Intelligent Attitude Control of an Unmanned Helicopter

Seongjun An, Bumjin Park, and Jinyoung Suk
Department of Aerospace Engineering, Chungnam National University, Daejeon, Korea
(Tel : +82-42-825-9225; E-mail: jsuk@cnu.ac.kr)

Abstract: This paper presents a new attitude stabilization and control of an unmanned helicopter based on neural network compensation. A systematic derivation on the dynamics of an unmanned small-scale helicopter is performed. Combined rotor-fuselage-tail dynamics is derived in body-fixed reference frame with its origin at the C.G. of the helicopter. And the resulting nonlinear equation of motion consists of 6-DOF air vehicle dynamics as well as the rotor flapping and engine torque equations. A simulation model was modified using the existing simulator for an unmanned helicopter dynamic model, which reflects the unmanned test helicopter(CNUHELI). The dynamic response of the refined model was compared with the flight test data. It can be shown that a good coincidence was accomplished between the real unmanned helicopter system and the mathematical model. This dynamic model was linearized for classical controller design using small perturbation method. A Neuro-PD control system was designed for both longitudinal and lateral flight modes, and the results were compared with the PD-only control response. Simulation results show that the proposed Neuro-PD control system demonstrates better performance.

Keywords: unmanned helicopter, neural network, autonomous, attitude control, nonlinearity

1. INTRODUCTION

Unmanned helicopters are emerging as an alternative to the conventional fixed-wing UAVs(Unmanned Aerial Vehicles) due to its superior flight characteristics: vertical take-off and landing in addition to the hovering capability. Various applications for unmanned helicopters such as Firescout and R-max include aerial photography, agricultural use as well as surveillance and reconnaissance missions. Domestic and world market revenue for this type of UAVs is growing in both military and commercial field, which activates a growing research momentum in various universities shown in Fig. 1[1]. However, development of a fully autonomous unmanned helicopter requires powerful flight control system because the airframe has higher order nonlinear dynamic characteristics caused by the main rotor, fuselage and tail rotor. This complex dynamics should be overcome by the control system, which accommodates various kind of nonlinearity and atmospheric disturbance. In this paper, a blended classical control theory and neural network compensation is studied for the attitude control of an unmanned helicopter. In recent years, classical control theory has been successfully applied to general class of fixed-wing UAVs due to its simple and efficient structure. It also pertain good physical meaning with respect to flight dynamics. However, it is not an easy job to apply the classical control theory to the helicopter directly, and there should be a certain type of additional compensation. It is also well known that the neural network control can be effectively applied to a variety of highly nonlinear systems[2-3].

Based on the assumption that the main rotor inflow is steady and uniform, an iterative method was adopted to calculate the thrust coefficients of the rotor in order to derive the dynamic equations of motion of an unmanned subscale helicopter. Flapping dynamics is represented as a Fourier series on the blade azimuth angle. The deflection of swash plate causes the change of cyclic pitch angle resulting in the coupled second order differential equations. The overall equations are derived on the body coordinate frame.

In this paper, a Neuro-PD control system was designed for both longitudinal and lateral flight modes. Control gains of a classical attitude stabilizer for an unmanned helicopter are updated each time. Therefore, outputs of the neural network are proportional, derivative gains that feedback the attitude errors in both longitudinal and lateral motion. A neural network emulator was used to provide Jacobian, which was tuned online after pre-learning.

Numerical simulations were demonstrated to validate the performance of the proposed Neuro-PD controller for the identified system model. The weights of the neural network converge within a second after some transients. The result also verified that the gains of the designed classical control system can be tuned in one step using the neural network compensation. The computational time for implementation of this algorithm is allowable so that it can be used for future flight test.

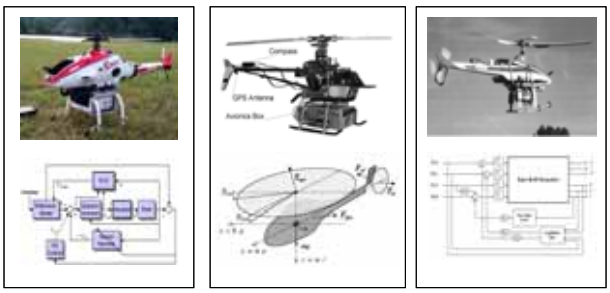

Fig. 1 Research Activities on Unmanned Helicopter in Universities

2. EQUATION OF MOTION FOR AN UNMANNED HELICOPTER

Dynamic modeling of large helicopters that have articulated rotor blades are highly complicated since each blade has

individual degree of freedom. However, the equation can be simplified for small unmanned helicopters if we consider only the dynamics of rotor system with respect to the airframe. The speed of rotor is assumed constant and the coordinate system can be shown in Fig. 2. Under this assumption integrated dynamics of the unmanned helicopter can be expressed as Eqs. (1)~(12)[4].

