American Journal of Intelligent Systems 2015, 5(1): 18-26
DOI: 10.5923/j.ajis.20150501.02

Position Difference for System Identification and Control of UAV Alap-Alap Using Back Propagation Algorithm Neural Network with Kalman Filter

Afrias Sarotama*, Benyamin Kusumoputro

Department of Electrical Engineering, University of Indonesia, Depok Indonesia

Abstract To derive System Identification and Control of dynamic MIMO UAV nonlinear system, based on the collection of input-output data during sampled from a test flights, using artificial neural network is more convenient compared to physics and mathematics methods. The data is used as both training and testing set for artificial neural networks. There were 36250 input-output sampled flight data and grouped into two flight data sets. The first flight data set, a chirp signal, are used for training the neural network to determine parameters (weights) for the network, using all sample flight which are not belong to the second data set. Validation of the network is performed using the second data set, which were not used for training, which are representation of UAV circular flight movement. After an artificial neural network was trained using the training data set, the network is excited by the second set input data set. To make data, in particular the position, free from noise/glitch, the Kalman Filter is used before the position is further processed. The novelty lies on using difference instead of the absolute position only to predict/calculate next position. The outputs (position, roll, pitch and yaw), on the next period, produced by real UAV system were similar to the predicted outputs produced by Neural Network model. Furthermore adaptive direct inverse control is used to control the UAV follows a predetermined reference position.

Keywords Artificial neural network, Identification, Back propagation, Direct inverse control, Kalman filter

1. Introduction

To accurately control a system, it is beneficial to first develop a model of the system. The main objective for the modeling task is to obtain a good and reliable tool for analysis and control system development. A good model can be used in off-line controller design and implementation of new advanced control schemes. In some applications, such as in Unmaned Aerial Vehicle (UAV), it is very safe and advantageous to tune controllers off-line before implemented directly on the plant. In such cases, an accurate model must be used off-line for the tuning and verification of the controller. While nearly all aspects of modeling and simulation in control systems have now reached a reasonable stage of development, the aspect which remains least satisfactory at the present time is that of representing the loads supplied from systems due to the very wide range of load types.

Most motion control systems driven by motors exhibit nonlinear behavior and are often difficult or unrealistic to model directly using laws of physics. The presence disturbance such as wind in any direction, is the main nonlinear element in motion control systems. In general, a linear system allows the use of more sophisticated advanced control schemes to achieve higher performance. However, various other nonlinear elements exist in UAV system. The voltage source pulse width modulation (PWM) amplifier is used and dead time is required to prevent the shoot-through phenomenon during switching, which results in a nonlinear effect to the system. In an extreme case, the distorted output voltage produces torque pulsation and instability at low-speed. Hur et al. [1] proposed a 2 degree-of freedom (2 DOF) controller employing an inverse current dynamic model and a PI controller to compensate the effects of the dead time for induction motor control. The 2 DOF controllers have also been extensively studied in the area of motion control to suppress disturbances [1].

In this paper, we propose a method to obtain an accurate nonlinear system model to identify UAV system based on neural networks (NNs). Modeling techniques based on NNs have proven to be quite useful for building good quality models from measured data. If such an NN model is available, various control synthesis approaches may be attempted, even if the controllers themselves are not implemented in neural networks. It is possible to use a number of conventional nonlinear design techniques such as feedback linearization, generalized predictive control, or

* Corresponding author:
afrias.sarotama@ui.ac.id (Afrias Sarotama)
Published online at http://journal.sapub.org/ajis

model linearization followed by a linear design. Another approach is to use a neural network as the controller; e.g., direct inverse control or internal model control [6-9].

A model must be found that combines both robustness and accuracy to the desired extent. As well, the model should be computationally efficient and economical in order to be applied in mass-produced systems. To succeed in fulfilling these criteria, we apply a Neural Network controller to compensate the effects of disturbance without degrading tracking performance for a real-life system modeled in a NN.

For the experimental system we consider UAV-Alap-alap system. It requires high speed, robustness and accuracy. It is equipped with servo motors to control the positions of left elevon, right elevon and throtle. The servo motors are widely used as an actuator since it has a high torque-to-weight ratio, are easy to control, and has high efficiency and negligible maintenance requirements. To name a few, torque ripples and shifting of central of gravity during the flight are, however, two of the disadvantages of the servo motor and may be considered as a nonlinearity. In the present work, we model an UAV system, which has servo motors, using NNs and then proceed to develop suitable controller synthesis techniques for such a system.

The next step after modeling the UAV Alap-alap is to design control system for the UAV based on adaptive direct inverse control.

Apart from this introductory section, the article is organized as follows: Section 2 describes the setup used for data acquisition in our experiments. The system identification procedure for the sewing machine is detailed in Section 3, where controller synthesis methodology and experimental results are also presented. Concluding remarks are given in Section 4.

2. Data Aquisition

Data aquisition is done by UAV Flight Management System, Picollo both on board and on the ground.

2.1. Hardware Configuration

Figure 1. Data aquisition preparation of AUV-Alap-alap

To collect input and output data, an inverted-V-tail twin boom UAV made by BPPT is used. This UAV, shown in Fig. 1, is called Alap-alap has 3.51 meter of wingspan, max 18kg of take-off weight and cruise speed of 55Knot.

Input and output data such as time, longitude, latitude, altitude, North-velocity, East-velocity, Down-velocity, roll, pitch, yaw, angles speed, aileron, left ruddervator, right ruddervator, throttle, are collected during a test flight by a UAV Flight Management System, Picollo, made by Cloud Cap Technology.

