Performance Analysis of a Backpropagation Neural Controller System for a Double-Propeller Boat Model

Karlisa Priandana[1], Benyamin Kusumoputro*
Department of Electrical Engineering
Universitas Indonesia
Depok, Indonesia
karlisa@ipb.ac.id, kusumo@ee.ui.ac.id

***Abstract*—This paper aims to analyze the performance of Backpropagation (BPNN)-controller system for an autonomous double-propeller boat model in terms of its settling time, in addition to the commonly analyzed control error. It is found that for a direct inverse neural controller system, the control error is highly affected by the required settling time, where most of the control errors are observed at the beginning of the control process.**

Keywords—Backpropagation; USV; boat; control system; direct inverse control; neural network controller.

I. Introduction

Until now, the proportional-integral-derivative (PID) controllers are the commonly preferred controllers for various autonomous systems, including Unmanned Surface Vehicle (USV), due to their low cost and simplicity in structure and design. One obvious drawback of these controllers is their inability to provide the same level of accuracy for the whole operating range, especially when the system is nonlinear. To overcome this problem, control systems based on mathematical models such as the backstepping techniques [1], sliding mode control [2], surface control [3], linear quadratic control [4] and internal model control [5] have been developed. However, to reduce the complexity of the problem, these mathematical based control systems require some considerable assumptions for simplifications and linearization [6]. Therefore, the model may not be able to represent the entire plant characteristics and properties. As a consequence, the designed controller may not perform well when it is utilized to control the system in a highly nonlinear environment.

In order to minimize the use of mathematical assumptions, controller designs using Artificial Neural Network (ANN) approach, hereinafter called the "NN-controller", has been widely studied. Several controllers for USV based on ANN are backpropagation controller [7], [8], backstepping-ANN hybrid controller [9], Self-Organizing Maps controller [10] and Radial-Basis Function neural network controller [11]. Among all of the developed ANN-controllers, backpropagation controller (BPNN-controller) is the most widely adopted NN-controller due to its simple structure and straightforward learning principle. The basis of backpropagation learning is iteratively adjusting the neural connection weight using the back-propagated error to minimize the difference between the desired output vector and the real output vector. Therefore, prior to training, each neuron connection in the determined network requires an initial weight value. When utilized as a control system, the designed BPNN-controller also suffers from the unknown initial values for the delayed inputs, so that it requires some settling time before the system becomes stable.

A detailed analysis of BPNN-controller has been presented in [12] and it is shown that this controller can produce a very low control error. However, the analysis on the performance of BPNN-controller in terms of its settling time has not been described yet. This paper aims to analyze the best method to determine the unknown initial values for the delayed inputs of the BPNN-controller system, as well as analyzing the controller performance in terms of its settling time, despite the control error as commonly conducted. The comparison of settling time and control error will be analyzed in this paper.

This paper is organized as follows. The next section describes the theory of NN-controller. Then, BPNN-controller system for a developed boat model is proposed in section 3. Section 4 compares some methods to determine the input-neurons initial values and analyzes the control error and settling time characteristics of the proposed BPNN-controller. The finding from this study is summarized in Section 5.

II. Neural Network Inverse Controller System

A. Neural Network Inverse Controller System

This study utilizes an open-loop direct inverse neural network controller as depicted in Fig. 1. In this scheme, the inverse neural network (NN-INV) controller is directly connected with the controlled plant which is modeled by a neural network plant identification system (NN-ID). In this scheme, the output of the system, $\boldsymbol{y(k)}$, is expected to be as similar as possible to the reference signal, $\boldsymbol{r(k)}$. The advantage of this controller system compared to the other controllers is its ability to utilize the superiority of neural network learning algorithm to calculate the most appropriate control signal, $\boldsymbol{u(k)}$. However, the values of initial plant outputs rely on the chosen values of initial weight matrix of the neural network as well as the initial values of the delayed input vector, $\boldsymbol{u(k-1)}$, in which these values cannot be determined at the initial

[1]The author is also with the Department of Computer Science, Bogor Agricultural University, Bogor, Indonesia.

operation of the plant. Therefore, the system may require some settling time at the beginning of the control process.