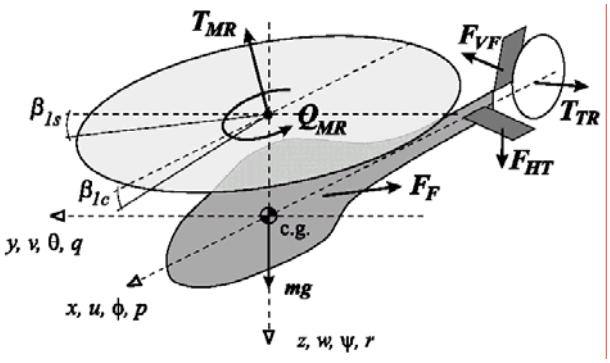

Fig. 2 Moments and force acting on helicopter

$$\dot{v} = wp - ur + g\sin\phi\cos\theta + (Y_{mr} + Y_{fus} + Y_{tr} + Y_{vf})/m \tag{1}$$

$$\dot{w} = uq - vp + g\cos\phi\cos\theta + (Z_{mr} + Z_{ht})/m \tag{2}$$

$$\dot{p} = qr(I_{zz} - I_{yy})/I_{xx} + (L_{mr} + L_{vf} + L_{tr})/I_{xx} \tag{3}$$

$$\dot{q} = pr(I_{xx} - I_{zz})/I_{yy} + (M_{mr} + M_{ht})/I_{yy} \tag{4}$$

$$\dot{r} = pq(I_{yy} - I_{xx})/I_{zz} + (-Q_e + N_{vf} + N_{tr})/I_{zz} \tag{5}$$

$$\dot{\phi} = p + \tan\theta(q\sin\phi + R\cos\phi) \tag{6}$$

$$\dot{\theta} = q\cos\phi - r\sin\phi \tag{7}$$

$$\dot{\psi} = (q\sin\phi + r\cos\phi)/\cos\theta \tag{8}$$

$$\dot{b}_1 = -p - \frac{b_1}{\tau_e} - \frac{1}{\tau_e}\frac{\partial b_1}{\partial \mu_v}\frac{v - v_w}{\Omega R} + \frac{B_{\delta_{lat}}}{\tau_e}\delta_{lat} \tag{9}$$

$$\dot{a}_1 = -q - \frac{a_1}{\tau_e} + \frac{1}{\tau_e}\left(\frac{\partial a_1}{\partial \mu}\frac{u - u_w}{\Omega R} + \frac{\partial a_1}{\partial \mu_z}\frac{w - w_w}{\Omega R}\right) + \frac{A_{\delta_{lon}}}{\tau_e}\delta_{lon} \tag{10}$$

$$\dot{\Omega} = -\dot{r} + \frac{1}{I_{rot}}[Q_e - Q_{mr} - n_{tr}Q_{tr}] \tag{11}$$

$$\dot{Q}_e = -\frac{1}{\tau_{eng}}Q_e + \frac{1}{\tau_{eng}}\left(Q_e^0 + K_{gov}(\Omega_c - \Omega)\right) \tag{12}$$

