulation. The actual flying robot catches the wind from the side, but, the computer simulator does not take it into consideration. More details of the computational model, and comparison between data from model and real robot are described in [2].

III. Flying Control

This section shows three learning methods to design the flying controller based on the human operation data. This paper investigates a fuzzy controller, k-nearest neighbor algorithm and artificial neural network system that is described in subsection III-A, III-B and III-C, respectively.

A. Flying Control by Fuzzy Control

Our fuzzy controller is inspired by how a human flies a kite. Fuzzy set can reasonably represent unwritten strategy for flying a kite by human. A human controls the line of the kite based on the wind speed, the drag force of the line, the altitude and the motion of the kite, and so on. We consider that wind speed and altitude are essential to reflect human operating characteristics. Moreover, it is necessary to take motion of the kite into account.

Here, we design 3-inputs 1-output fuzzy controller for the flight. It controls the drag and release force based on a fuzzy set of altitude, altitude change of the kite and wind speed measured by the flight unit. The fuzzy controller is represented based on a simplified reasoning method by Eqs.(1) and (2).

Rule i : if W_t is WS_i and A_t is ALT_i and dA_t is $DALT_i$ then φ is b_i $(i = 1, 2, ..., \mathrm{n})$

$$h_i = \min(\mu_{WS_i}(W_t), \mu_{ALT_i}(A_t), \mu_{DALT_i}(dA_t)) \quad (1)$$

$$\varphi = \frac{\sum_{i=1}^{n} h_i b_i}{\sum_{i=1}^{n} h_i} \quad (2)$$

h_i in Eq.(1) indicates the degree of rule i if the wind speed W_t, the altitude A_t, and the altitude change dA_t are given. $\mu_{WS_i}(W_t)$, $\mu_{ALT_i}(A_t)$, and $\mu_{DALT_i}(dA_t)$ are membership functions corresponding to wind speed, altitude, and altitude change for the rule i, respectively. The membership functions are shown in Fig.3. Each label (WZO, WPS, and so on) corresponds to the fuzzy rule table. φ in Eq.(2) indicates control input given to the winding machine. It is calculated as the weighted sum of the output variable b_i of the rule i with the weight of h_i. $\varphi = 100$ indicates 100 [%] drag force. On the other hand, $\varphi = -100$ indicates the brake force is zero to release the tether line. $\varphi = 0$ indicates the drag force is zero and the brake force is maximum to fix the tether line length.

Table 4 shows the rule table of the fuzzy controller based on 3-inputs, that is, wind speed, altitude, and change of altitude. Each constant number in the table is b_i. If the kite falls down, the controller tends to drag the line to keep the altitude of the kite as higher as possible. If the kite goes higher, the controller reduces dragging force and releases the line if possible.

It tends to become hard to define the rules correctly if the number of inputs or membership functions for the fuzzy rule

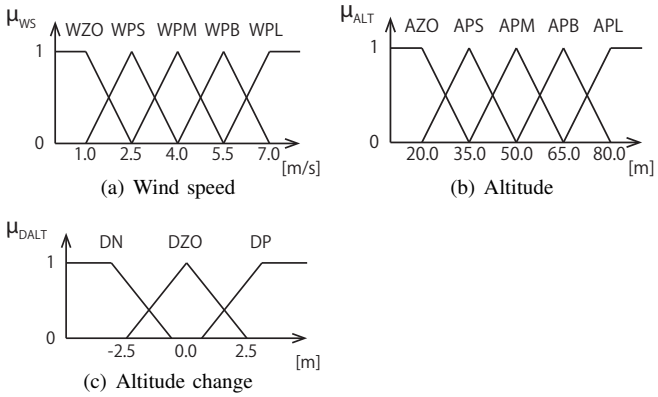


Fig. 3. Antecedent membership function

				Altitude				
				AZO	APS	APM	APB	APL
Wind speed	WZO	Altitude change	DN	100	100	100	100	100
			DZO	100	100	70	30	0
			DP	100	70	30	0	0
	WPS	Altitude change	DN	70	70	70	30	30
			DZO	30	0	0	0	0
			DP	0	0	0	0	0
	WPM	Altitude change	DN	70	70	30	30	30
			DZO	30	0	-30	0	0
			DP	0	-30	-30	0	0
	WPB	Altitude change	DN	70	30	30	30	30
			DZO	-30	-30	-30	0	0
			DP	-70	-70	-70	0	0
	WPL	Altitude change	DN	30	30	30	30	0
			DZO	-70	-70	-70	0	0
			DP	-100	-70	-70	0	0