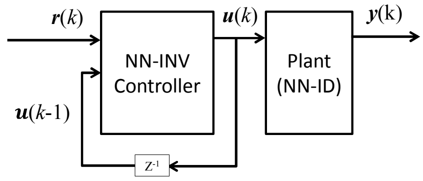

Fig. 1. Open-loop NN-based direct inverse control system

The neural network plant identification is done by adopting the nonlinear autoregressive exogenous model (NARX) [13], [14], expressed as the following equation:

$$\boldsymbol{y}(\boldsymbol{k}) = f\{\boldsymbol{y}(\boldsymbol{k}-\mathbf{1}), \ldots, \boldsymbol{y}(\boldsymbol{k}-\boldsymbol{n}_{\boldsymbol{y}}), \boldsymbol{u}(\boldsymbol{k}-\mathbf{1}), \ldots , \boldsymbol{u}(\boldsymbol{k}-\boldsymbol{n}_{\boldsymbol{u}})\} \quad (1)$$

where $\boldsymbol{y}$ is the plant output vector, $\boldsymbol{u}$ is the plant input vector, whereas n_y and n_u is the number of memory operators for plant output and input respectively. In this case, f is the transfer function of the plant that will be defined by the neural network and its weight configuration, as the result of the neural network training stage. The equation explicitly stated that the plant output vector $\boldsymbol{y}(\boldsymbol{k})$ is a function of its previous output vectors, $\boldsymbol{y}(\boldsymbol{k}\text{-}\mathbf{1})$, …, $\boldsymbol{y}(\boldsymbol{k}\text{-}\boldsymbol{n_y})$ and previous input vectors, $\boldsymbol{u}(\boldsymbol{k}\text{-}\mathbf{1})$, …, $\boldsymbol{u}(\boldsymbol{k}\text{-}\boldsymbol{n_u})$.

Meanwhile, the adopted principal for the NN-controller is the inverse control, where the controller is the inverse function of the plant, with the following NARX equation:

$$\boldsymbol{u}(\boldsymbol{k}) = f^{-1}\{\boldsymbol{u}(\boldsymbol{k}-\mathbf{1}), \ldots, \boldsymbol{u}(\boldsymbol{k}-\boldsymbol{n}_{\boldsymbol{u}}+\mathbf{1}), \boldsymbol{y}(\boldsymbol{k}+\mathbf{1}), \ldots , \boldsymbol{y}(\boldsymbol{k}-\boldsymbol{n}_{\boldsymbol{y}}+\mathbf{1})\} \quad (2)$$

In equation (2), $\boldsymbol{y}$ is the plant output, $\boldsymbol{u}$ is the plant input, whereas n_y and n_u is the number of memory operators for plant output and input respectively. In this case, f^{-1} is the inverse transfer function of the plant that is will be replaced by the trained neural network controller configuration.

III. Backpropagation Neural Controller System For A Boat Model

A. Double-Propeller Boat Model

To analyze the performance of the NN-based controller system, a double-propeller boat model without a rudder was developed [8] as depicted in Fig. 2. The diagram illustrating the main components are shown in Fig. 3, which consists of a set of double-propeller activators, a microcontroller, a compass sensor, an Inertial Measurement Unit (IMU) sensor, a radio control, a voltage regulator, and a Li-Po battery. The activators consist of two Graupner E-propellers 25-12.5 cm / 10-5" that are driven by two sets of MT-4006 T-BLDC motor and T18A T-ESC.

The developed double-propeller boat model is a MIMO system, with 2 inputs and 3 outputs. The inputs consist of the control signals of the left motor (u_1) and right motor (u_2), whereas the outputs consist of the boat's direction or heading (θ), front velocity (v_x), and side velocity (v_y). The data acquisition for the boat model is done with manual control as shown in Fig. 4. The control signals u_1 and u_2 are controlled with a radio control and the values are obtained from the controller's console screen. The heading data is obtained from the compass sensor, and the front and side velocities are derived from the accelerometer and the controller's timer.