Aerodynamic loads generated by the rotor can be derived in the hub-wind coordinate system, and then coordinate-transformed into the hub-mast frame. the aerodynamic components on hub-mast axes are again transformed into the body frame, and they consist the force and moment for the construction of the whole equation of motion. Flapping dynamics of the main rotor and aerodynamics of the main/tail rotors are calculated on hub-wind frame. Similarly, aerodynamic components of the fuselage and stabilizer are added. Thrust and torque are calculated on the wind-axes, and they are converted into the body frame. From MBC(Multi-Blades Coordinates), the longitudinal and lateral flap equations of blades can be represented as [5]

$$\beta'' + C\beta' + K\beta = F \tag{13}$$

where ( )' with respect to an azimuth angle is defined as

$$(\,)' = \frac{d}{d\psi} = \frac{1}{\Omega}\frac{d}{dt} \tag{14}$$

Damping matrix, stiffness matrix and load vector which compose flap equations can be written as

$$C = \frac{\gamma}{8}\begin{bmatrix} 1 & \frac{16}{\gamma} \\ -\frac{16}{\gamma} & 1 \end{bmatrix} \tag{15}$$

$$K = \frac{\gamma}{8}\begin{bmatrix} \frac{8(\lambda_\beta^2 - 1)}{\gamma} & 1 + \frac{\mu^2}{2} \\ -(1 - \frac{\mu^2}{2}) & \frac{8(\lambda_\beta^2 - 1)}{\gamma} \end{bmatrix} \tag{16}$$

$$F = \frac{\gamma}{8}\begin{bmatrix} \frac{16}{\gamma}(\bar{p}_{hw} + \frac{\bar{q}_{hw}}{2}) + \theta_{1cw}(1 + \frac{\mu^2}{2}) + (\bar{q}_{hw} - \lambda_{1cw}) \\ -\frac{16}{\gamma}(\bar{q}_{hw} - \frac{\bar{p}_{hw}}{2}) + \frac{8}{3}\mu\theta_0 + 2\mu\theta_{tw} + \theta_{1sw}(1 + \frac{3}{2}\mu^2) + 2\mu(\mu_z - \lambda_0) + (\bar{p}_{hw} - \lambda_{1sw}) \end{bmatrix} \tag{17}$$

where $()_{hw}$ means the hub-wind axes, θ_0 is the collective pitch angle, θ_{1cw} is the longitudinal cyclic pitch angle, θ_{1sw} is the lateral cyclic pitch angle and θ_{tw} is the blade twist angle. μ is the advance ratio and μ_z is the vertical component of normalized velocity. λ_0 is the uniform inflow component, λ_{1cw} and λ_{1sw} are the first harmonic inflow components. The normalized angular velocity components are written as

$$\bar{p}_{hw} = \frac{p_{hw}}{\Omega}, \quad \bar{q}_{hw} = \frac{q_{hw}}{\Omega} \tag{18}$$

The flap equations of conventional helicopters are reduced by the following assumption for RC helicopters.

a. The inflow is uniform
b. Blades have no twist.
c. Flap equations are approximated to 1st order derivative equations.
d. The rotor speed is constant.
e. Higher order terms are ignored.

In general, it is shown that the stabilizer bar of the RC helicopter augments rotor control inputs and flap equations of the stabilizer bar can be derived from simplified flap equations

of blades as

$$\bar{C}\dot{\bar{\beta}} = -\Omega(\bar{K}\bar{\beta} + \bar{F}) \tag{19}$$

Using the stabilizer bar effect, flap equations of blades can be written as

$$C\dot{\beta} = -\Omega(K\beta + F + K_s\bar{\beta}) \tag{20}$$

where K_s is the flap response gain of the stabilizer bar.

3. A NEURO-PD ATTITUDE CONTROL

In this paper, a combined classical PD controller and neural network compensation are used for the attitude compensation of the unmanned helicopter. Back propagation algorithm is used to update the weight of the multi-layer neural network. A single hidden layer is used and sigmoid function is used for each node. It is well known that the neural network is based on the simulation of the human brain consisting of neuron and connecting structure. Learning is a major process of updating weights and biases at each step. Automatic flight control system of high-performance aircraft, flight path simulation, and performance enhancement of the automatic flight are three major applications for aeronautical use of the intelligent neural network based control.

