Development of Self-Organizing Maps Neural Networks Based Control System for a Boat Model

Karlisa Priandana[1,2], Benyamin Kusumoputro[1]
[1]*Computational Intelligence and Intelligent Systems Research Group,*
Department of Electrical Engineering, Faculty of Engineering,
Universitas Indonesia, Kampus Baru Universitas Indonesia, Depok, West Java, Indonesia.
[2]*Department of Computer Science,*
Faculty of Mathematics and Natural Sciences, Bogor Agricultural University,
Kampus IPB Dramaga, Bogor, West Java, Indonesia.
kusumo@ee.ui.ac.id

***Abstract*—This paper describes the development of a controller system for a developed double-propeller boat model using the unsupervised learning neural networks, namely the Self-Organizing Maps (SOM). The performance characteristics of the proposed SOM-based controller are then compared with that of the well-known Back-propagation Neural Networks (BPNN)-based controller through a direct inverse control scheme. Experimental results showed that the SOM-based controller can produce a low error, even lower than that of the widely used BPNN-based controller. Furthermore, the computational cost of the SOM-based controller is found to be more than 700 times faster than that of the BPNN-based controller. These findings suggest that the utilization of the proposed SOM-based controller for the control of a boat is highly effective.**

***Index Terms*—Artificial Neural Network; Direct Inverse Control; Double-Propeller; USV.**

I. Introduction

The control of autonomous Unmanned Surface Vehicles (USV) is a very challenging topic since its working environment is usually dynamic, complex, and unstructured. As its actual dynamics is highly non-linear, time-varying, and coupled [1], it may be too complex to design the USV controller using a mathematical approach. The Internal Model Control [2] [3], for example, was obviously derived from a mathematical simplification approach, and may not perform well when it is used as a USV controller in a highly nonlinear environment.

To minimize the use of mathematical assumptions in the USV modeling, and at the same time, to reduce the efforts required to derive an accurate mathematical model, artificial neural networks (ANN) based controllers for USV have been widely proposed. The methods are generally designed for solving the unstable ship dynamics problems, and are mainly developed based on the supervised learning algorithms, such as Back-propagation [4], Neural Network Model Reference Adaptive Controller (NN-MRAC) [5], ANN which is combined with back-stepping technique [6], and a Radial-Basis Function neural networks [7].

Back-propagation Neural Networks (BPNN)-based controller is the most widely used ANN-based controller due to its simple but powerful structure. This method is proven to be able to produce a very low error [8]. However, the iterative procedure during its learning is quite time consuming, especially when the data and the chosen number of neurons in its network is large.

Self-Organizing Maps (SOM) is introduced by Kohonen [9] and categorized as an unsupervised learning mechanism. Its main functions are reducing the dimensions of data and display data similarities. Some advantages of SOM are its simplicity, fast computation, and easy evaluation. However, it is commonly used for clustering and classification. The utilization of SOM to approximate the dynamical input–output mappings was first introduced as *Vector-Quantized Temporal Associative Memory* (VQTAM) model [10], and it was further developed as autoregressive SOM (ARSOM) model [11]. However, the discussion on VQTAM and ARSOM was only focusing on the possibilities to use the methods through some mathematical models and simulations, and therefore, lack of empirical analysis on any real experimental system.

Using the background knowledge of VQTAM, in this paper, SOM is developed as a neural controller system for a USV. As to design a controller using a neural network requires sufficient information of the input and output data, empirical data obtained from a boat model as a simulation of a real double-propeller USV system is utilized. We then derived a Multiple-Input and Multiple-Output (MIMO) system controller using the unsupervised learning SOM neural networks using direct inverse control scheme. The performance characteristics of the developed SOM-based controller is then analyzed and compared with that of the commonly used BPNN-based controller, especially in terms of error and computational cost.

The organization of this paper is as follows. First, the developed boat model and the data acquisition system are described in section 2. Section 3 presents the concept of the direct inverse control used in our system together with the SOM-based controller for a MIMO boat model. The experiment and the characteristic comparison of the SOM-based controller with that of the BPNN-based controller is written in section 4, and finally, the paper is concluded in section 5.