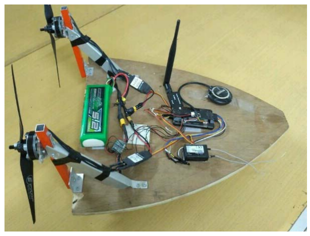

Fig. 2. Double-propeller boat model

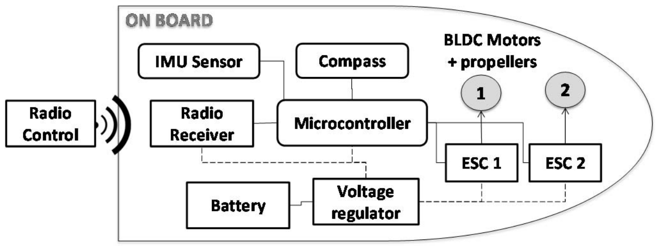

Fig. 3. The architecture of the double-propeller boat model components

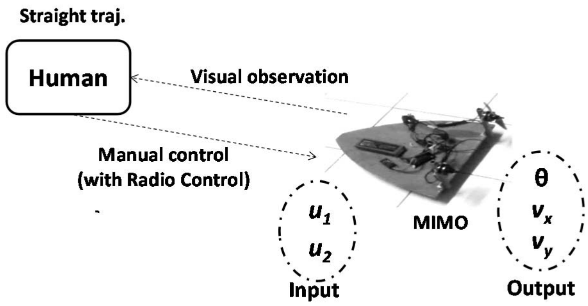

Fig. 4. Boat model data acquisition: manual control

To train the neural network, sufficient data is required. In this study, to acquire enough data, the boat model is remotely controlled to form 4 consecutive straight trajectories. An example of the first straight trajectory data acquisition is depicted in Fig. 5. The two input signals of left and right motors are shown in the upmost graph, whereas the three output signal, i.e. the direction of the boat model, is shown in the middle graph, and the value of front and side velocities, v_x and v_y, are show in the bottom graph. Since the boat model is controlled to move straight, the direction of the boat is mostly constant and the side velocity v_y is nearly *0*.

To identify the boat model, a backpropagation neural network with 15-30-3 network configuration as depicted in Fig. 6 is utilized. The NN-ID learning mechanism is adopted by using learning rate *0.2* and without momentum. After *701,595* iterations, a normalized training mean-square-error (MSE) of 2.2383×10^{-4} is obtained. Testing the NN-ID model with the real boat model input signals, the obtained normalized testing MSE is 3.679×10^{-4} as shown in Fig. 7.

This testing result shows that the NN-ID can approximate the boat model transfer function with good justification.

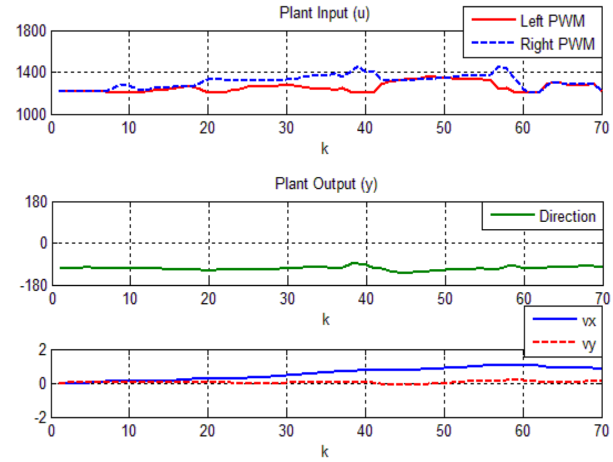

Fig. 5. The first straight trajectory data acquisition

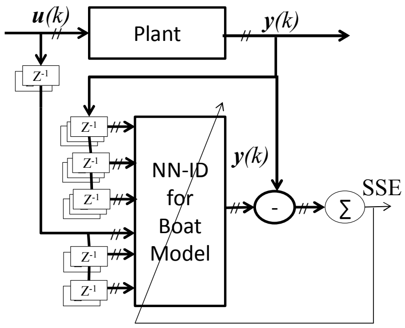

Fig. 6. Neural network identification for the boat model

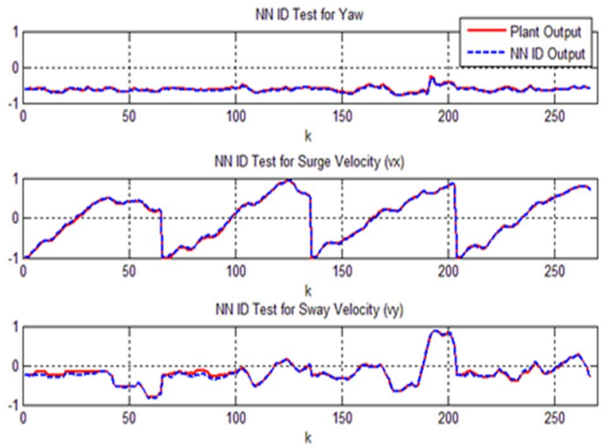

Fig. 7. Boat model NN-ID testing results (normalized)

B. BPNN Inverse Controller for The Boat Model

The NN-INV controller for boat model utilizes 21-15-2 network configuration as depicted in Fig. 8. The training is conducted by using Backpropagation (BPNN) learning mechanism with *0.01* learning rate and without using momentum. The BPNN training is continued until it reached its maximum iteration number of *2,000,000* epoch as shown in Fig. 9. The normalized training MSE value for this final training epoch is 5.2001×10^{-6}.

