System Identification for Quad-rotor Parameters Using Neural Network

Dief, Tarek N.
Graduate School of Engineering Sciences, Kyushu University

Yoshida, Shigeo
Faculty of Engineering, Kyushu University

https://doi.org/10.5109/1657380

出版情報：Evergreen. 3 (1), pp.6-11, 2016-03. Green Asia Education Center
バージョン：

System Identification for Quad-rotor Parameters Using Neural Network

Tarek N. Dief[1,*], Shigeo Yoshida[2]

[1]Graduate School of Engineering Sciences, Kyushu University, Japan

[2]Faculty of Engineering, Kyushu University, Japan

*Author to whom correspondence should be addressed,
E-mail: tarek.na3em@riam.kyushu-u.ac.jp

(Received January 4, 2016; accepted February 24, 2016).

This paper presents a new technique to identify the system parameters without using the system governing equations. This technique is the time series prediction using neural network. The theoretical model was applied using simulations, after that the experiments were done to get the suitable construction for the neural model. A comparison between neural network and placket's model is discussed. The advantages and disadvantages of both models were explained. The main idea of neural network is based on back-propagation algorithm. The equations and steps for iteration are presented and the relation between changing the number of iteration with the system frequency. The controller used is pole placement controller based on the neural network results as a system model.

Keywords: system identification, neural network, Quad-rotor, and pole placement

1. Introduction

In the recent years, Quad-rotors become a very popular unmanned aerial vehicle (UAV) platform, Owing to the always increasing performance of the Micro Electro Mechanical Systems (MEMS). The applications of Quad-rotor are increasing by a high rate, and also the control algorithms which are applied to Quad-rotors are depending on these applications. Some of these applications are such as line tracking [1-3], Videos capturing, pollution measuring, and high endurance flight using fixed wing UAV for power saving [4]. And we have some applications needs the system's parameters change during the flight, and the controller should be changed due to this change of the model parameters. Applying Neural Network for the Quad-rotor in most researches were used to design the controller and choose the suitable control gains, but in this paper we tried to use the neural network as a system identification to update the governing equations of the Quad-rotor in real-time. Then the control algorithm also will be updated due to the change in the model of the system. The work of system identification using placket's model [5] was discussed to be compared with the neural network algorithm. The advantages and disadvantages of each algorithm were discussed in this paper, and the application will decide which algorithm is better to be used.

2. Neural Network Model

From the work of parameter estimation in [5], we can develop the model and controller of the Quad-rotor system. We can identify the system parameters in real-time using placket's model. These results are acceptable with some comments such as the settling time, and the effect of initial conditions. For equations (1) & (2), we have the governing equations for the Quad-rotor in S-domain and Z-domain. In equation (1), we just have one unknown; this unknown is function of the Quad-rotor structure such as mass, inertia, and arm length. Also each of roll, Pitch and yaw has different value for c . But for equation (2), we have three unknowns because the transformation to Z-domain will be for second order polynomial equation. Also the values of a_1 & a_2 will be fixed even for any changes could be happened for c as shown in figure (7).

$$G(s) = \frac{y}{u}(s) = \frac{c}{s^2} \tag{1}$$

$$G(z) = \frac{y(z)}{u(z)} = \frac{\frac{1}{2}cT^2(Z^{-1}+Z^{-2})}{1+a_1Z^{-1}+a_2Z^{-1}} \tag{2}$$

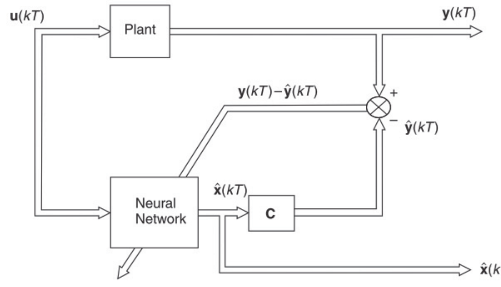

Fig 1: Block diagram of the neural network model and the system controller.

$$b = \frac{1}{2} c T^2 \tag{3}$$

The structure of the neural network is shown in figure (1); it depends on the sensor readings then the model will be predicted by getting the estimated parameters. Applying the back-propagation algorithm is a good method for training Artificial Neural Network (ANN), and it is one of the most used techniques for representing the gradient-descent optimization method [8], [15]. And for the cost function J is defined as the square difference of the error as shown in equation (4),

$$J = \frac{1}{2} \sum_{j=1}^{N} (d_j - y_j)^2 \tag{4}$$

where d_j is the desired value and y_j is the actual network outputs respectively. After applying the gradient-descent, the negative slope will be changed proportional to the weight increment Δw_{ji} as shown in equation (5).