2.2. Data Collections

Collections of input and output data were recorded from a flight test of the UAV. During the flight test, UAV Alap-alap was flown from the ground 40 meter above sea level, to the maximum height 436.5 meter above the sea level, and landed back after 36250 sampling of inputs and outputs. Fig. 2 shows Longitude-Latitude of the UAV during data acquisition. Fig.3 shows UAV Longitude-Height during data acquisition.

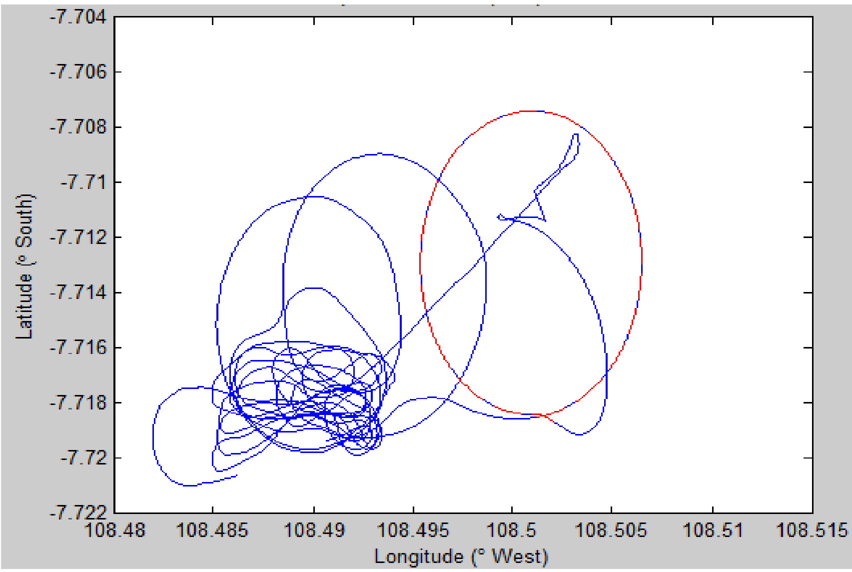

Figure 2. UAV-Alapalap track in longitude-latitude during flight test

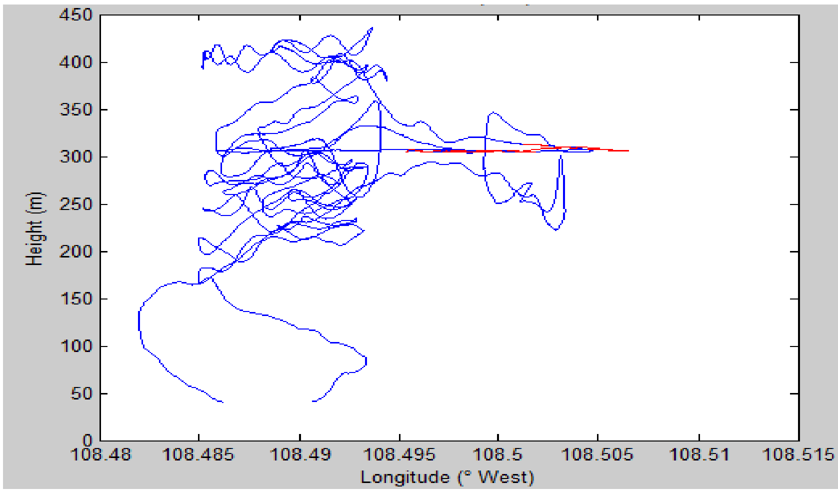

Figure 3. UAV-Alapalap track in longitude-height during flight test

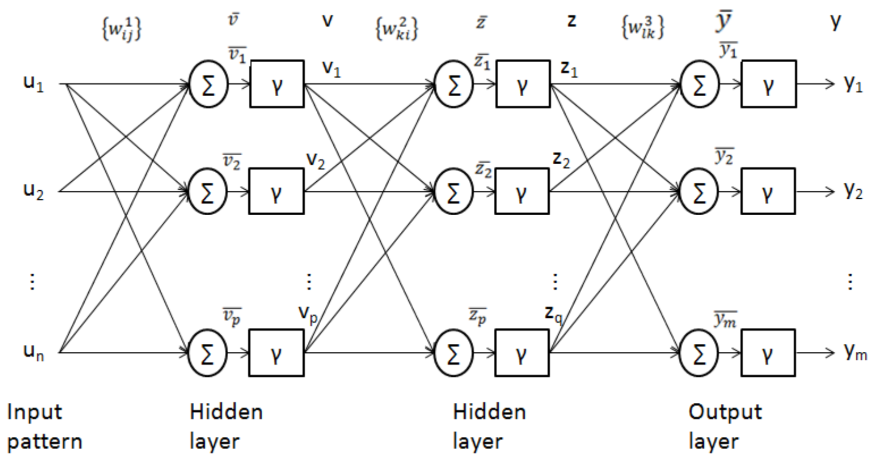

Figure 4. A Three layer neural network

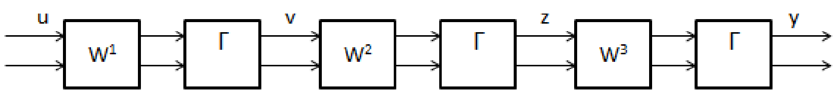

Figure 5. A block diagram representation of a three layer neural network

Two sets of data were assigned from the available data. Flight data from a circular form flight, given by red line in Fig. 1 and Fig. 2, is assigned as test data and the rest is assigned as training data.