II. The Boat model

The developed boat model is an unmanned vessel model that can be operated on the ground to mimic an unmanned double-propeller boat system. The purpose of using this boat model is to obtain the most ideal environment but not considering the effect of the ocean waves and currents. The realization of the boat model and the block diagram of the components are shown in Figure 1 and Figure 2, respectively. The main components of the boat model are: two T18A T-ESC, two MT-4006 T-BLDC motors, two Graupner E-propellers 25-12.5 cm / 10-5", a microcontroller, a compass sensor, an Inertial Measurement Unit (IMU) sensor that consist of gyroscope, accelerometer and barometer, a radio control, a voltage regulator, and a Li-Po battery, respectively.

Figure 1: The boat model

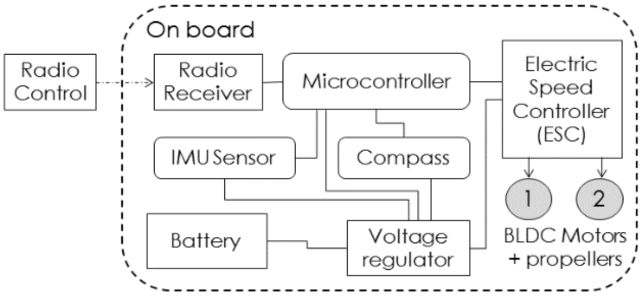

Figure 2: Block diagram of boat model's components

The developed double-propeller boat model is a MIMO system with two inputs and three outputs. The inputs consist of two control signals for the left motor (*PWM1*) and the right motor (*PWM2*). The outputs of the boat model consist of the boat's direction (*yaw*), the front/surge velocity (v_x), and the side/sway velocity (v_y). The data concerning with the boat model is acquired through a radio control system as the manual controller, which can be shown in Figure 3. The output data of the system that consists of the yaw data is obtained from the compass sensor, while the front and side velocities are derived from the accelerometer and the controller's timer.

For the preliminary experiment, the boat model is manually controlled for forward motion (straight trajectory) for four times, consecutively, and circular motion (circular trajectory). As a greater thrust for the boat model is necessary in the beginning of the movement, the data for simulation of the continuous movement of the boat model is taken after the boat model has started to move. The example of data used for the neural network training is shown in Figure 4. The upper graph plots the two control signals for left and right motors/propellers that are controlled by a human through a remote control in order to keep the course straight. The middle graph shows the direction of the boat in the inertial frame, obtained from the compass sensor. As can be seen from Figure 4(a), the direction of the boat is relatively constant, which reflects that the boat model is moving straight. Whereas in Figure 4(b), the direction of the boat is changing within 360º, which reflects that the boat model is constantly turning. The lower graph depicts the values of surge and sway velocities, v_x and v_y. The value of sway velocity is nearly zero, which means that the boat is not moving to the either side, whereas the value of surge velocity is constantly increasing due to the accelerations from the two rotating propellers.