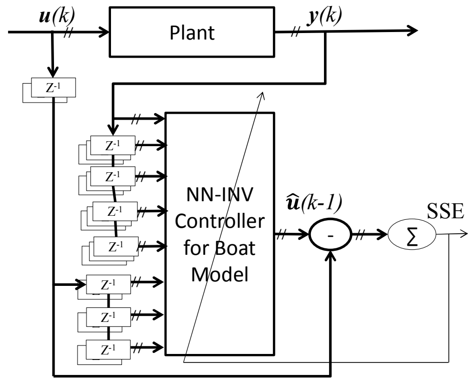

Fig. 8. Boat model BPNN inverse controller training scheme

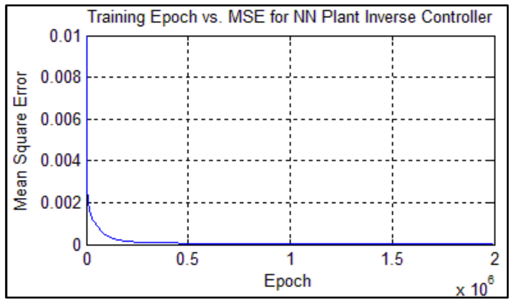

Fig. 9. Boat model BPNN inverse controller training: training epoch vs. the obtained training mean-square error (MSE)

IV. Performance Analysis of The Backpropagation Neural Controller System

A. Determination of The Unknown Initial Delayed Inputs

To compare the effect of the number of training epochs on the performance of the NN-INV controller, the controller tests are performed for the NN-INV configuration which is obtained from *2,000,000* epochs, or the final epoch, of training and that which is obtained from *200,000* epochs of training, or one-tenth of that of the first configuration. Fig. 10 and Fig. 11 shows the NN-INV controller testing results when using unknown delayed input values, i.e. the values of the delayed inputs for testing are taken from the last training data which may be highly differ from the real delayed inputs. The overall testing results are given in detail in Table 1, which shows that the NN-INV controller obtained from *2,000,000* training epoch yields a total MSE of *0.0168*, i.e. MSE = *0.0120* for u_1 and MSE = *0.0216* for u_2. Meanwhile, the total MSE for the NN-INV controller obtained from *200,000* training epochs is *0.0133*, i.e. MSE = *0.0082* for u_1 and MSE = *0.0184* for u_2. This phenomenon indicates that NN-INV controller training with too much iteration can produce overfitting, i.e. the case where the performance of the controller for the training data is very good, but the performance of the controller for the testing data is not as good as that when the training epoch is smaller.

The second test is done with the same scenario, but by giving 0 values for all of the unknown initial delayed input values. The test results are shown in Fig. 12, Fig.13 and Table

2, respectively. The obtained testing results are better than the previous one, where the MSE of the NN-INV controller obtained from *2,000,000* training epoch is *0.0133*, i.e. MSE = *0.0106* for u_1 and MSE = *0.0160* for u_2. Meanwhile, the total MSE of the NN-INV controller obtained from *200,000* training epochs is *0.0091*, i.e. MSE = *0.0113* for u_1 and MSE = *0.0068* for u_2. These results reflect that the selection of the initial values of the unknown input values affects the performance of the NN-INV controller.

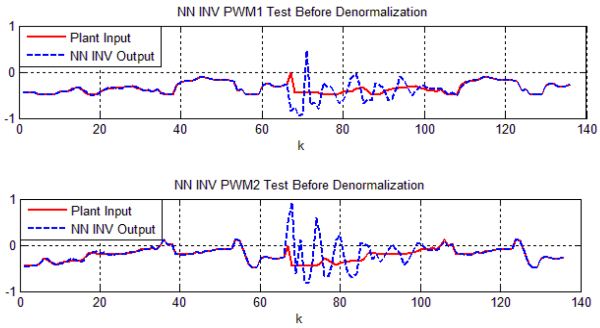

Fig. 10. Boat model NN-INV controller testing results (normalized) – first scenario: *2,000,000* training epochs