3. System Identification

The objective is to carry out system identification of the UAV system by using neural networks. Three inputs (u), consist of left elevon, right elevon and throtle are applied to the UAV system while the output (y) are roll, pitch and yaw. Six states variables Ax, Ay, Az, P, Q adn R are also recorded during a flight test. Two sets of data flights were derived from 36250 sample flight test data. For the first data flight set, a chirp signal, are used for training the neural network to determine parameters (weights) for the network, using all sample flight which are not belong to the second data set. Validation is performed using the second data set, which were not used for training, representation if UAV circular flight movement. We shall refer to the second data set as the test data. It is important to use the test data for validation to ensure that our neural network model does replicate the UAV system in general rather than memorize a specific data set.

3.1. Back Propagation

This part will describe back propagation method to deter-mine/tune the wiegths/parameters in a neural network which determine the output. A typical multilayer network with an input layer, an output layer, and two hidden layers is shown in Fig. 4. For convenience we denote this in block diagram form as shown in Fig. 5 with three weight matrices W^1, W^2, and W^3 and a diagonal nonlinear operator γ with identical sigmoidal elements $\gamma[i.e., \gamma(x) = (1 - e^{-x})/(1 + e^{-x})]$ following each of the weight matrices. Each layer of the network can then be represented by the operator $N_i[u] = \Gamma[W^k u]$ and the input-output mapping of the multilayer network can be represented by $y = N[u] =$

$\Gamma\left[W^3\Gamma[W^2\Gamma[W^1]]\right]$.

In practice, multilayer networks have been used successful-ly in pattern recognition problems [2-5]. The weights of the network W^1, W^2, and W^3 are adjusted to minimize a suitable function of the error e between the output y of the network and a desired output yd. This results in the mapping function $N[u]$ realized by the network, mapping vectors into corresponding output classes. Generally a discontinuous mapping such as a nearest neighbor rule is used at the last stage to map the input sets into points in the range space corresponding to output classes. From a systems theoretic point of view, multilayer networks can be considered as versatile non-linear maps with the elements of the weight matrices as parameters. In the following sections we shall use the terms "weights" and "parameters" interchangeably. Detail of weights adjustment can be seen on [10].

3.2. System Identification Using a Neural Network

In the identification framework, we assume that the UAV model can be represented in discrete input-output form by the identification structure:

$$\hat{y}[k] = \hat{g}\,[y(k-1), \ldots, y(k-n_a), y(k-1), u(k-n_k), \ldots, u(k-n_b-n_k+1)\ ] \qquad (1)$$

where $\hat{y}[k]$ is the one-step ahead prediction of the output; and na, nb, nk are system order and delay, respectively. This is essentially a one-step ahead prediction structure in which we use past inputs and outputs to predict the current output. Output network at moment k is determined by input at moment k and several periods before k, and also by output several periods before k.

Using our intuition concerning the input-output model for the UAV system, a third order system is selected for the identification structure. Therefore, na = nb = 3 and nk = 1 in the structure above. We use the neural network $\hat{g}[.]$ to model [.]. The [ ] contains the regressor structure, which is implemented as Tapped Delay Lines (TDLs) in code. Therefore, the regressor structure for this network is given by:

$$\phi(k) = [\hat{y}(k-1), \ldots, \hat{y}(k-n_a), u(k-n_k), \ldots, u(k-n_b-n_k+1)] \qquad (2)$$

where $\hat{y}(k)$ are delayed versions of the predicted outputs and $u(.)$ are delayed inputs to the system. At every instant, the predicted output is parameterized in terms of network weights Θ by:

$$y(k,\Theta) = g(\phi(k),\Theta) \qquad (3)$$

and is depicted in Fig.6.

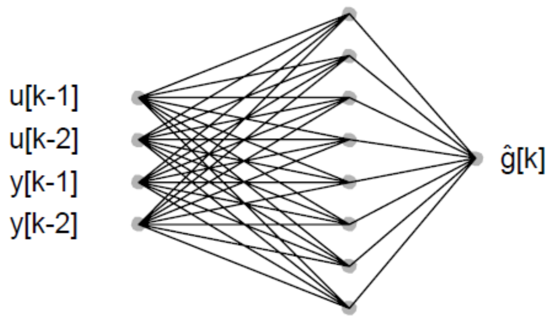

Figure 6. The architecture for the $\hat{g}[\cdot]$ network

Third order of dynamic system is considered adequate to represent most of real system. Learning process to set parameters in neural network model to produces predicted output, $y_p(k)$, is given in Figure 7.

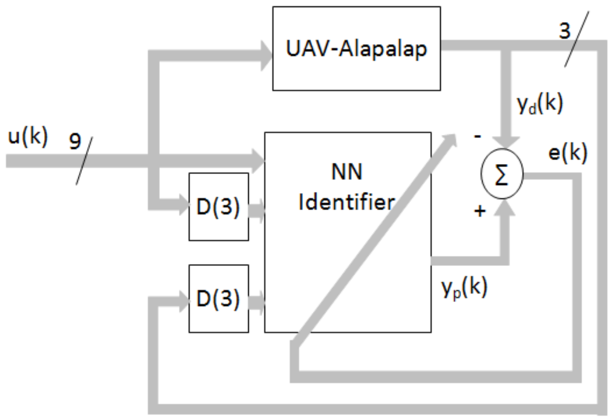

Figure 7. Neural-network learning process

Where:

u(k): Three inputs, LeftAileron(k), LeftRuddervator(k), RightRuddervator (k),

y_d (k): Five real outputs, $Roll_d$ (k), $Pitch_d$ (k) , $Longitude_d$ (k), $Latitude_d$(k) and $Altitude_d$ (k)

y_p (k): Five predicted outputs, Roll $_p$ (k), Pitch $_p$ (k) , Longtude $_p$ (k), Latitude $_p$ (k) and Altitude $_p$ (k)