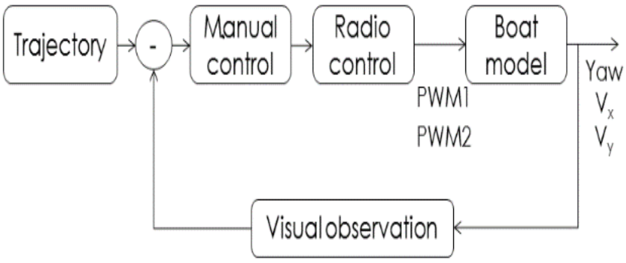

Figure 3: Data acquisition with manual control

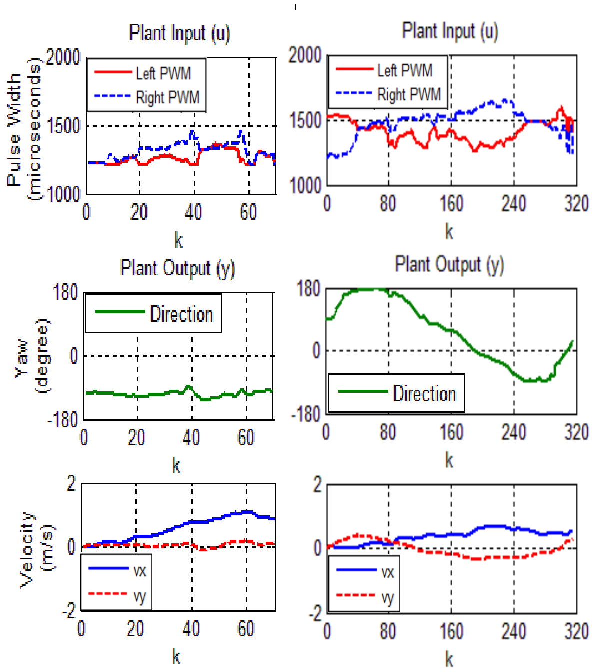

Figure 4: Boat model data for neural networks learning

III. The Proposed Neural Networks Based Controller Design

A. Direct Inverse Control

The neural networks based controller design utilizes the open-loop direct inverse control as depicted in Figure 5 [8]. In this scheme, the inverse neural controller is directly cascaded with the controlled plant to provide an identity mapping between the desired system's output (signal reference) and the actual plant output or the plant response. Since the neural controller that is applied directly on the system during its learning stage may disturb the plant, the use of a model to represent the plant is recommended. In this work, the plant is identified by using a neural network with back-propagation learning mechanism.

B. Neural Networks Based System Identification

The approximation of plant identification is done by adopting the Nonlinear Auto Regressive with eXogenous input (NARX) [12] model structure for MIMO system, as expressed in the Equation (1).

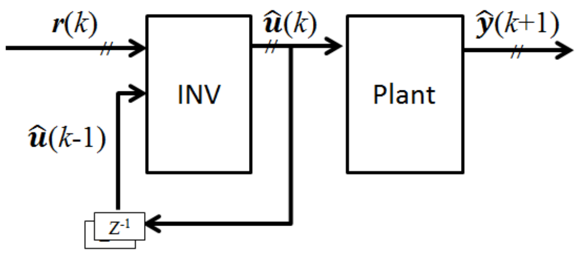

Figure 5: Open-loop Direct Inverse Control scheme

$$\boldsymbol{y}[k] = f(\boldsymbol{y}[k-1], \dots, \boldsymbol{y}[k-n_y], \boldsymbol{u}[k-1], \dots, \boldsymbol{u}[k-n_u]) \quad (1)$$

where $\boldsymbol{y}$ is the plant outputs, $\boldsymbol{u}$is the plant inputs, n_y and n_u are the number of memory operators for each plant output and input, respectively. In this case, *f*is the transfer function of the plant that will be replaced by the neural networks. This equation explicitly stated that the plant outputs $\boldsymbol{y}[k]$ is a function of its previous outputs, $\boldsymbol{y}[k\text{-}1]$, …, $\boldsymbol{y}[k\text{-}n_y]$ and the previous inputs, $\boldsymbol{u}[k\text{-}1]$, …, $\boldsymbol{u}[k\text{-}n_u]$.

The ANN-based system identification consists of one input layer, one hidden layer, and one output layer with *15*, *30*, and *3* neurons, respectively, as depicted in Figure 6. As the neurons used a bipolar Sigmoid activation function, the data should be normalized into the range of *-1* and *+1* prior to the training stage. Back-propagation learning mechanism is adopted for training the neural networks, with the learning rate of *0.2*. For straight trajectory, the training requires *701,595* iterations to converge with the training mean-sum-square error (MSSE) of 2.2383×10^{-4}, while for the testing phase, the MSSE is 3.679×10^{-4}. For the circular trajectory, identification training requires *1,000,000* iterations to obtain a training MSSE of 2.1301×10^{-5}, whereas for the testing phase, the resulted MSSE is 9.88×10^{-6} as shown in Figure 7.These low errors show that the BPNN-based system identification successfully mimics the transfer function of the real plant.