$$\Delta w_{ji} = -\mu \frac{\partial J}{\partial w_{ji}} \tag{5}$$

From equations (4) and (5) we can get:

$$\frac{\partial J}{\partial w_{ji}} = \frac{1}{2} \sum_{k=1}^{N} \frac{\partial}{\partial w_{ji}} (d_k - y_k)^2 \tag{6}$$

After using chain rules we can get:

$$\frac{\partial J}{\partial w_{ji}} = \frac{1}{2} \sum_{k=1}^{N} \frac{\partial}{\partial y_j} (d_k - y_k)^2 \frac{\partial y_j}{\partial w_{ji}} \tag{7}$$

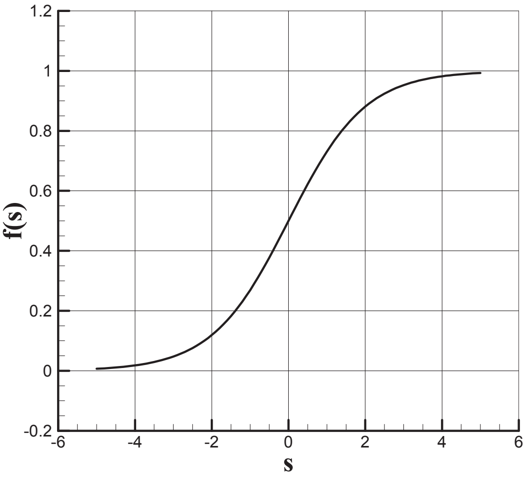

Fig 2: Sigmoid function.

So the results will be as shown in equation (8)

$$\frac{\partial J}{\partial w_{ji}} = -\sum_{k=1}^{N} (d_k - y_k) \frac{\partial y_j}{\partial w_{ji}} \tag{8}$$

For applying this algorithm we need to use activation function such as a sigmoid function as shown in figure (2) and equation (9).

$$\frac{\partial f}{\partial s} = \frac{e^{-s}}{(1+e^{-s})^2} = \frac{1}{1+e^{-s}} - \frac{1}{(1+e^{-s})^2} \tag{9}$$

And after simplification we can derive:

$$\frac{\partial f}{\partial s} = f(s) - f(s)^2 \tag{10}$$

As we consider the sigmoid function as a neuron's output y_j, so we can rewrite the equation as shown in equation (11).

$$\frac{\partial y_j}{\partial s_j} = y_j (1 - y_j) \tag{11}$$

Also, we can use chain rules for equation (8). Then,

$$\frac{\partial y_j}{\partial w_{ji}} = \frac{\partial y_j}{\partial s_j} \frac{\partial s_j}{\partial w_{ji}} \tag{12}$$

And for the sigmoid function, we can rewrite it as a function of weights as shown in equation (13), where x_i is the vector which has the system estimated parameters.

$$s_j = \sum_{i=1}^{N} w_{ji} x_i \qquad (13)$$

So we can get,

$$\frac{\partial s_j}{\partial w_{ji}} = \frac{\partial}{\partial w_{ji}} \sum_{i=1}^{N} w_{ji} x_i = \sum_{i=1}^{N} \frac{\partial w_{ji}}{\partial w_{ji}} x_i = x_i \qquad (14)$$

From equation (11) , (12) and (14), we can get

$$\frac{\partial y_j}{\partial w_{ji}} = y_j(1 - y_j)x_i \qquad (15)$$

Substituting by equation (15) into (8) gives

$$\frac{\partial J}{\partial w_{ji}} = -\sum_{j=1}^{N} (d_j - y_j) y_j (1 - y_j) x_i \qquad (16)$$

And assuming δ_j as shown in equation (17)

$$\delta_j = (d_j - y_j) y_j (1 - y_j) \qquad (17)$$

Then,

$$\frac{\partial J}{\partial w_{ji}} = -\sum_{j=1}^{N} \delta_j x_i \qquad (18)$$

And after simplification we can rewrite equation (18) as shown in equation (19).