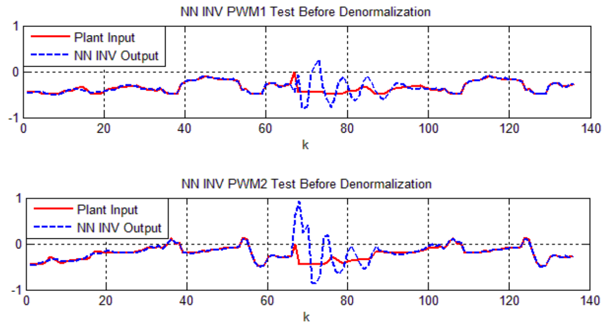

Fig. 11. Boat model NN-INV controller testing results (normalized) – first scenario: *200,000* training epochs

TABLE I. BOAT MODEL NN-INV CONTROLLER TESTING, FIRST SCENARIO

Training		**Testing MSE**			**Fig.**
Epoch	***MSE***	***total***	u_1	u_2	
2,000,000	5.20×10^{-6}	*0.0168*	*0.0120*	*0.0216*	10
200,000	1.54×10^{-4}	*0.0133*	*0.0082*	*0.0184*	11

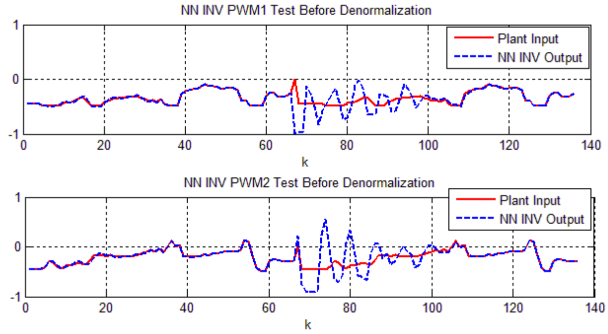

Fig. 12. Boat model NN-INV controller testing results (normalized) – second scenario: *2,000,000* training epochs

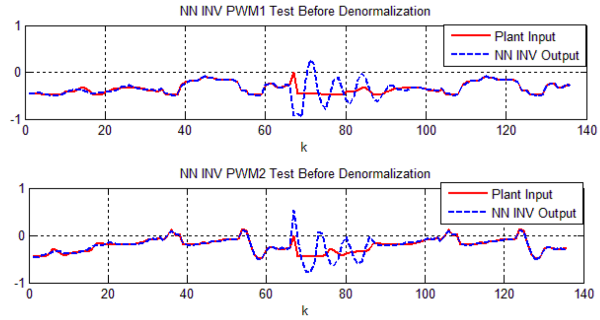

Fig. 13. Boat model NN-INV controller testing results (normalized) – second scenario: *200,000* training epochs

TABLE II. BOAT MODEL NN-INV CONTROLLER TESTING, SECOND SCENARIO

Training		**Testing MSE**			**Fig.**
Epoch	***MSE***	***total***	u_1	u_2	
2,000,000	5.20×10^{-6}	*0.0133*	*0.0106*	*0.0160*	12
200,000	1.54×10^{-4}	*0.0091*	*0.0113*	*0.0068*	13

The third test is done by taking random values from the training data as the unknown initial delayed input values. The obtained testing results are different from the first and second testing scenarios; moreover, if the testing in this scenario is repeated by using different random values, the resulting MSE will also differ. An example of a one-time testing for this third testing scenario is shown in Fig. 14, Fig.15 and Table 3. The MSE of the NN-INV controller obtained from *2,000,000* training epoch is *0.0121*, i.e. MSE = *0.0100* for u_1 and MSE = *0.0143* for u_2. Meanwhile, the total MSE of the NN-INV controller obtained from *200,000* training epochs is *0.0105*, i.e. MSE = *0.0078* for u_1 and MSE = *0.0131* for u_2.