Fig. 4. FUZZY RULE TABLE (3-inputs 1-output)

table becomes large. As the fuzzy controller is inspired by how a human flies a kite, the learning of the parameters is also based on the human operation data. The human operation data are collected on the computational simulator while a human operates with the simulated kite-based tethered flying robot. The output variable b_i in the fuzzy rule table is updated as follows. Eq.(3) shows the learning equation which we refer learning method based on simplified fuzzy inference model proposed by Ichihashi et al.[10].

$$b_i \leftarrow b_i + \beta h_i(\varphi^* - \varphi) \tag{3}$$

where φ^* and β are the control input of the human operation data and learning rate, respectively. $\beta = 0.1$ in this paper.

The experiments conditions are as follows. The wind in the simulator is generated by sine function with wind speed range from 0.0 to 4.0 [m/s] and 20 [sec] cyclic periods. The line length between the winding machine and the flight unit is set 50 [m] and the flight unit is on the ground when the data collection experiment starts. The goal is flying the robot higher and higher if possible. Figure 5 shows the human operation data used for learning.

Figure 6 shows the state of the tethered flying robot based on the fuzzy controller and the human operation at 100 [sec]. According to Figs.5 and 6, the tether line length of the learned fuzzy controller becomes shorter while it becomes longer in case of human operation. The elevation angle of the learned fuzzy controller is larger than the human operation.

Here, we extend the fuzzy controller from 3-inputs 1-output system to 4-inputs 1-output system in order to solve these problems. This fuzzy controller is also represented based on

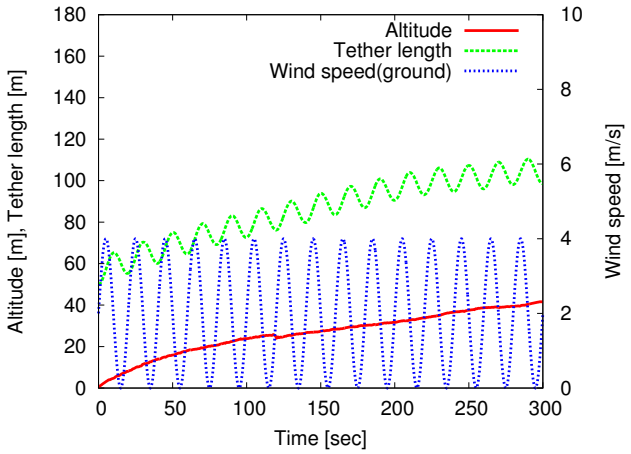


Fig. 5. Human operation (Flight log with human operation on the simulator between the winding machine and the flight unit is 50 [m])

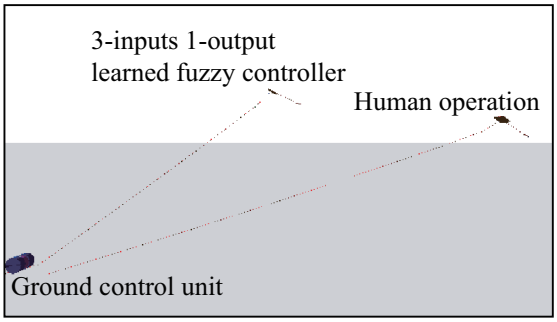


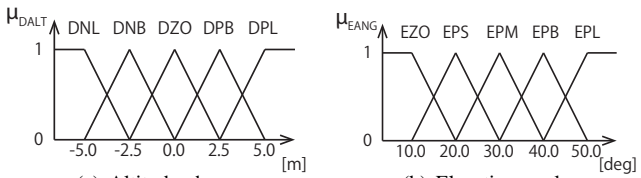
Fig. 6. State of flight based on the fuzzy controller and human operation at 100 [sec]

a simplified reasoning method by Eqs.(4) and (2).