3.1 Back Propagation Algorithm

Back propagation algorithm is a kind of learning method for multilayered perception that updates the weights of each layer. The weights are to be determined to minimize the error signal and performance measure on the basis of the delta-rule and gradent decent method. Therefore, this algorithm can be applied on condition that the nonlinear neural network function can be differentiable. Let the input layer, hidden layer, output layer, number of inputs, number of hidden layers and number of neurons on output layer be I, j, k, N_I, N_B and N_O, respectively. Then, the output can be represented as follows[6]:

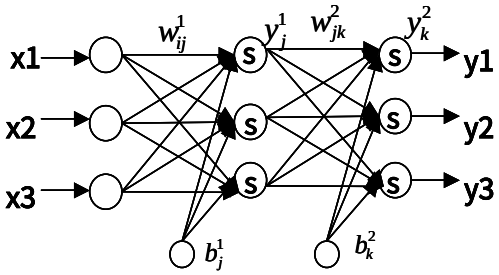

Fig. 3 Three-layered Neural Network

$$y_k = \frac{1}{1+e^{-\sum_{j=0}^{N_H} w_{jk}\left[\frac{1}{1-e^{-\sum_{i=0}^{N_I} w_{ij}x_i + b_j}}\right] + b_k}} \tag{21}$$

Output value can be expressed by the weights and biases of the neural network as can be seen in Eq.(21). An error between the output from the neural network and the desired output is a direct difference:

$$e_k = y_{dk} - y_k \tag{22}$$

The performance index to minimize the generated error can be shown below

$$E = \frac{1}{2}\sum_k e_k^2 \tag{23}$$

Partial differentiation of the above performance index with respect to time produces the gradient on the weights of the error, which means the direction of change that minimizes the error in the next step. Error change rate of the output layer can be determined by the delta-rule as follows:

$$\Delta w_{jk} = -\eta\frac{\partial E}{\partial w_{jk}} \tag{24}$$

where η denotes learning rate. Now the gradient can be expressed as follows to minimize the performance index.

$$\begin{aligned}\frac{\partial E}{\partial w_{jk}} &= \frac{\partial E}{\partial e_k}\frac{\partial e_k}{\partial y_k}\frac{\partial y_k}{\partial w_{jk}} \\ &= \left(\frac{1}{2}\frac{\partial e_k^2}{\partial e_k}\right)\left(\frac{\partial (y_{dk} - y_k)}{\partial y_k}\right)\left(\frac{\partial y_k}{\partial w_{jk}}\right) \\ &= (e_k)(-1)\left(\frac{\partial y_k}{\partial w_{jk}}\right) \\ &= -e_k\frac{\partial y_k}{\partial w_{jk}}\end{aligned} \tag{25}$$

where y_k is a function of s_k, and s_k is a function of w_{jk}. From this a chain rule can be applied to both Eqs.(26)~(27).

$$\begin{aligned}\Delta w_{jk} &= -\eta\left(-e_k\frac{\partial y_k}{\partial w_{jk}}\right) = \eta e_k\frac{\partial y_k}{\partial s_k}\frac{\partial s_k}{\partial w_{jk}} = \eta e_k f'(s_k)\frac{\partial s_k}{\partial w_{jk}} \\ &= \eta e_k f'(s_k)\frac{\partial\left(\sum_{j=1}^{N_H} w_{jk}y_j + b_k\right)}{\partial w_{jk}} \\ &= \eta e_k f'(s_k)y_j\end{aligned} \tag{26}$$

Eq.(27) can be expressed using delta

$$\Delta w = \eta\delta_k y_j,\ \ \delta_k = e_k f'(s_k) \tag{27}$$

Weights on the hidden layer can be expressed using the similar way. Change rate on each weight can be expressed as follows:

$$\Delta w_{jk} = \eta\, e_k f'(s_k)y_j \tag{28}$$

$$\Delta w_{ij} = \eta\, f'(s_j)x_i\sum_{k=1}^{N_0} e_k f'(s_k)w_{jk} \tag{29}$$

Or it can be expressed using the delta-rule

$$\begin{aligned}\Delta w_{jk} &= \eta\delta_k y_j \\ \delta_k &= e_k f'(s_k)\end{aligned} \tag{30}$$

$$\begin{aligned}\Delta w_{ij} &= \eta\delta_j x_i \\ \delta_j &= \sum_{k=1}^{N_0}\delta_k w_{jk} f'(s_j)\end{aligned} \tag{31}$$

Sum of weights on the output layer is included in the delta on hidden layer. This means that the opposite sides of the neural

network, and the error on the hidden layer can be expressed as a sum of the weights on the output layer. As a result, the weights can be updated if we know the deltas on both output layer and hidden layer, hence we can update the neural network that have multi layers and differentiable node functions.