$$\Delta w_{ji} = \mu \sum_{j=1}^{N} \delta_j x_i \qquad (19)$$

Where μ is a learning rate between 0 and 1, T is the sample time in second, and K is the counter of the to summing. The new weight increment will be as shown in equations (20-23):

$$\Delta w_{ji}(KT) = \mu \delta_j x_i \qquad (20)$$

$$w_{ji}(KT) = w_{ji}((K-1)T) + \Delta w_{ji}(KT) \qquad (21)$$

$$w_{ji}(KT) = w_{ji}((K-1)T) + \mu \delta_j x_i \qquad (22)$$

For the system of Quad-rotor, the neural network algorithm is consisting of three layers; input layer $l0$, hidden layer $l1$, and output layer $l2$. And the hidden layer consists of three nodes as shown in figure (3). The back-propagation starts with the input layer $l0$ as the value of d_j is already known and the values of δ_j are calculated from equation (17), and the weights are calculated from equation (20). Then, the weights for the hidden layers $l1$ can be calculated, but equation (17) will be replaced by equation (23).

$$[\delta_j]_l = [y_j(1 - y_j)]_l [\sum_{j=1}^{N} w_{ji} \delta_j]_{l+1} \qquad (23)$$

So the updates for value δ at layer l will be calculated from the outputs at layer $l2$ and the summation of δ and w products from layer $l+1$. Then the iterations will be till convergence for the error. After that, a new set of inputs will be applied using the weights of hidden layers after iterations and go in the forward direction and till going to the output layer. These processes are going till:
- The cost function reaches to an acceptable value.
- Exceeding the maximum number of iterations.

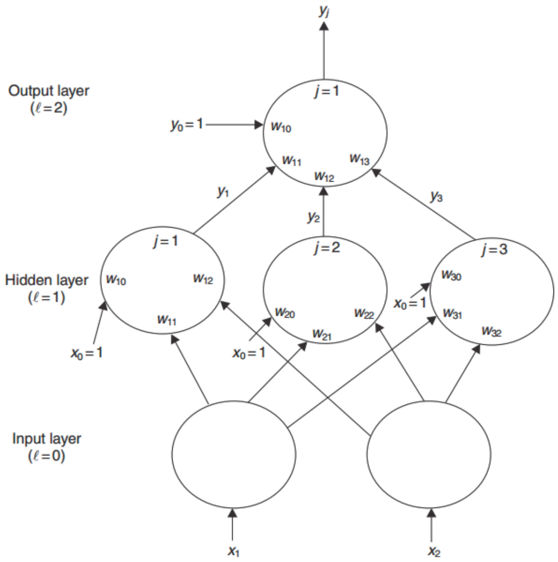

Fig 3: Neural network construction.

3. Results

Finally, for applying this algorithm to Roll, Pitch, and Yaw, the results as shown in figures (4-6). These results got from the theoretical simulations. As shown in figures (8) the comparison between the actual data and the estimated data using the neural network technique. The initial values for this simulation are zeros.

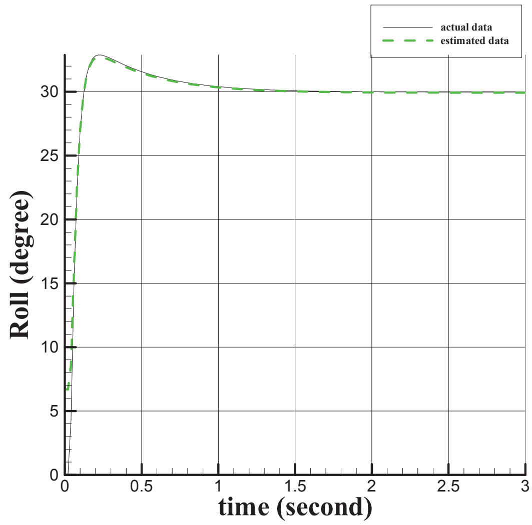

Fig 4: Roll response with the time for actual and estimated data.

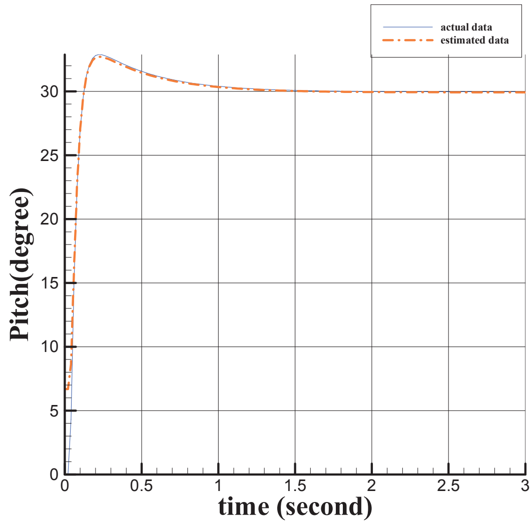

Fig 5: Pitch response with the time for actual and estimated data.