Rule i : if W_t is WS_i and A_t is ALT_i and dA_t is $DALT_i$ and eA_t is $EANG_i$ then φ is b_i $(i = 1, 2, ..., \mathrm{n})$

$$h_i = \min(\mu_{WS_i}(W_t), \mu_{ALT_i}(A_t), \mu_{DALT_i}(dA_t), \mu_{EANG_i}(eA_t)) \tag{4}$$

The elevation angle is added to the inputs and the antecedent member functions are redesigned as shown in Fig.7.



(a) Altitude change (b) Elevation angle

Fig. 7. Antecedent membership functions for 4-inputs 1-output fuzzy controller

We conduct experiments with 4-inputs 1-output learned fuzzy controller under the same simulation conditions. Figure 8 shows the comparison between 3-inputs 1-output and 4-inputs 1-output learned fuzzy controllers. The tether line length of 3-inputs becomes shorter around 50 [sec], while tether line

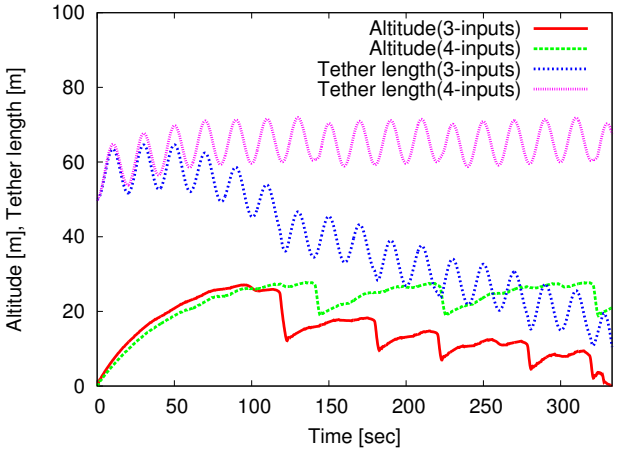


Fig. 8. Comparison between data from 3-inputs 1-output and 4-inputs 1-output learned fuzzy controllers

length of 4-inputs does not become shorter. As a result, the altitude of 4-inputs system does not go down and keep at approximately the same height. The 4-inputs 1-output system has characteristics similar to human operation than 3-inputs 1-output system.

B. Flying Control by K-Nearest Neighbor Algorithm

In k-nearest neighbor algorithm, the control input given to the winding machine of ground control unit is determined by the similarity between data from input and human operated. The similarity is defined by Eq.(5).

$$g_i = \frac{1}{(\sqrt{2\pi})^m \sqrt{|\boldsymbol{\Sigma}|}} \exp(-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_t)^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_t)) \quad (5)$$

where g_i indicates similarity and corresponds to ith human operation data $\boldsymbol{x}_i$. $\boldsymbol{\mu}_t$, $|\boldsymbol{\Sigma}|$, $\boldsymbol{\Sigma}^{-1}$ and T indicate the input informations at time t, determinant of covariance matrix $\boldsymbol{\Sigma}$, inverse matrix of covariance matrix $\boldsymbol{\Sigma}$ and transposition of the vector, respectively. The m indicates the number of data types, so that $m = 4$. The covariance matrix $\boldsymbol{\Sigma}$ of wind speed, altitude, altitude change and elevation angle, has the element of the variance vector, $\boldsymbol{\Sigma} = (\sigma_{ws}^2, \sigma_a^2, \sigma_{da}^2, \sigma_{ea}^2)$ in diagonal, other values are zero. σ_{ws}^2, σ_a^2, σ_{da}^2 and σ_{ea}^2 indicate variance of wind speed, altitude, altitude change, and elevation angle, respectively. The neighbors number is expressed as k, $k = 10$ in this paper. Here, high similarity data are selected from all g_i. Control input φ to the winding machine is calculated by Eq.(6).

$$\varphi = \frac{\sum_{i=1}^{k} g_i O_i}{\sum_{i=1}^{k} g_i} \quad (6)$$

where O_i indicates ith control input operated by human.

C. Flying Control by Artificial Neural Network

The output of the artificial neural network is the control input given to the winding machine of ground control unit. The weights of the artificial neural network are updated with human operation data by back propagation. In this paper, artificial neural network is a feedforward type multilayer perceptron (MLP) which is most commonly used.