D(3): Three period Tapped Delay Line/TDL

It will be shown later that by doing this way, the position will be calculated with bad accuracy. Po-sition will be better calculated if neural network uses position difference in one period instead of absolute position. If the position difference is de-fined as:

$$\delta\hat{y}(k+1) = \hat{y}(k+1) - \hat{y}(k);\ \ \delta\hat{u}(k+1) = \hat{u}(k+1) - \hat{u}(k) \qquad (4)$$

$$\delta\hat{y}(k+n_k) = \hat{g}\left[\delta y(k+n_k-1), \ldots, \delta y(k+n_k-n_a+1), \ldots, \delta u(k), \ldots, \delta u(k-n_b+1)\right] \qquad (5)$$

Then the neural network structure uses the position difference in one periode can be given in Fig. 8.

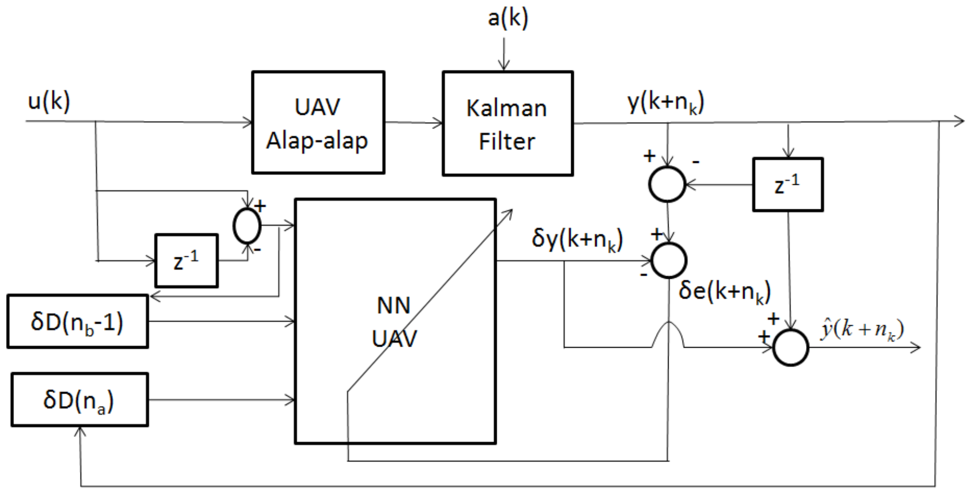

Figure 8. Neural network structure using one periode position difference

3.3. Direct Inverse Control

In direct inverse control (DIC), currennt inputs are calculated given the outputs and inputs in other periods, according to eq. (6).

$$\hat{u}[k]=\hat{g}^{-1}[y(k+n_k),y(k+n_k-1),...,y(k),y(k+d-na+1),u(k-1),...,u(k-n_b+1)] \tag{6}$$

Also Eq.(7) can be used.

$$\hat{u}[k]=\hat{g}^{-1}[y(k+n_k),y(k),y(k-na+1),...,u(k-1),...,u(k-n_b)] \tag{7}$$

The whole system, both identification and control is depicted in Fig.9.

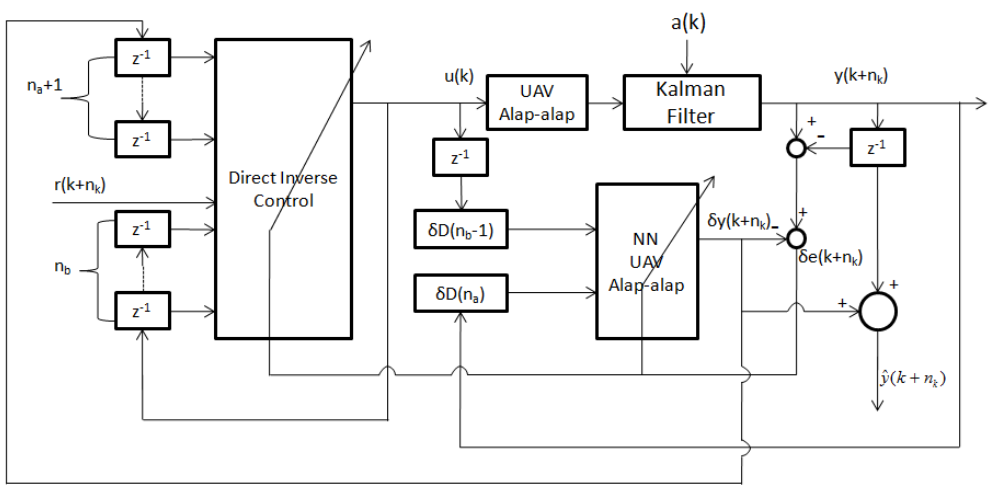

Figure 9. Identification and Control of UAV Alap-alap

3.4. The Results

After learning to determine Neural Network parameters, as depicted in the Fig. 9., the second set of input were fed in to the network to get the first predicted output, $y_p(k)$ (roll, given by red line) and can be compared to the desired/real roll(given by blue line), $y_d(k)$.