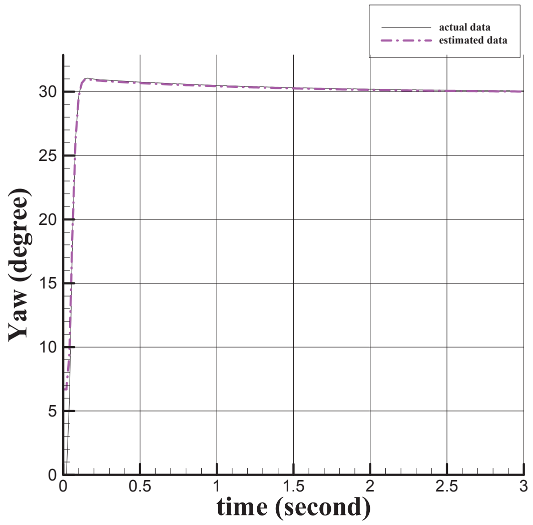

Fig 6: Yaw response with the time for actual and estimated data.

To estimate the system parameters, we used the difference equation to estimate the open loop transfer function parameters shown in equations (2) and (3). From the results, we found the values of a_1 and a_2 will not change; as the system originally consists of two integrators. And the change will be only for the value of b . For calculating this value we used the difference equation and we already have the values of sensor readings and inputs. So this technique can give accurate response in short time even the initial values are zeros. On the other hand, the placket's model needs around 50 second to settle the accurate data. If the initial values are zeros or not close to the real values, it leads to instability for the system.

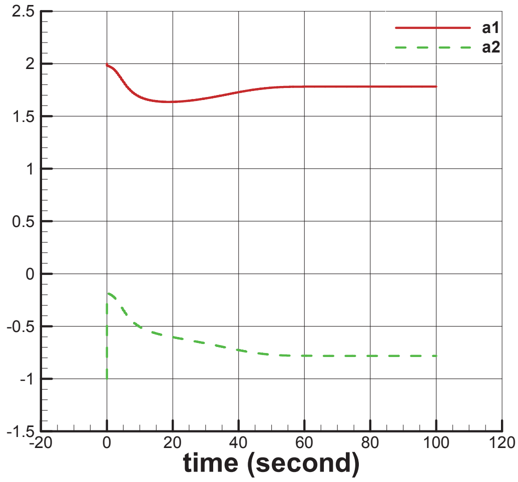

Fig 7: Results from placket's model to identify the value of a1 & a2.

Finally, after applying the neural network to Quad-Rotor and plot the resulting data in real-time, we found the estimated data is close to the real data got from the sensor. Also the estimated data is function of the number of iterations which is used during implementation. The most suitable number of iterations for the data estimation was above 60 iterations, this number of iteration costs high rating time for the microcontroller, which means instability for the system. To solve the problem of sample rate, we reduced the number of iterations till 30 iterations.

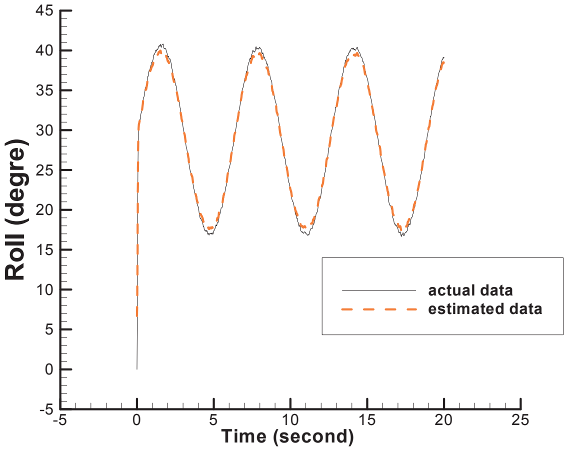

Fig 8: Comparison between the actual data and the estimated roll from neural network.

For the reason of using neural network instead of placket's model, we can conclude the work for placket's model as following:

➢ Advantages:
- Running time is very low (no iterations).
- Simple in programming.
- No singularities in the calculation, even the initial conditions are zeros.

➢ Disadvantages:
- Time settling for the system parameters is high if the initial conditions are not close to the real values.

Also for the using of neural network, it has some advantages and disadvantages such as:

➢ Advantages:
- Can give accurate data for any initial conditions.
- The iteration can recover quickly for any wrong data from the input. As the system depends mainly on the output data.
- Can be applied and combined with other controller such as (fuzzy, genetic, classical, etc…) as it's a discrete system.
- No-singularity.