The input layer, middle layer and output layer have neurons of four, twenty and one, respectively. Each data is normalized by Eq.(7).

$$x_i = \frac{x_n - x_{min}}{x_{max} - x_{min}} \quad (7)$$

where x_n, x_i, x_{min}, and x_{max} indicate input value before normalizing, minimum of each input of the human operation data, and its maximum, respectively. x_i is one of the values of wind speed, altitude, altitude change and elevation angle. φ is calculated by Eqs.(8) and (9), and through the reverse of z normalized expression based on Eq.(7).

$$y_j = f(\sum_i x_i w_{ij}) \quad (8)$$

$$z = f(\sum_j y_j w_{j1}) \quad (9)$$

where w_{ij} and w_{j1} are the weight between ith unit of input layer and jth unit of middle layer, the weight between jth unit of middle layer and unit of output layer, respectively. The function f indicates sigmoid function is equation as below:

$$f(a) = \frac{1}{1 + exp(-a)} \quad (10)$$

The error function is below:

$$E = \frac{1}{2} \sum_N (z^* - z)^2 \quad (11)$$

where z^* is normalized control input of the human operation. Each weight is updated by back propagation as follows.

$$w \leftarrow w - \gamma \frac{\partial E}{\partial w} \quad (12)$$

where γ is learning rate. ($\gamma = 0.05$ in this paper.) The end conditions of learning is that the error function value is less than 0.1 or the learning count reaches 10000.

IV. Simulation and Results

We prepare two cases of wind situation in the simulator. In the first case, the wind speed is fixed to 2.5 [m/s]. In the second case, the wind speed is varied. The wind is generated by sine function with wind speed range from 0.0 to 4.0 [m/s] at the ground and 20 [sec] cyclic periods. Also generally, it is known that the higher altitude becomes, the stronger wind speed becomes. The wind speed applied to the kite calculates by Eq.(13).

$$W_H = W_R \left(\frac{H}{H_R} \right)^{\frac{1}{n}} \quad (13)$$

where W_H and W_R are wind speed [m/s] at H [m] above the ground and wind speed [m/s] at H_R [m] above the ground, respectively. H_R indicates height of criterion and $H_R = 1.5$ in this paper. We suppose to fly this tethered flying robot in grassland or wilderness, so that $n = 7$ in this paper. The line length between the winding machine of ground control unit and the flight unit is set 100 [m] and the flight unit is on the

ground when the data collection experiments start. The total length of tether line is 300 [m]. Therefore, if the tether line length between the winding machine of ground control unit and the flight unit reaches 300 [m], we finish computational simulations. When a human demonstrator operates this robot in the simulations, two goals are specified as follows.

1) Keeping the altitude of flight unit near 50 [m]
2) Lengthening tether line length between the winding machine of ground control unit and the flight unit as long as possible

Figures 11(a) and 12(a) show the collected human operation data. The altitude, the tether length and the wind speed (ground) of figures indicate the altitude of the flight unit, the length of the tether line between the winding machine of ground control unit, and the wind speed at H_R [m] above the ground, respectively. The human operation data are collected every 0.2 [sec]. Figure 11(a) shows the flight logs on condition that the wind speed is constant. The altitude of the flight unit keeps near 50 [m] and the line length between the winding machine of ground control unit and the flight unit becomes longer. Figure 12(a), on the other hand, shows the flight logs on condition that the wind speed changes by time. The altitude of the flight unit keeps near 50 [m], however, after the altitude attained its height, the line length between the winding machine of ground control unit and the flight unit becomes shorter, unfortunately. It is difficult for human to keep near 50 [m] and lengthen the line between the winding machine of ground control unit and the flight unit on this condition.

A. Simulation based on Learned Fuzzy Rule Table

Figure 9 shows antecedent membership function in these experiments. The fuzzy rule table is written by human in the beginning, then it learns by using human operation data. The number of membership functions of each input value changes 5 to 10 in order to reflect the human operation more precisely.