As seen from the Fig 9., the neural network approximate the desired output(roll) quite accurately. The difference between the first desired output, $y_d(k)$, and predicted output, $y_p(k)$, is shown by the Figure 10.

The second output (pitch) is shown in the figure 11., where desired output, $y_d(k)$, is drawn in blue line, and predicted output, $y_p(k)$, is drawn in red line.

As seen from the Figure 11., the neural network approximate the desired output (pich) quite accurately. The difference between the second desired output, $y_d(k)$, and predicted output, $y_p(k)$, is shown by the Figure 12.

The third output (yaw) is shown in the Figure 13., where desired output, $y_d(k)$, is drawn in blue line, and predicted output, $y_p(k)$, is drawn in red line.

As seen from the Figure 14., the neural network approximate the desired output (yaw) quite accurately. The difference between the third desired output, $y_d(k)$, and predicted output, $y_p(k)$.

The real positon of the UAV is calculated aproximately by neural network which uses position difference in calculation is shown in Figure 15.

The real positon of the UAV is calculated with less precision by neural network which uses absolute position in calculation is shown in Fig. 16.

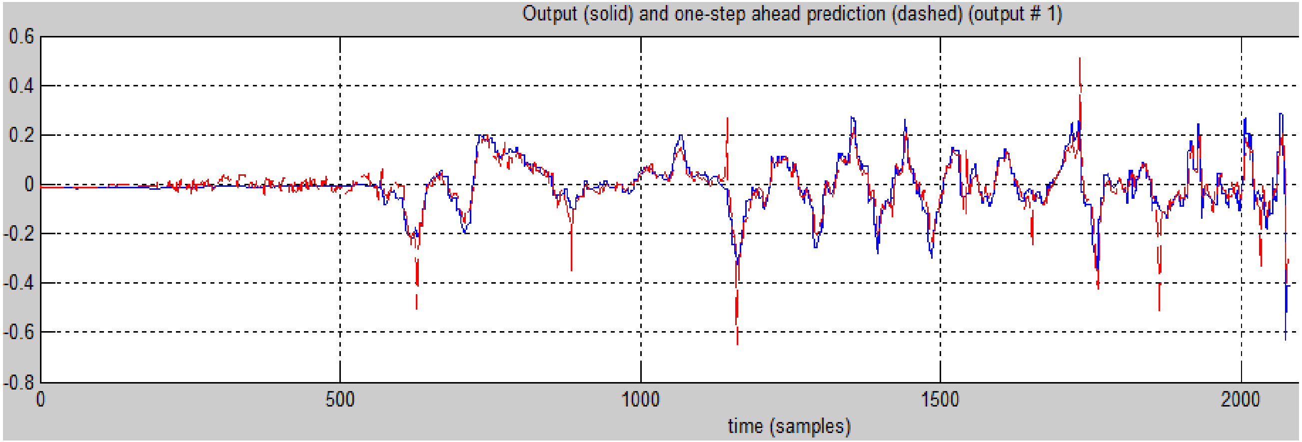

Figure 9. Comparison of predicted and desired roll

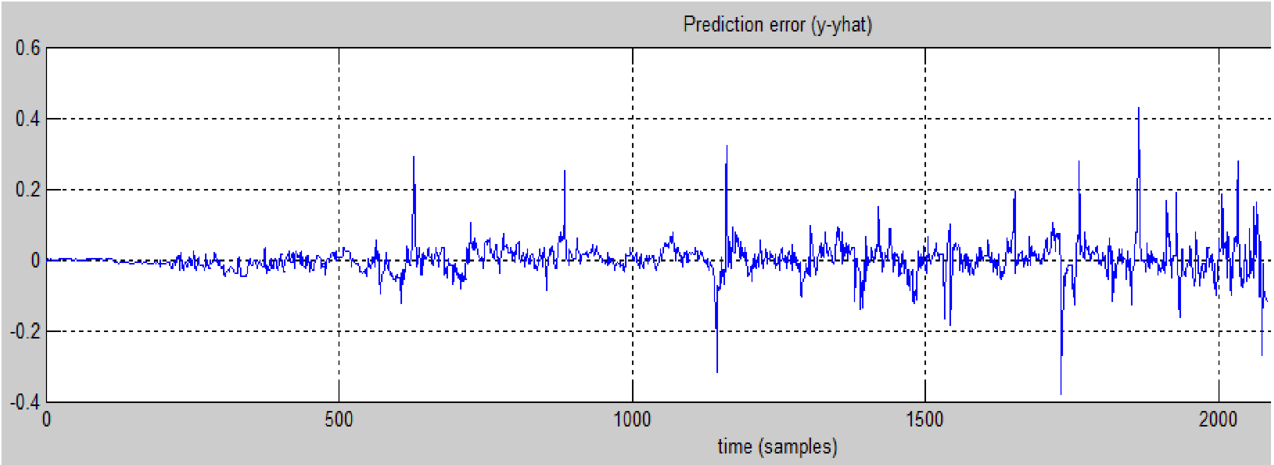

Figure 10. The difference/Error between predicted and desired roll

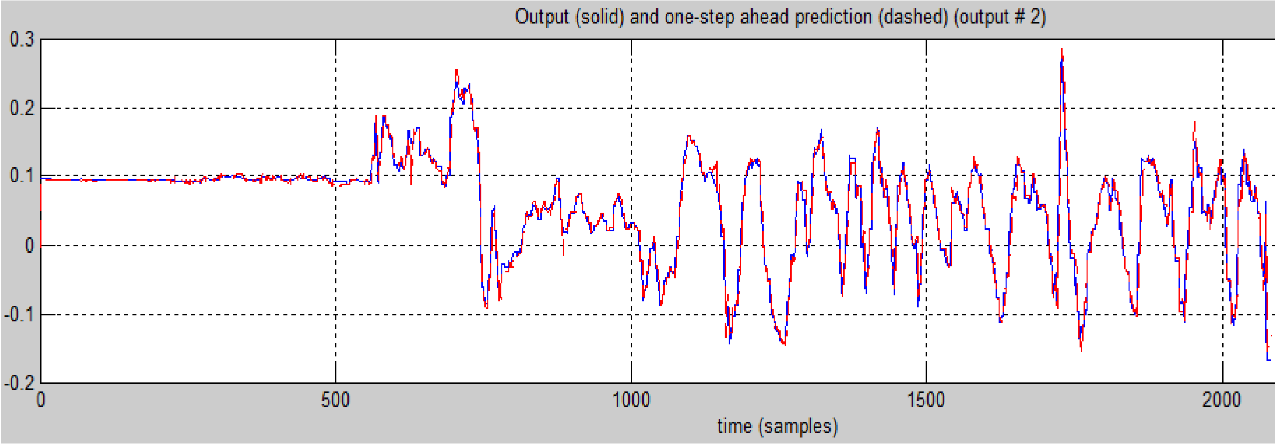

Figure 11. Comparison of predicted and desired pitch

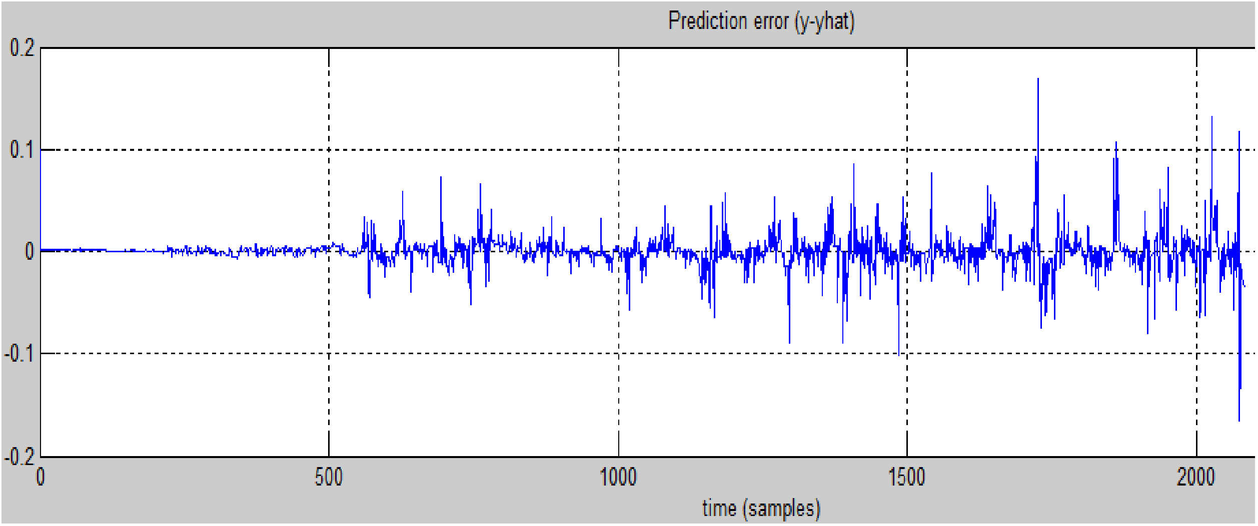

Figure 12. The difference/error between predicted and desired pitch

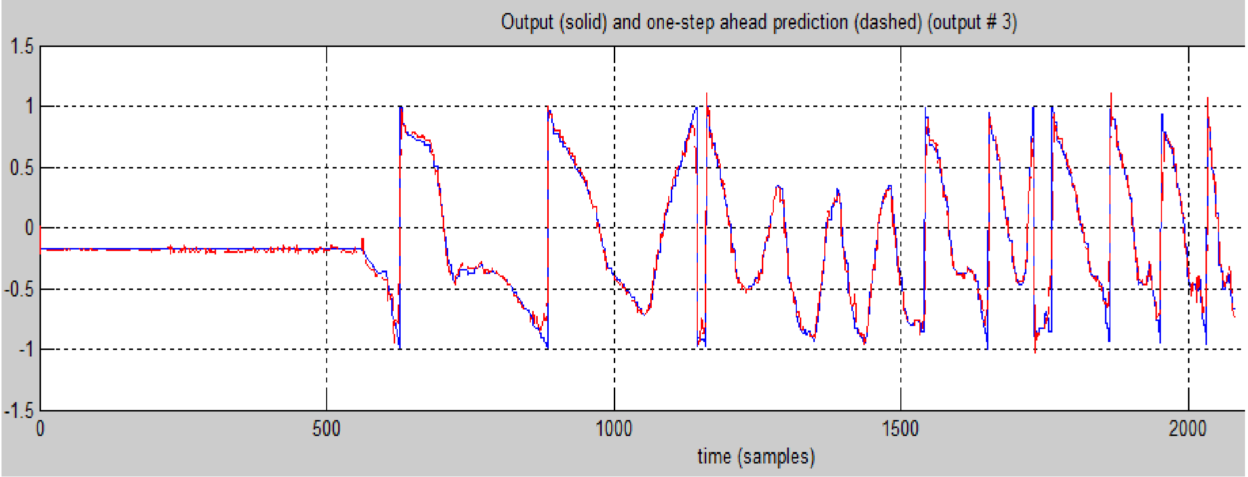

Figure 13. Comparison of predicted and desired yaw

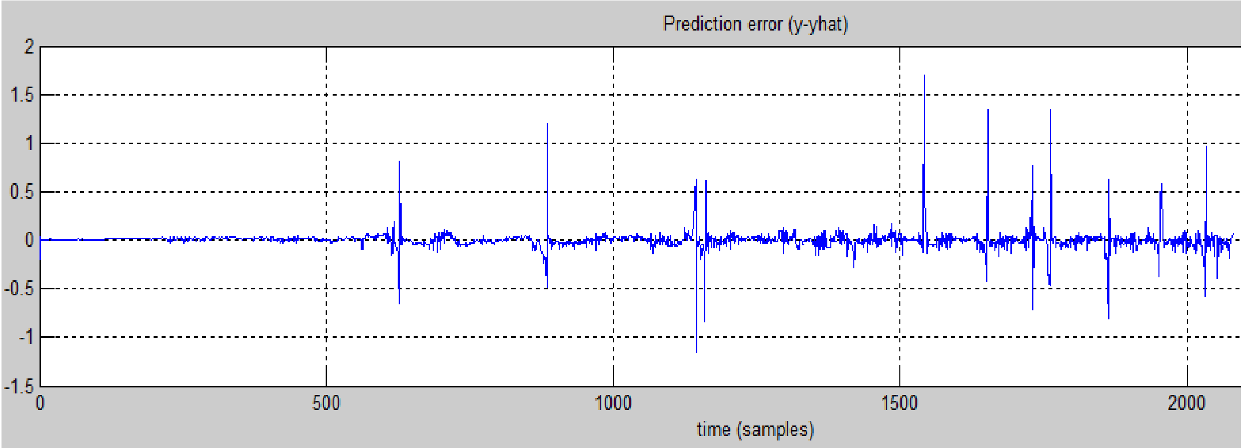

Figure 14. The difference/error between predicted and desired yaw

4. Conclusions

Neural network Back Propagation was used to model and control UAV-Alap-alap. Using artificial neural network is more convenient compared to physics and mathematics mehods. Parameters of neural network were calculated using inputs and