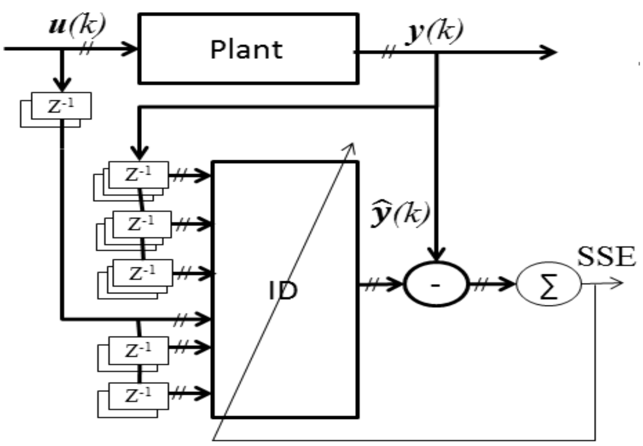

Figure 6: Boat model identification using ANN

C. SOM-Based Control System

The utilization of ANN for controller was adopted as an inverse control system scheme [13], where the controller acts as the inverse of the plant through:

$$\boldsymbol{u}[k] = f^{-1}(\boldsymbol{u}[k-1], \dots, \boldsymbol{u}[k-n_u+1], \boldsymbol{y}[k+1], \dots, \boldsymbol{y}[k-n_y+1]) \quad (2)$$

where the nomenclature of Equation (2) is the same with that of Equation (1), but f^{-1} is the inverse transfer function of the plant.

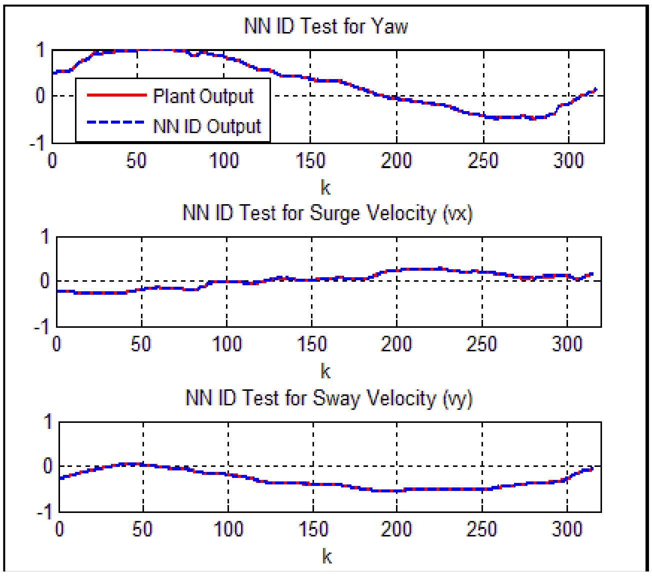

Figure 7: ANN-based identification for circular trajectory (MSSE = 9.88×10^{-6})

SOM was initially used for static input-output mappings, which means that the current output depends solely on the current input. For the inverse control scheme in Equation (2), however, the plant inputs $\boldsymbol{u}[k]$ is a function of its previous inputs $\boldsymbol{u}[k\text{-}1]$, …, $\boldsymbol{u}[k\text{-}n_u\text{+}1]$, expected outputs $\boldsymbol{y}[k\text{+}1]$ and previous outputs $\boldsymbol{y}[k]$, …, $\boldsymbol{y}[k\text{-}n_y\text{+}1]$, which will be replaced by the reference signal in the direct inverse control scheme.

For this purpose, some modifications to the original Kohonen SOM algorithmare required [10]. To be used as a neural networkscontrol system, the input vector of SOM is augmented into:

$$\boldsymbol{x}[k] = \begin{pmatrix} \boldsymbol{x}^{in}[k] \\ \boldsymbol{x}^{out}[k] \end{pmatrix} \quad (3)$$

where:

$$\boldsymbol{x}^{in}[k] = \boldsymbol{y}[k+1], \boldsymbol{y}[k], \dots, \boldsymbol{y}[k-n_y+1], \boldsymbol{u}[k-1], \dots, \boldsymbol{u}[k-n_u+1] \quad (4)$$

$$\boldsymbol{x}^{out}[k] = \boldsymbol{u}[k] \quad (5)$$

The vector $\boldsymbol{x}^{in}[k]$ in Equation (4) contains the input data of the dynamic mapping being learned. Meanwhile, vector $\boldsymbol{x}^{out}[k]$ is the desired output of this mapping, which is actually the plant input, $\boldsymbol{u}$[k]. The reference vectors or weights of neurons, $\boldsymbol{v}$[k], are also augmented accordingly as shown in Figure 8.