Biases on output and hidden layer can be expressed using the delta-rule.

$$\Delta b_k = \eta \delta_k \qquad \delta_k = e_k f'(s_k) \tag{32}$$

$$\Delta b_j = \eta f'(s_j) \sum_{k=1}^{N_0} e_k f'(s_k) w_{jk} \tag{33}$$

The followings are the update rule for both weights and biases at each simulation step.

$$\begin{aligned} w_{jk}(t+1) &= w_{jk}(t) + \Delta w_{jk} \\ w_{ij}(t+1) &= w_{ij}(t) + \Delta w_{ij} \\ b_j(t+1) &= b_j(t) + \Delta b_j \\ b_k(t+1) &= b_k(t) + \Delta b_k \end{aligned} \tag{34}$$

3.2 Neuro-PD Control

Disadvantage of the classical control system is that it is almost impossible to obtain the desired output for highly nonlinear and complicated system such as unmanned helicopters, nor it can give good performance in the event of severe external disturbance of variation of internal parameters. Neuro-PD control compensates the PD control gains at each sampling step. Two methods are used for Neuro-PD control: with and without an emulator. A neural network emulator provides the Jacobian information of the system to learn the network in the former one. Two neural networks are used: the emulator is learned off-line, and the neural network is tuned on-line. In this case, a convergence rate of each neural network is of critical issue. When the convergence rate is different, the neural network emulator may provide inaccurate Jacobian information. In order to overcome this, a modified method is used in this paper, which is shown in Fig. 4.

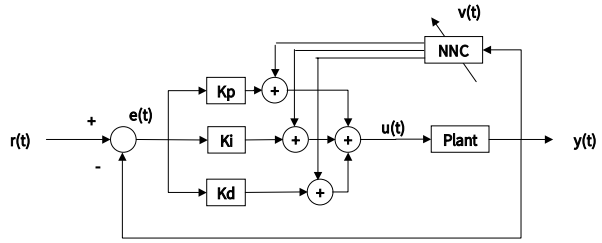

Fig.4 Neural PID Controller without Jacobian

A Learning signal $v(t)$ can be assigned

$$v(t) = K_p e(t) + K_I \int e(t)dt + K_D \dot{e}(t) \tag{35}$$

Control input $u(t)$ is a sum of the output of the neural network.

$$\begin{aligned} v(t) &= u(t-1) + (O_1 + K_P)(e(t) - e(t-1)) \\ &\quad + (O_2 + K_1)e(t) \\ &\quad + (O_3 + K_D)(e(t) - 2e(t-1) + e(t-2)) \\ &= u(t-1) + O_1(e(t) - e(t-1) + O_2 e(t) \\ &\quad + O_3(e(t) - 2e(t-1) + e(t-2) + v \end{aligned} \tag{36}$$

Combining the control input with the dynamics of the system, we can get

$$\begin{aligned} u(t) &= (K_P + O_1)e(t) + (K_1 + O_2)\int e(t)dt + (K_D + O_3)\dot{e}(t) \\ &= O_1 e(t) + O_2 \int e(t)dt + O_3 \dot{e}(t) + v \end{aligned} \tag{37}$$

Eq. (37) can be rearranged as

$$v = u(t) - (O_1 e(t) + O_2 \int e(t)dt + O_3 \dot{e}(t)) \tag{38}$$

Learning signal $v(t)$ affects the learning algorithm in the neural network. It is defined as an error in the output layer. The performance index can be expressed as follows:

$$E = \frac{1}{2} v^2 \tag{39}$$

4. MODEL IDENTIFICATION AND SIMULATION

Dynamic modeling of the CNUHELI unmanned helicopter developed in Chungnam national university was performed using the existing unmanned helicopter modeling data. The modification was based on a laboratory-based ground test. Configuration parameters of the CNUHELI were summarized in Table 1. Several experiments were conducted to extract the moment of inertia of the unmanned helicopter. An alternative geographic modeling was conducted using CATIA, and the result was used in the simulation.