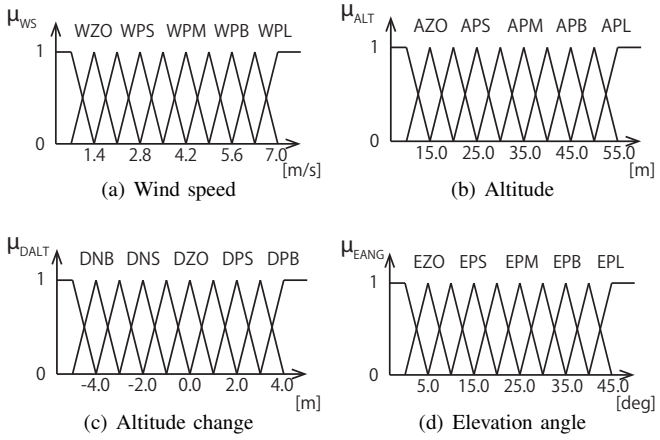


Fig. 9. Antecedent membership function

Figures 11(b) and 12(b) show flight logs when wind speed is constant, and changes by time variation. Figure 11(b) shows similar operation to human operation (Fig.11(a)). If the altitude of the flight unit is much more than 50 [m], the learned fuzzy controller brings the robot down, intentionally. If the robot is less than an altitude of 50 [m], it is also possible to bring the robot up. The robot repeats going up and down near 50 [m] until the line length reaches its limit. Unfortunately, because of line length between the winding machine of ground control unit and the flight unit reaching 300 [m], this computational simulation is finished around 420 [sec]. Transition of tether line length corresponds to human operation. Figure 12(b) also shows similar operation to human operation (Fig.12(a)) in terms of bringing the robot down when it is much more than 50 [m]. However, after bringing the robot down, this controller does not work well. We consider the reason so that the learning count is not enough.

B. Simulation by K-Nearest Neighbor Algorithm

Figures 11(c) and 12(c) show flight logs when wind speed is constant, and changes by time, respectively. Figure 11(c) shows that the operation is similar to human operation (Fig.11(a)). Moreover, this controller is better than human operation because altitude reaches 50 [m] more rapidly than the human operation and altitude is kept 50 [m] around 130 to 180 [sec]. The robot repeats going up and down near 50 [m] until reaching limit of tether line length.

Figure 12(c) shows this controller is also better than human operation (Fig.12(a)). The altitude is reached more rapidly compared to the human operation at 50 [m] and kept at altitude, line length between the winding machine of ground control unit and flight unit is longer than the human operation at around 350 [sec]. When altitude of the flight unit is much more than 50 [m], the learned fuzzy controller is impossible to bring the robot down well in the condition wind speed changes. While, in k-nearest neighbor algorithm, flight logs show similar operation to human operation. In both constant and changing wind speed, k-nearest neighbor algorithm controller is much better than human operation. The reason can be considered as follows: Even if the sensor output is similar, the human operator sometimes gives different control inputs to the winding machine so that the flight unit does not stay stably. The k-nearest neighbor algorithm makes the output smooth so that the flight unit stays stably.

C. Simulation by Artificial Neural network

Figures 11(d) and 12(d) show flight logs when wind speed is constant, and changes by time. Figure 11(d) shows the inability to operate the robot as human operation (Fig.11(a)) except bringing it up. This controller does not acquire the operation which brings the robot down. Figure 12(d) shows this controller brings the robot down intentionally when altitude of flight unit is much more than 50 [m]. However, after bringing the robot down, this controller does not work well, as the learned fuzzy controller does. We consider that this controller fails to reflect human operation characteristics on the grounds that a human does not necessarily enter the same control input to the same or similar situation. In the learning fuzzy rule table, there is antecedent membership function so that altitude

of bring the robot down is expressed. The goal of the altitude is set at near 50 [m], therefore, the altitude of bringing the robot down is not determined clearly. For example, sometimes a human brings the robot up, and sometimes a human brings it down at 53 [m]. Here, we excluded operation data to bring the robot up from the data which the altitude of the flight unit is more than 53 [m]. We conducted learning using the remaining data by artificial neural network and simulated on the condition wind speed is constant.

Figure 10 shows flight logs by learning using improved data. The controller acquires the operation that brings the robot down, and the robot repeats going up and down near 50 [m] until tether line length reaches its limit. From this result, learning of learned data itself is especially important for ambiguous data that has come up with different results in the same or similar situation.