During training, the winning neuron l^* at time k is decided solely from the smallest Euclidean distance between $\boldsymbol{x}^{in}[k]$ and $\boldsymbol{v}_l^{in}[k]$. Then, both winning reference vectors, which are the reference vectors with the same index as the winning neuron l^*, $\boldsymbol{v}_{l*}^{in}[k]$ and $\boldsymbol{v}_{l*}^{out}[k]$, are updated. Similarly, on the testing stage, the winning neuron l^* is obtained from $\boldsymbol{x}^{in}[k]$ and $\boldsymbol{v}_l^{in}[k]$, whereas the resulted control signal is similar to the output reference vector:

$$\boldsymbol{u}[k] \cong \boldsymbol{v}_{l*}^{out}[k] \tag{6}$$

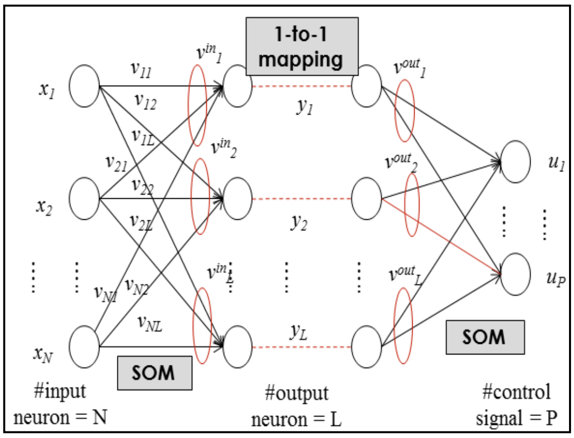

Figure 8: The architectural structure of SOM-DIC system

IV. Experimental Analysis

Offline experimental analysis on the developed double-propeller boat model was conducted to compare the performance characteristics of the proposed Self-Organizing Maps (SOM)-DIC system and the well-known back-propagation neural network (BPNN)-DIC system.

The chosen numbers of memory operators of input and output data for both controllers are $n_u = 4$ and $n_y = 3$. Therefore, there are *7* elements in $\boldsymbol{x}^{in}[k]$ to represents each output data (e.g. direction, surge velocity and sway velocity), resulting *21* elements in total. This number represents the number of input neurons in both ANN-based controllers. Meanwhile, the number of elements in $\boldsymbol{x}^{out}[k]$ for the SOM-based controller or the number of output neurons in BPNN-based controllers depends solely on the number of plant input data. For the double-propeller USV, which is the case of interest, there are two inputs that drive the motor to propel (e.g. *PWM1* and *PWM2*).

A. SOM-Based Controller

The SOM-DIC systems are developed by utilizing*21* input neurons with *10*, *30*, and *66* output neurons for straight trajectory, and *79*, *158*, and *316* output neurons for circular trajectory, respectively. Different numbers of output neurons are empirically used to analyze their effects on the proposed controller. The initial learning rate is set to be*0.9* and the learning rate reduction factor is set to be *0.9*. The number of training iteration is limited to *131*, which requires very low computational costs, i.e., less than *1* second for the straight trajectory, and less than *4* seconds for the circular trajectory.

The SOM-DIC system is then tested on direct inverse control scheme (see Figure 5) and the experimental results for the straight trajectory are shown in Figure 9 to 11, for *10*, *30*, and *66* output neurons, respectively. The obtained MSSE for SOM based controller with *10* output neurons is *0.0048*, with MSEs *0.0006* for yaw, *0.0013* for v_x, and *0.0125* for v_y, respectively. The MSSE for *30* output neurons is *0.0040*, with MSEs *0.0005* for yaw, *0.0006* for v_x, and *0.0110* for v_y, respectively. Meanwhile, the MSSE for *66* output neurons is *0.0042*, with MSEs *0.0005* for yaw, *0.0003* for v_x, and *0.0117* for v_y, respectively.As can be seen in the bottom graph of each figures, compare with that of the surge velocity (v_x), the error of the sway velocities (v_y) for all of the three SOM-DIC system configurations are higher.However, by observing the upper most graph of these figures, the boat model is moving straight with a very small MSE for yaw direction with relatively lower settling time, showing that the SOM-DIC systems are in good agreement with the reference signals.

The three figures also reflect that the more number of output neurons is utilized, the better characteristics performance of the SOM-DIC is achieved, especially, in terms of keeping the boat model's direction and surge velocity, which are the main focus of this particular case.