Table 1 Configuration Parameters of CNUHELI

Mass	10.0 (kg)
Rolling moment of inertia	0.27 (kg*m^2)
Pitching moment of inertia	0.50 (kg*m^2)
Yawing moment of inertia	0.45 (kg*m^2)
M.R. radius	0.94 (m)
M.R. chord	0.063 (m)
M.R. hub height above c.g.	0.305 (m)
T.R. radius	0.187 (m)
T.R. chord	0.027 (m)
T.R. height above c.g.	0.065 (m)
T.R. hub location behind c.g.	0.91 (m)
Gear ration T.R. to M.R.	5.7
Effective vertical fin area	0.0141 (m^2)
Horizontal fin area	0.0148 (m^2)

Control stick command is directly interfaced to the simulation program so that the manual flight can be done online. The nonlinear dynamics of the CNUHELI was linearized to design a reference PD controller. A numerical perturbation was used to linearize the model. The linearized model was validated by comparing both responses in a finite time after actuating the model in hover condition. The linearized models for both longitudinal and lateral axes are shown below.

$$\begin{bmatrix}\dot{u}\\ \dot{w}\\ \dot{q}\\ \dot{\theta}\\ \dot{a}_1\\ \dot{\Omega}\end{bmatrix}=\begin{bmatrix}-0.0284 & -0.0169 & 0.0094 & -9.8098 & -9.6010 & -0.0006\\ -0.3553 & -1.1369 & 1.4921 & 0.0622 & 0.0000 & -0.1421\\ 0.0588 & -0.0150 & -0.0658 & 0.0000 & 166.5657 & -0.0026\\ 0.0000 & 0.0000 & 0.9918 & 0.0000 & 0.0000 & 0.0000\\ 0.0014 & 0.0001 & -1.0000 & 0.0000 & -8.3500 & 0.0002\\ -0.7511 & -0.2336 & 0.7637 & 0.0000 & 0.0000 & -2.2027\end{bmatrix}\begin{bmatrix}u\\ w\\ q\\ \theta\\ a_1\\ \Omega\end{bmatrix}+\begin{bmatrix}-0.8555 & 0.0000\\ -19.9378 & 0.0000\\ 6.4664 & 0.0000\\ 0.0000 & 0.0000\\ 0.0340 & -35.0700\\ -606.0358 & 0.0000\end{bmatrix}\begin{bmatrix}\delta_{collective}\\ \delta_{lon}\end{bmatrix}$$

$$\begin{bmatrix}\dot{v}\\ \dot{p}\\ \dot{r}\\ \dot{\phi}\\ \dot{\psi}\\ \dot{b}_1\\ \dot{\Omega}\end{bmatrix}=\begin{bmatrix}-0.1410 & -0.0178 & -1.4255 & 9.7300 & 0.0000 & 9.6012 & -0.0167\\ -0.2049 & -0.0187 & 0.1820 & 0.0000 & 0.0000 & 308.4570 & -0.0271\\ 1.7141 & 0.1793 & -1.5310 & 0.0000 & 0.0000 & 0.0000 & -0.0762\\ 0.0000 & 1.0000 & -0.0062 & 0.0000 & 0.0000 & 0.0000 & 0.0000\\ 0.0000 & 0.0000 & 0.9918 & 0.0000 & 0.0000 & 0.0000 & 0.0000\\ 0.0014 & -1.0000 & 0.0000 & 0.0000 & 0.0000 & -8.3500 & 0.0011\\ -1.7959 & -0.1793 & 1.5895 & 0.0000 & 0.0000 & 0.0000 & -2.2027\end{bmatrix}\begin{bmatrix}v\\ p\\ r\\ \phi\\ \psi\\ b_1\\ \Omega\end{bmatrix}+\begin{bmatrix}0.0000 & -185.807\\ 0.0000 & -447.355\\ 0.0000 & 375.7562\\ 0.0000 & 0.0000\\ 0.0000 & 0.0000\\ 350.700 & 0.0000\\ 0.0000 & -495.5084\end{bmatrix}\begin{bmatrix}\delta_{lat}\\ \delta_{rud}\end{bmatrix}$$

Dynamic simulation was performed for the linear model, using the simulink block diagram. A doublet command was applied as a reference input. A very sensitive pitch response was obtained for both longitudinal and lateral axis. Overall system response can be seen in Figs. 5-6.