outputs data, got from a flight test, from 40 m above sea level to 354.4 meter above the sea level and back. From 36250 sampling data, two sets of data were formed. A set of 3981 sample data, taken from a circular form track is assigned a test data and the rest is set as a training data set. After an Artificial Neural network was trained using the training data set, the network is excited by the second set input data set. To make data, in particular the position, free from noise/glich, the Kalman Filter is used before the position is further processed. The desired outputs (roll, pitch and yaw) produced by real UAV system were identical to the predicted outputs produced by Neural Network model. The Neural Network has to be adjusted when position output is involved. The precision will be more accurate when the position differences are used in the calculation, instead of absolute position.

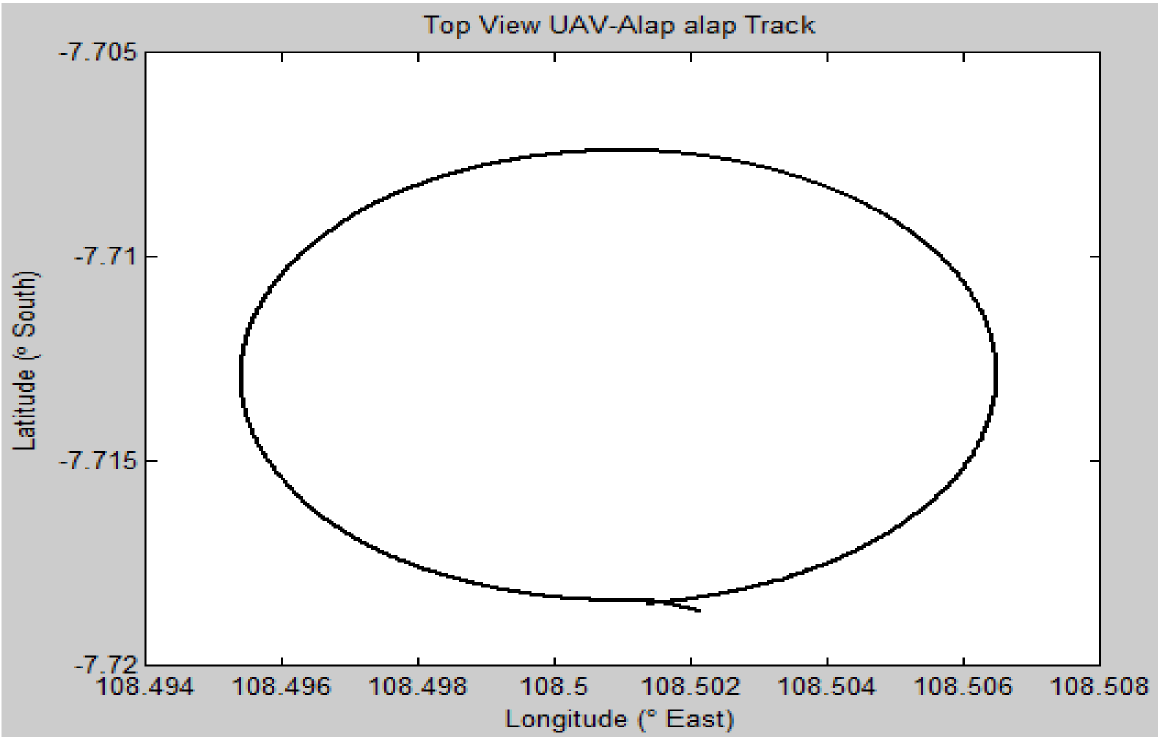

Figure 15. Comparison of real and calculated neural network position. Position difference are used in calcultaion