The experimental results for the circular trajectory are shown in Figure 12 to 14, for 79, 158, and 316 output neurons, respectively. The obtained MSSE for SOM based controller with 79 output neurons is 1.0835 x 10^{-4}, with MSEs 1.176 x 10^{-4} for yaw, 0.561 x 10^{-4} for v_x, and 1.513 x 10^{-4} for v_y, respectively. The MSSE for 158 output neurons is 4.8784 x 10^{-5}, with MSEs 1.397 x 10^{-5} for yaw, 4.240 x 10^{-5} for v_x, and 2.020 x 10^{-5} for v_y, respectively. Meanwhile, the MSSE for 316 output neurons is 9.8792 x 10^{-6}, with MSEs 1.115 x 10^{-5} for yaw, 1.208 x 10^{-5} for v_x, and 6.41 x 10^{-6} for v_y, respectively. These results further justify that the proposed controller can control the boat model's direction and surge velocity according to the given reference signal, and that larger number of output neurons will result in better characteristics performance of the SOM-DIC.

B. BPNN-Based Controller

As a comparison, the same data from this double-propeller boat model is also used to train the back-propagation neural network (BPNN)-DIC system with a *21-15-2* network configuration [8]. The number of the input neuron is set to *21*, for a comparability of the characteristics performance analysis with that of the SOM-DIC system.

Backpropagation learning algorithm is used with 0.01 learning rate and without momentum. For the straight trajectory, the training requires 99,899 iterations in 672.06 seconds to converge with a MSSE of 4.45 x 10^{-4}. The converged connection weights from the training stage are then applied to the neural network direct inverse control scheme and the result of the control performance experiment is depicted in Figure 15. As shown in the figure, the MSSE of the boat model by using the BPNN-DIC system is9.8 x $10^{-3}$with MSE for each parameter is8 x 10^{-4} for yaw, 1 x 10^{-3} for v_x, and 2.77 x 10^{-2} for v_y, respectively.

Meanwhile, for the circular trajectory, the training requires 100,000 iterations in 1482.03 seconds to converge with a MSSE of 5.31 x 10^{-4}. Utilizing the obtained weights to the neural network direct inverse control scheme, the control performance experiment is shown in Figure 16. As can be seen from the figure, the MSSE is6.1316 x $10^{-4}$with MSE for each parameter is15 x $10^{-4}$for yaw, 1 x 10^{-4} for v_x, and 2 x 10^{-4} for v_y, respectively.

Table 1 and Table 2 shows the overall comparison of BPNN-based controller and SOM-based controller on the boat model direct inverse control scheme, for the straight and circular trajectories, respectively. It can be seen that the SOM-based controller requires a much lower computation time, e.g. less than *1* second for the straight trajectory and less than *4* seconds for the circular trajectory, compared to the BPNN-based controller which requires more than *11* minutes for its straight trajectory training and more than *24* minutes for circular trajectory training. Furthermore, the testing error of SOM-based controller is generally lower than that of the BPNN-based controller.

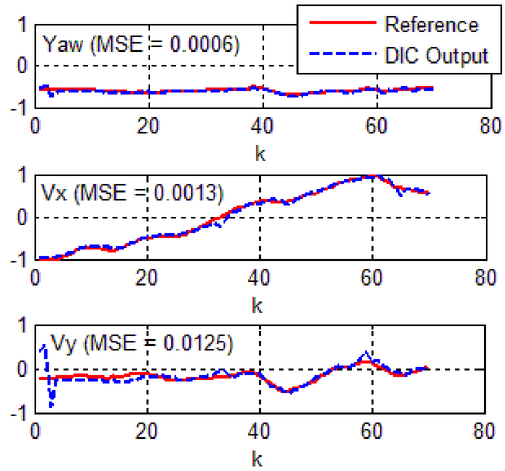

Figure 9: SOM-DIC system with 21-10-10-2 network configuration, straight traj. (MSSE = 0.0048)