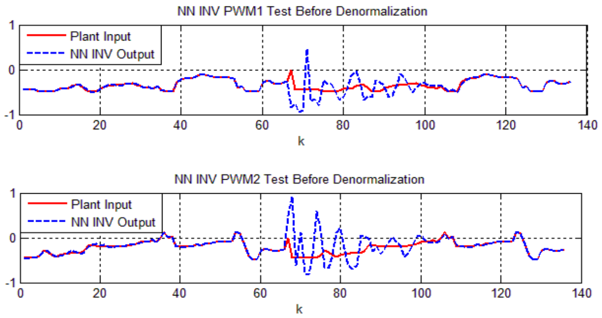

Fig. 14. Boat model NN-INV controller testing results (normalized) – third scenario: *2,000,000* training epochs

TABLE III. BOAT MODEL NN-INV CONTROLLER TESTING, THIRD SCENARIO

Training		**Testing MSE**			**Fig.**
Epoch	***MSE***	***total***	***u_1***	***u_2***	
2,000,000	5.20 x 10^{-6}	*0.0121*	*0.0100*	*0.0143*	12
200,000	1.54 x 10^{-4}	*0.0105*	*0.0078*	*0.0131*	13

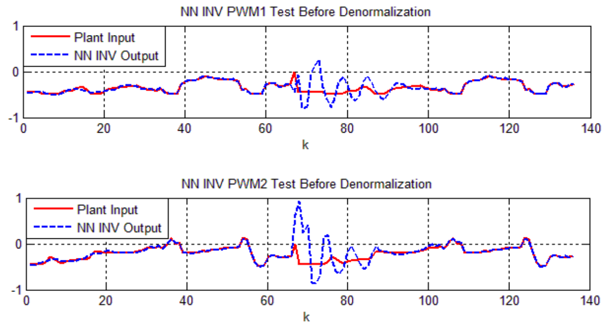

Fig. 15. Boat model NN-INV controller testing results (normalized) – third scenario: *200,000* training epochs

The simulation results in Table 1, Table 2 and Table 3 indicate that the testing error value or MSE for NN-INV controller configuration with *2,000,000* training epoch is higher than that for NN-INV controller configuration with *200,000* training epoch. This fact indicates the occurrence of overfitting when the number of training epoch is too high. Despite the number of training epoch, the control errors are also affected by the selection of initial values of the delayed inputs on the NN-INV controller. Although taking random values from the training data may give a better control performance, in the next subsection, all of the unknown initial values are set to 0 to guarantee the comparability of the testing results.

B. Settling Time and Control Error Analysis of The BPNN Direct Inverse Control System

In this section, the designed BPNN-controller is tested in an open-loop Direct Inverse Control (DIC) scheme. Various BPNN-INV configurations which are obtained from different values of training epochs are implemented in DIC systems and then analyzed in terms of control error and settling time. Fig. 16 shows that the DIC control error initially decreases as the training epoch increases, but after *100,000* training epoch, both the DIC control error and the settling time increases as the epoch increases. This trend indicates the occurrence of overfitting phenomena. When overfitting happens, the control error is highly affected by the required settling time, as the errors are generally observed at the beginning of the control process when the system is not yet stable.

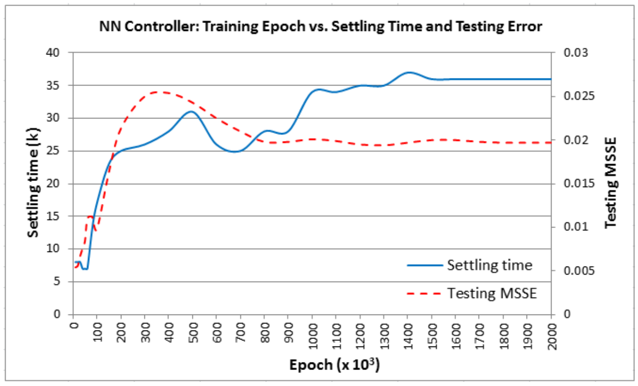

Fig. 16. BPNN-controller: training epoch vs DIC control error and settling time

The analysis as shown in Fig. 16 suggests that the optimum BPNN-controller configuration is the one that is obtained when the training iteration is *99,899* epochs. The training MSE for this iteration is 4.4500×10^{-4}. The DIC system testing results can be seen on Fig. 17, where the resulted testing MSE is *0.0098*, i.e. MSE of direction is *0.0008*, MSE of v_x is *0.0009*, and MSE of v_y is *0.0277*.