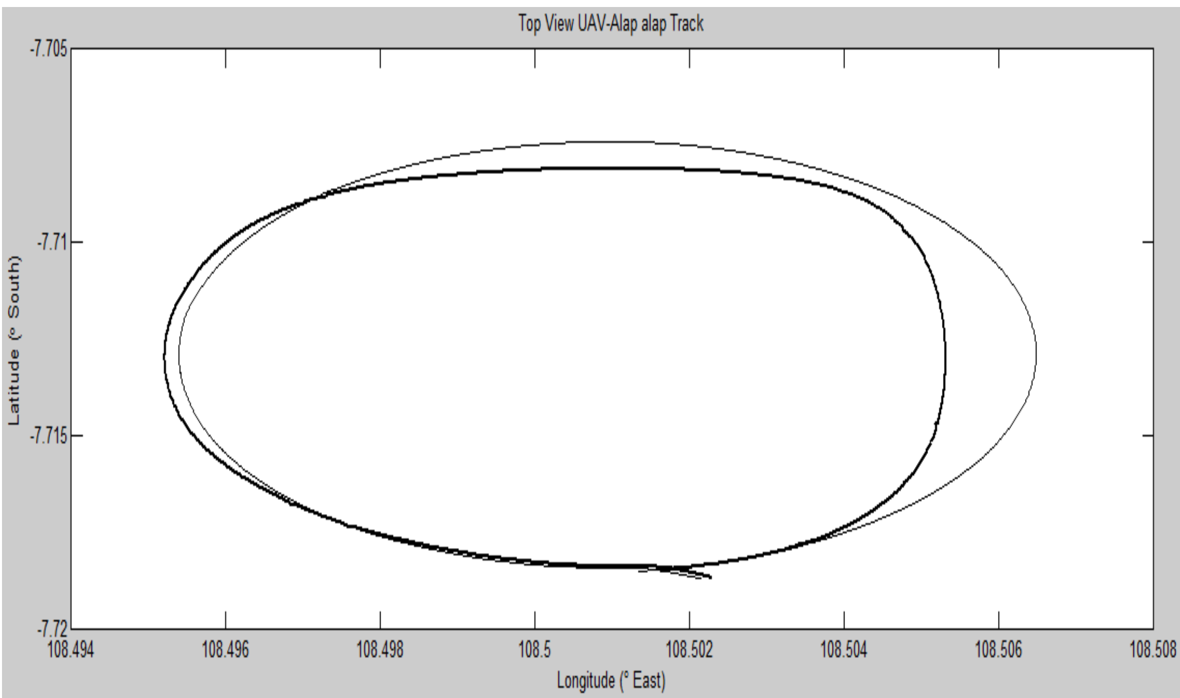

Figure 16. Comparison of real and calculated neural network position of UAV-Alap-alap. Absolute position are used in calcultaion

REFERENCES

[1] Il-Hwan Kim, Stanley Fok, Kingsley Fregene, Dong-Hoon Lee, Tae-Seok Oh, and David W. L. Wang(2004), Neural Network-Based System Identification and Controller Synthesis for an Industrial Sewing Machine, International Journal of Control, Automation, and Systems Vol. 2, No. 1.

[2] Lee, R., Shen, L. (2005), System Identification of Cessna 182 Model UAV, Stanford University.

[3] Narendra, K. S., Pathasarathy, K. (1990), Identification and Control of Dynamical Systems Using Neural Networks, IEEE Transactions on Neural Networks, Vol. 1, No.1.

[4] Sefer Kurnaz, Omer Cetin, Okyay Kaynak (2010), Adaptive neuro-fuzzy inference system based autonomous flight control, Expert Systems with Application Elsevier of unmanned air vehicles.

[5] Sofyan, E. (1996), Aerospace Engineering Department Royal Melbourne Institute of Technology Victoria, Australia, Identification of Model Aircraft Dynamic Using Flight Testing, Phd Dissertation.

[6] Zhang Huaguang and Quan Yongbing, “Modeling, identification, and control of a class of nonlinear systems,” IEEE Trans. on Fuzzy Systems, vol. 9, no. 2, pp. 349-353, 2001.

[7] K. K. Safak and O. S. Turkay, “Experimental identification of universal motor dynamics using neural networks,” Mechatronics, vol. 10, pp. 881-896, 2000.

[8] M. Novakovic, “Discrete time neural network synthesis using input and output activation functions,” IEEE Trans. on SMC, vol. 26, no. 4, pp. 533-541, 1996.

[9] D. Schroder, C. Hintz, and M. Rau, “Intelligent modeling, observation, and control for nonlinear systems,” IEEE Trans. on Mechatronics, vol. 6, no. 2, pp. 122-131, 2001.

[10] Kumpati S. Narendra, Kannan Parthasarathy, “Identification and Control of Dynamical Systems Using Neural Networks”, IEEE Transaction on Neural Networks, Vol. 1 no. 1, March 1990.