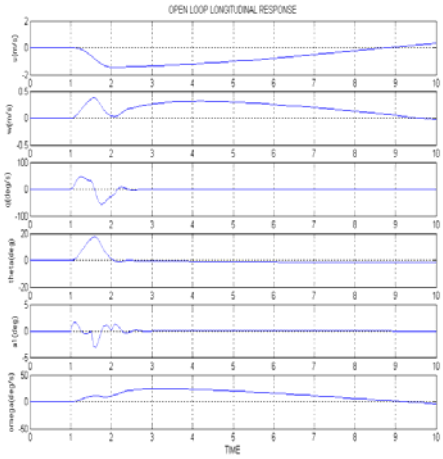

Fig.5 Longitudinal Response

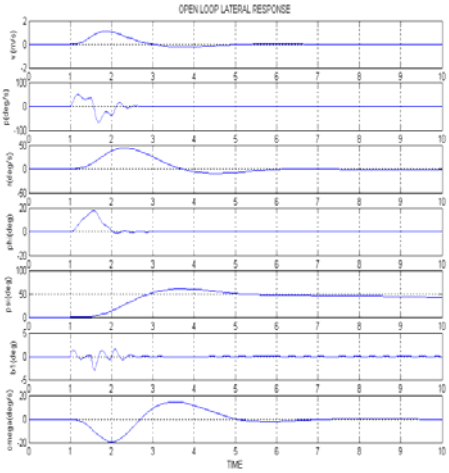

Fig.6 Lateral Response

Fight model was identified to better the model matching between real system and mathematical modeling. Model parameters were revised to give more reasonable response with respect to flight test data. Manual flight data were obtained by the flight test. Pitch response of the linear model shows up to 5 times larger than the flight test data when the revision was made only on geometric data. A further investigation on the flapping dynamics revealed that the input influence parameter should also be identified to extract relevant data. Flapping dynamics can be expressed using the pitch rate, rotor speed and longitudinal excitation.

$$\dot{a}_1 = -q - \frac{a_1}{\tau_e} + \frac{1}{\tau_e}\left(\frac{\partial a_1}{\partial \mu}\frac{u-u_w}{\Omega R} + \frac{\partial a_1}{\partial \mu_z}\frac{w-w_w}{\Omega R}\right) + \frac{A_{\delta_{lon}}}{\tau_e}\delta_{lon} \tag{40}$$

$$\dot{b}_1 = -p - \frac{b_1}{\tau_e} - \frac{1}{\tau_e}\frac{\partial b_1}{\partial \mu_v}\frac{v-v_w}{\Omega R} + \frac{B_{\delta_{lat}}}{\tau_e}\delta_{lat} \tag{41}$$

$A_{\delta_{lon}}$: effective steady-state longitudinal gain from the cyclic input to mainrotor flap angle

$B_{\delta_{lat}}$: effective steady-state lateral gain from the cyclic input to main rotor flap angle

Flight test data and identified model response were shown in Fig. 7, where we can see similar response for both pitch and roll motion.

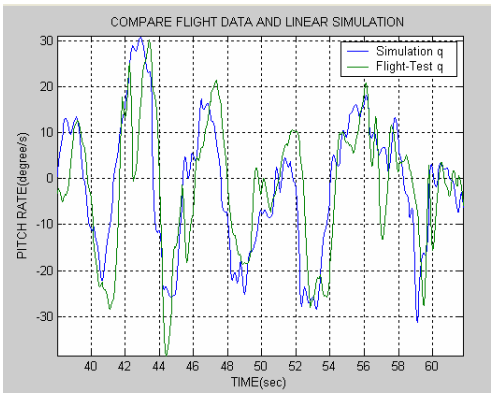

Fig.7(a) Linear Model and Flight Data Comparison

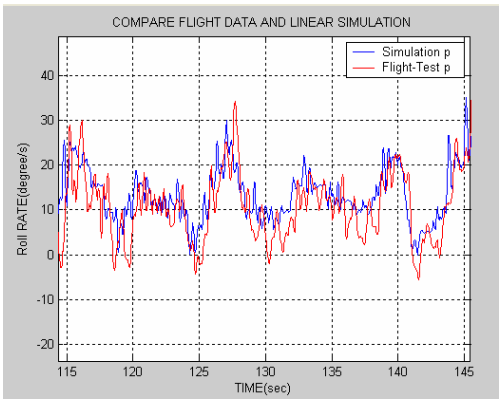

Fig.7(b) Linear Model and Flight Data Comparison

A classical control was applied at first, and an additional compensation was activated using the neural network designed in Sec. 3. 2 input variables and 10 hidden layer parameters with 2 output variables were used to compensate the PD control. Fig. 8 shows the overall longitudinal control scheme

for the attitude stabilization of the unmanned helicopter.