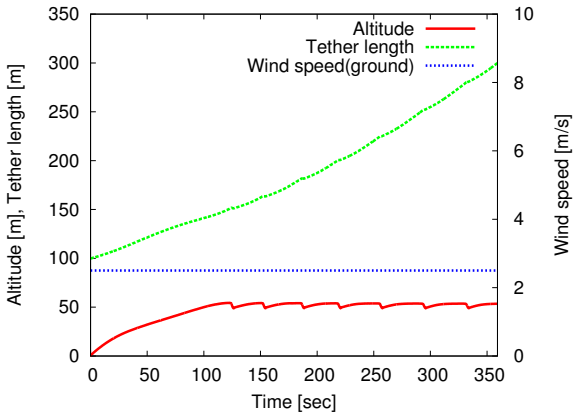


Fig. 10. Constant wind by artificial neural network (The learning data is selected)

V. Conclusion and Future Work

This paper shows human skill acquisition systems to control the kite-based tethered flying robot. The original controller for the kite-based tethered flying robot is designed based on 3-inputs 1-output fuzzy controller in order to imitate human skill of flying a kite. This paper extended controller from 3-inputs 1-output system to 4-inputs 1-output system in order to reflect human operation characteristics. Simulation results showed the validity of the 4-inputs 1-output system. We aimed at reflecting human operation characteristics by learned fuzzy rule table, k-nearest neighbor algorithm, and artificial neural network methods. The k-nearest neighbor algorithm controller showed the best performance to acquire human operation characteristics.

One of the future works is the improving the learning methods of learning data itself in an even better direction. Furthermore, we coordinate a training count, range about antecedent membership function, and so on. In this paper, antecedent membership function is set in a triangular type, however, this type might be not appropriate in the controller of this robot. It is one of our future work to investigate more sophisticated fuzzy logic controllers based on human operation data. We are going to develop a controller for the robot based on these things in order to realize more stable control for the long-term flight. Real robot experiment are also one of the future work.

Acknowledgment

This work was partially supported by JSPS KAKENHI Grant Number 24650118.

References

[1] T. Ishii, Y. Takahashi, Y. Maeda, and T. Nakamura, "Tethered flying robot for information gathering system," in *IROS'13 Workshop on Robots and Sensors integration in future rescue INformation system (ROSIN'13)*, November, 2013.

[2] C. Todoroki, Y. Takahashi, and T. Nakamura, "Learning fuzzy control parameters for kite-based tethered flying robot using human operation data," in *Proceedings of Joint 7th International Conference on Soft Computing and Intelligent Systems and 15th International Symposium on Advanced Intelligent Systems (SCIS & ISIS 2014)*, December 2014, pp. 111–116.

[3] H. Aichi, M. Kumada, T. Kobayashi, Y. Sugiyama, K. Horii, and S. Yokoi, "Application to power generation by wind and environmental watching and/or it system by floating platform tethered with cable, and preliminary study on climbing characteristics of a tethered aerodynamic platform with side-by-side twin rotors," *Bulletin of Daido Institute of Technology*, vol. 38, pp. 213–223, 2002.

[4] ——, "Application of floating platform tethered with cable in high altitude to wind power generation, telecommnication / broadcasting, and environmental watching," *Denkiseiko*, vol. 74, no. 3, pp. 167–172, 2003.

[5] S. Karim and C. Heinze, "Experiences with the design and implementation of an agent-based autonomous uav controller," in *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*. New York, NY, USA: ACM, 2005, pp. 19–26.