➢ Disadvantages:
- Takes higher sample time as a result of iteration (0.04 second).

4. Control Design

For the controller used in the design, it was the same as [5], it is considered pole placement in the Z- domain. And the advantage of this controller is to be updated in real-time due to the change in the open-loop system's parameters. The results of the response are shown in figures (4-6).This controller is very simple and just needs the response requirements, and the open loop transfer function parameters. Then it can generate the required control action to stabilize the Quad-rotor.

5. Conclusion

Applying the neural network is very effective idea for a lot of reasons. Cause the system of Quad-rotor is Multi-Input Multi-Output (MIMO), so we can work with the system without any linearization. Also, most of works which have been presented using neural network were just used to control design problems, but for this paper we used it to make parameters identification to the system. The control parameters can be adapted to any change in the system governing equations, also we can change the system characteristics such as overshoot and the settling time even we have no change in the system parameters.

References

1) G. M. Hoffmann, S. L. Waslander, and C. J. Tomlin, “Quadrotor Helicopter Trajectory Tracking Control”, *Electr.Eng.,*vol.44,(doi: 10.2514/6.2008-7410), pp. 1–14 , August (2008).

2) Guney, M.A.; Unel, M., “Formation Control of a Group of Micro Aerial Vehicles (MAVs)”, *2013 IEEE International Conference on Systems, Man, andCybernetics(SMC),*Doi: 10.1109/SMC.2013.163, pp.929-934, 13-16 Oct. (2013).

3) S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, Vision-based state estimation and trajectory control towards high-speed flight with a quadrotor, *Robot. Sci.,*(2013).

4) Oosedo, A.; Konno, A.; Matumoto, T.; Go, K.; Masuko, K.; Abiko, S.; Uchiyama, M., “Design and simulation of a quad rotor tail-sitter unmanned aerial vehicle”, *2010 IEEE/SICE International Symposium*

on System Integration (SII), vol.26, Doi: 10.1163/156855311X614590 , pp.254-259, 21-22 Dec. (2010).

5) Tarek N. Dief, Shiego Yoshida, and Mohamed G. Abdelhady, "Attitude and altitude stabilization of quad rotor using parameter estimation and self-tuning controller", *AIAA Atmospheric Flight Mechanics Conference,* June (2015), Doi: 10.2514/6.2015-2392.

6) Deif, T., Kassem, A., El Baioumi, G., "Modeling and Attitude Stabilization of Indoor Quad Rotor", *International Review of Aerospace Engineering (IREASE), 7*(2), pp.43-47 (2014).

7) Etkin, B.: *Dynamics of Flight. Stability and Control,* 2nd ed., Wiley, New York (1982).

8) C.Nicol, C.J.B.Macnab, A.Ramirez-Serrano, "Robust neutral network Control of a quadrotor helicopter", Canadian Conference on Electrical and Computer Engineering. June, 2008.

9) R. Patel, M. a Patel, and D. R. Vyas, "Modeling and analysis of quadrotor using sliding mode control", *Proc. 2012 44th Southeast. Symp. Syst. Theory,* pp. 111–114 (2012).

10) S. C. Quebe, "Modeling, Parameter Estimation, and Navigation of Indoor Quadrotor Robots" (2013). All Theses and Dissertations. Paper 3565.

11) F. Q. Elizabeth and L. R. Garc, *American Control Conference*, "Quad-Rotor Switching Control : An Application for the Task of Path Following", Doi: 10.1109/ACC.2012.6315545, pp. 4637–4642 (2012).

12) M. Hassan Tanveer, S. Faiz Ahmed, D. Hazry, Faizan. A. Warsi, and M. Kamran Joyo, "Stabilized controller design for attitude and altitude controlling of quad-rotor under disturbance and noisy conditions", *Am. J. Appl. Sci.,* vol. 10, no. 8, pp. 819–831 (2013).

13) Qiang Jiang; Yong Zeng; Qiang Liu; Hua Jing, "Attitude and Heading Reference System for Quadrotor Based on MEMS Sensors", *Instrumentation, Measurement, Computer, Communication and Control (IMCCC), 2012 Second International Conference* , pp.1090-1093, *Doi:* 10.1109/IMCCC.2012.257, 8-10 Dec. (2012).

14) Seungho Jeong; Seul Jung, "Vision-based localization of a quad-rotor system", *Ubiquitous Robots and Ambient Intelligence (URAI), 2012 9th International Conference,*Doi:10.1109/ 2012.6463105, pp.636-638, 26-28 Nov. (2012).