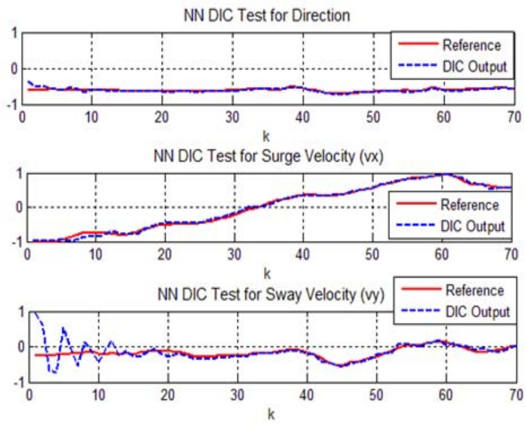

Fig. 17. Boat model DIC testing results (normalized)

V. CONCLUSION

The performance of backpropagation controller system for an autonomous double-propeller boat model have been tested and analyzed by using direct inverse control scheme. This study also investigates the performance of the control system in terms of its settling time, other than the commonly analyzed control error. It is revealed that for the proposed system, the control error also depends on the required settling time, especially when the training epoch is too high (overfitting). An adaptation of fine-tuning method to decrease the settling time and reduce the system control error is still under investigation.

ACKNOWLEDGMENT

This work was partly supported by *Penelitian Disertasi Doktor* (PDD) research fund from The Ministry of Research and Higher Education of The Republic of Indonesia through Universitas Indonesia and Institut Pertanian Bogor.

REFERENCES

[1] T. I. Fossen and A. Grovlen, "Nonlinear output feedback control of dynamically positioned ships using vectorial observer backstepping," *IEEE Trans. Control Syst. Technol.*, vol. 6, no. 1, pp. 121–128, Jan. 1998.

[2] T. Temel and H. Ashrafiuon, "Sliding-mode speed controller for tracking of underactuated surface vessels with extended Kalman filter," *Electron. Lett.*, vol. 51, no. 6, pp. 467–469, 2015.

[3] D. Chwa, "Global tracking control of underactuated ships with input and velocity constraints using dynamic surface control method," *IEEE Trans. Control Syst. Technol.*, vol. 19, no. 6, pp. 1357–1370, 2011.

[4] N. E. Kahveci and P. A. Ioannou, "Adaptive steering control for uncertain ship dynamics and stability analysis," *Automatica*, vol. 49, no. 3. pp. 685–697, 2013.

[5] S. D. Lee, C. Y. Tzeng, Y. Z. Kehr, C. C. Huang, and C. K. Kang, "Autopilot System Based on Color Recognition Algorithm and Internal Model Control Scheme for Controlling Approaching Maneuvers of a Small Boat," *IEEE J. Ocean. Eng.*, vol. 35, no. 2, pp. 376–387, Apr. 2010.

[6] L. P. Perera and C. Guedes Soares, "Lyapunov and Hurwitz based controls for input-output linearisation applied to nonlinear vessel steering," *Ocean Eng.*, vol. 66, pp. 58–68, 2013.

[7] R. S. Burns, "The use of artificial neural networks for the intelligent optimal control of surface ships," *IEEE J. Ocean. Eng.*, vol. 20, no. 1, pp. 65–72, Jan. 1995.

[8] K. Priandana, W. Wahab, and B. Kusumoputro, "Comparison of neural networks based direct inverse control systems for a double propeller boat model," in *Proceedings of The V International Conference on Network, Communication and Computing (ICNCC 2016)*, 2016, pp. 310–315.

[9] K. P. Tee and S. S. Ge, "Control of fully actuated ocean surface vessels using a class of feedforward approximators," *IEEE Trans. Control Syst. Technol.*, vol. 14, no. 4, pp. 750–756, 2006.

[10] K. Priandana and B. Kusumoputro, "Development of Self-Organizing Maps neural networks based control system for a boat model," *J. Telecommun. Electron. Comput. Eng.*, vol. 9, no. 1–3, 2017.

[11] S. L. Dai, C. Wang, and F. Luo, "Identification and learning control of ocean surface ship using neural networks," *IEEE Trans. Ind. Informatics*, vol. 8, no. 4, pp. 801–810, 2012.

[12] B. Kusumoputro, K. Priandana, and W. Wahab, "System identification and control of pressure process rig® system using backpropagation neural networks," *ARPN J. Eng. Appl. Sci.*, vol. 10, no. 16, pp. 7190–7195, 2015.

[13] P. J. Werbos, "Neural networks for control and system identification," *Proc. 28th IEEE Conf. Decis. Control.*, pp. 260–265, 1989.

[14] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks.," *IEEE Trans. Neural Netw.*, vol. 1, no. 1, pp. 4–27, 1990.