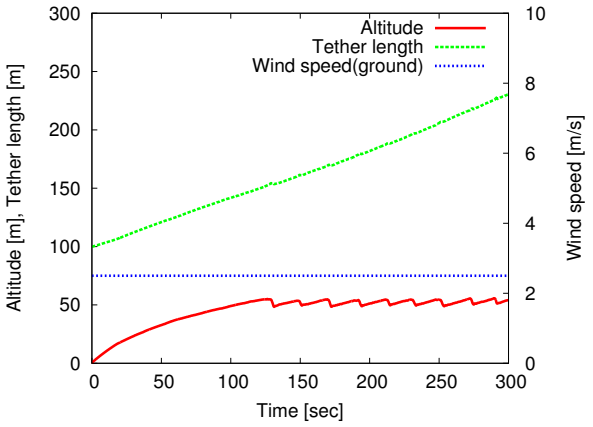
[6] J. Fujinaga, H. Tokutake, and S. Sunada, "Guidance and control of a small unmanned aerial vehicle and autonomous flight experiments," *Journal of the Japan Society for Aeronautical and Space Sciences*, vol. 56, no. 649, pp. 57–64, 2008-02-05. [Online]. Available: http://ci.nii.ac.jp/naid/10021152872/

[7] T. Suzuki, J. Meguro, Y. Amano, T. Hashizume, D. Kubo, T. Tsuchiya, S. Suzuki, R. Hirose, K. Tatsumi, K. Sato, and J. Takiguchi, "Development of information collecting system using a small unmanned aerial vehicle for disaster prevention and mitigation," *Journal of Robotics Society of Japan*, vol. 26, no. 6, pp. 553–560, 2008-08-29. [Online]. Available: http://ci.nii.ac.jp/naid/10024269382/

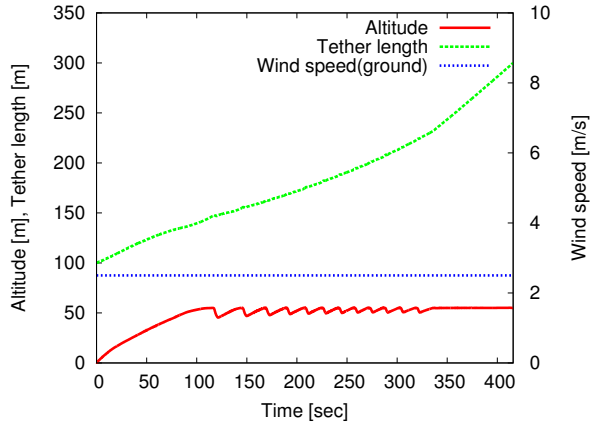
[8] T. Ishii, Y. Takahashi, Y. Maeda, and T. Nakamura, "Fuzzy control for kite-based tethered flying robot," in *Proceedings of 2014 IEEE World Congress on Computational Intelligence*, vol. DVD-ROM, July 2014, pp. 746–751.

[9] R. Smith, "Open dynamics engine," 2008, http://www.ode.org/. [Online]. Available: http://www.ode.org/

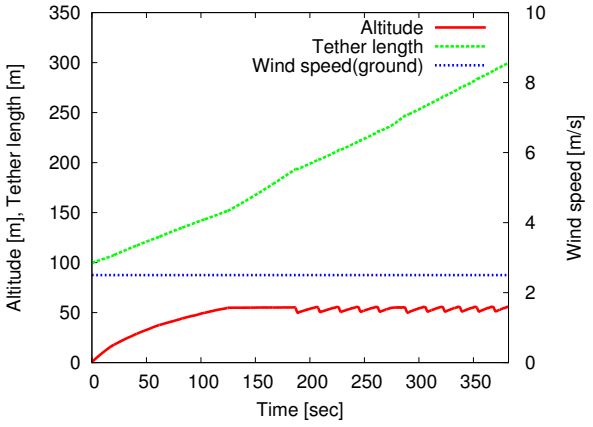
[10] H. Ichihashi and T. Watanabe, "Learning control by fuzzy models using a simplified fuzzy reasoning," *Japan Society for Fuzzy Theory and Systems*, vol. 2, no. 3, pp. 429–437, August 15, 1990.