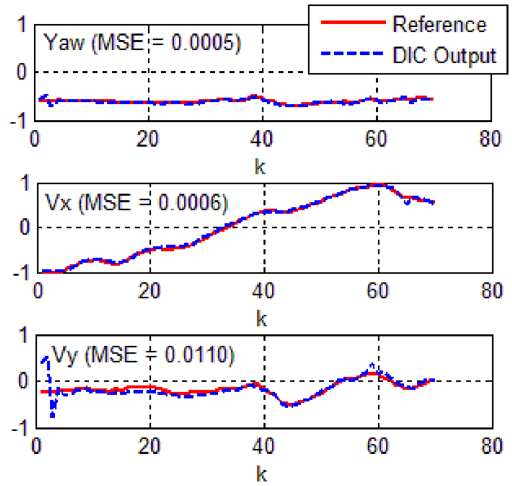

Figure 10: SOM-DIC system with 21-30-30-2 network configuration, straight traj. (MSSE = 0.0040)

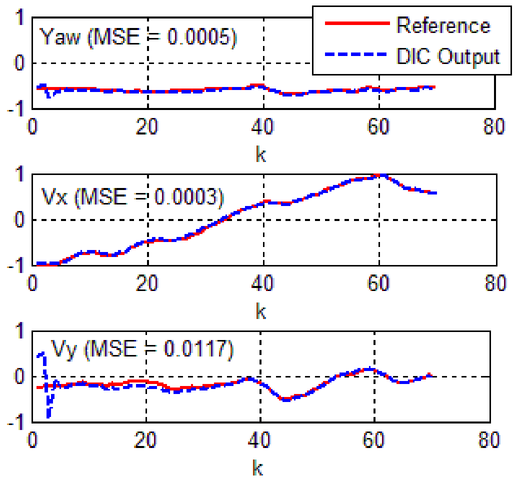

Figure 11: SOM-DIC system with 21-66-66-2 network configuration, straight traj. (MSSE = 0.0042)

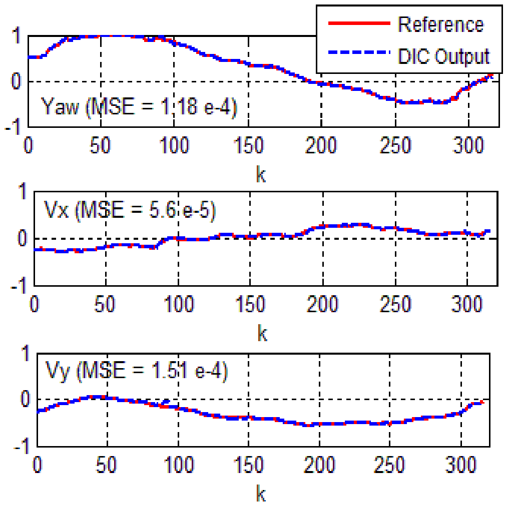

Figure 12: SOM-DIC system with 21-79-79-2 network configuration, circular traj. (MSSE = 1.08 x 10^{-4})

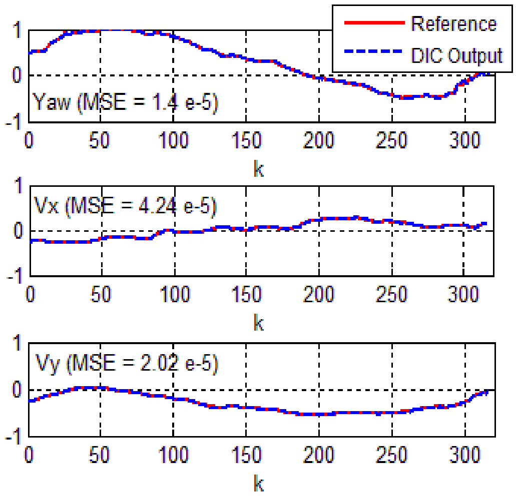

Figure 13: SOM-DIC system with 21-158-158-2 network configuration, circular traj. (MSSE = 4.9 x 10^{-5})