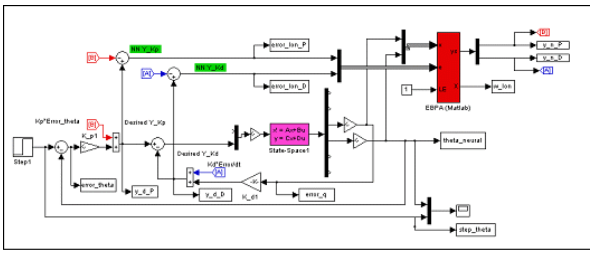

Fig. 8 Longitudinal Neuro-PD Controller Block

The performance of the proposed Neuro-PD control system was compared with the conventional PID control. The same values on proportional and derivative gains were used for 5 second duration of numerical simulation. The error between the output of the neural network and flight control command was selected as learning signal. Comparative simulation results for longitudinal and lateral cyclic input were shown in Fig. 9.

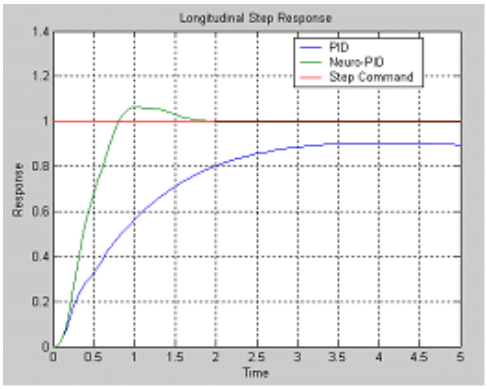

Fig.9(a) Longitudinal Neuro-PD and PD Comparison

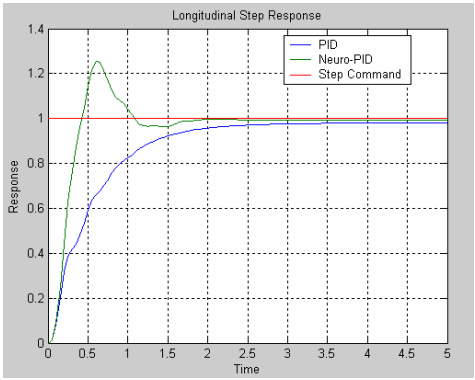

Fig. 9(b) Lateral Neuro-PD and PD Comparison

5. Conclusions

In this paper, a nonlinear simulation was performed based on the helicopter dynamic model. An attitude stabilization and control was implemented using the Neuro-PD control based on the identified linear model. Simulation results show that the neural network compensation can be used effectively to enhance the control performance of the unmanned helicopter. It is expected that the flight control performance of the existing automatic flight control system can be upgraded by just adding the neural network compensation. Future flight test on the robustness with respect to severe atmospheric disturbance is required for further validation of the proposed Neuro-PD control. An automatic take-off and landing experiment can be expected using the precise attitude stabilization. A more accurate and systematic model identification method is also a future research. All these efforts will expand the flight control regime on unmanned helicopters.

REFERENCES

[1] Bernard Mettler, "Identification Modeling and Characteristics of Miniature Rotorcraft", Kluwer Academic Publishers, 2003.

[2] J.Leitner, A.J.Calise and J.V.R.Prasad, "Analysis of Adaptive Neural Networks for Helicopter Flight Controls," AIAA Journal of Guidance, Control, and Dynamics, Vol. 20, No. 5, Sept.-Oct., pp.972-979, 1997.

[3] Eric N. Johnson and Suresh K. Kannan, “Adaptive Flight Control for an Autonomous Unmanned Helicopter,” In AIAA Guidance, Navigation and Control Conference, number AIAA-2002-4439, Monterey, CA, August 2002

[4] V. Gavrilets, “Dynamic Model for X-Cell Helicopter in Low Advanced Ratio Flight”, MIT, 2002

[5] Gareth D. Padfield, "Helicopter Flight Dynamics: The Theory and Application of Flying Qualities and Simulation Modelling," Blackwell Science, 1996

[6] S. Jeong, *Artificial Intelligence System*, Chungnam National University, 2003.