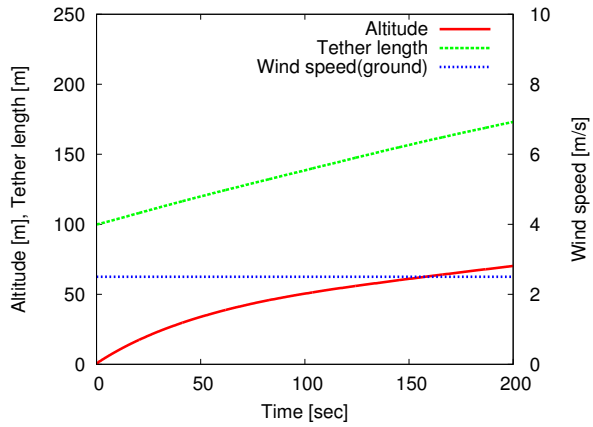
(a) Human operation



(b) Learned fuzzy table

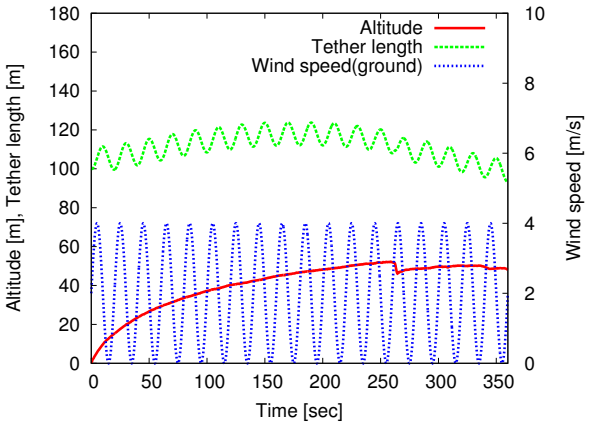


(c) K-nearest neighbor algorithm

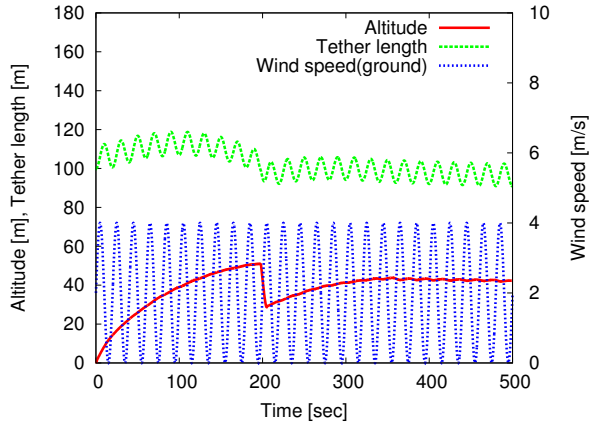


(d) Artificial neural network

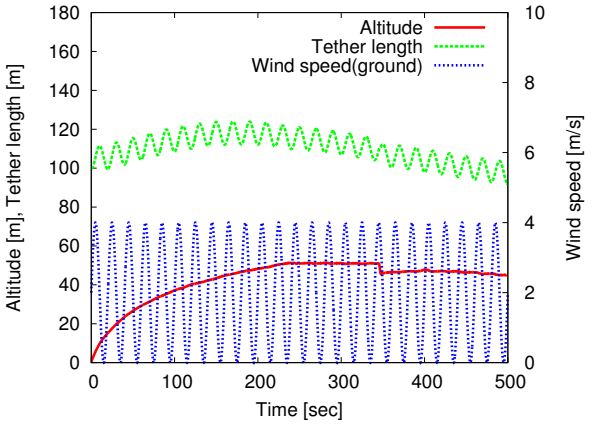
Fig. 11. Constant wind



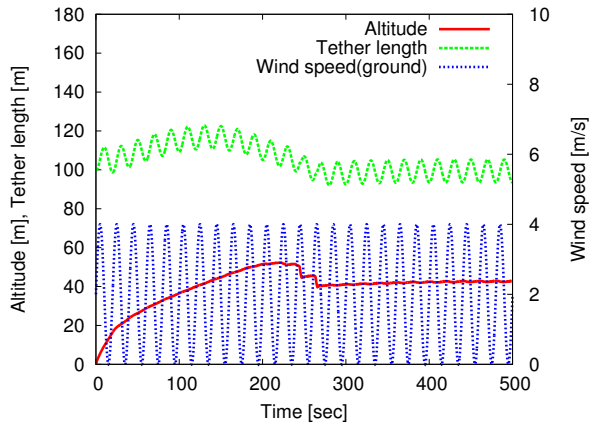
(a) Human operation



(b) Learned fuzzy table



(c) K-nearest neighbor algorithm



(d) Artificial neural network

Fig. 12. Wind with speed change