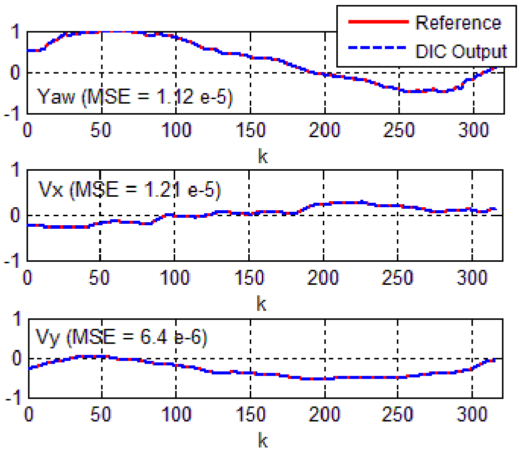

Figure 14: SOM-DIC system with 21-316-316-2 network configuration, circular traj. (MSSE = 9.9 x 10^{-6})

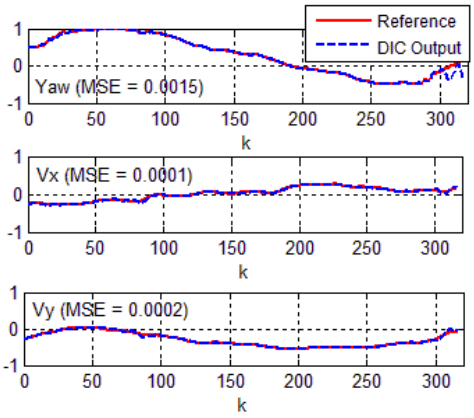

Figure 15: BPNN-DIC system with 21-15-2 network configuration, straight trajectory (MSSE = 0.0098)

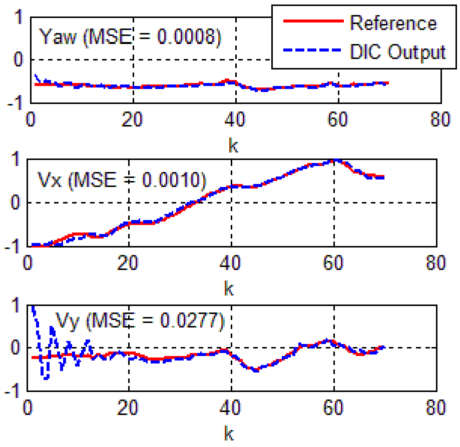

Figure 16**:** BPNN-DIC system with 21-15-2 network configuration, circular trajectory (MSSE = 6.13 x 10^{-4})

Table 1
Performance Comparison of the Boat Model Controllers on the Direct Inverse Control Scheme, for Straight Trajectory

	SOM 21-10-10-2	SOM 21-30-30-2	SOM 21-66-66-2	BPNN 21-15-2
Comp. time (seconds)	0.57	0.66	0.87	672.06
Testing MSSE	0.0048	0.0040	0.0042	0.0098
MSE yaw	0.0006	0.0005	0.0005	0.0008
MSE v_x	0.0013	0.0006	0.0003	0.0010
MSE v_y	0.0125	0.0110	0.0117	0.0277

Table 2
Performance Comparison of the Boat Model Controllers on the Direct Inverse Control Scheme, for Straight Trajectory

	SOM 21-79-79-2	SOM 21-158-158-2	SOM 21-316-316-2	BPNN 21-15-2
Comp. time (seconds)	1.734	2.46	3.46	1482.03
Testing MSSE (x 10^{-4})	1.08	0.488	0.099	6.13
MSE yaw	1.18	0.14	0.112	15
MSE v_x	0.56	0.424	0.121	1
MSE v_y	1.51	0.202	0.064	2

V. Conclusion

A neural controller system based on the unsupervised learning mechanism Self-Organizing Maps (SOM) neural networks has been successfully used as the controller of a double-propeller boat model using direct inverse control scheme. Experimental analysis has proven that the SOM-based controller can produce a low error, even lower than that of the widely used back-propagation-based controller. From our research results, it is clearly seen that the higher number of the SOM output neurons will produce lower error. The significant advantage of SOM-based controller lies on the much lower computational time during training due to the fewer numbers of iterations required for convergence.

Acknowledgment

This research was supported by the Universitas Indonesia Research Grant No. 1870/UN2.R12/HKP.05. 00/2015.

References

[1] Fossen, "T.I. Marine Control Systems: Guidance, Navigation, and Control of Ships, Rigs and Underwater Vehicles", Trondheim, Norway: Marine Cybernetics, 2002.

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

[3] Tzeng, C.-Y. "An internal model control approach to the design of yaw-rate-control ship-steering autopilot",*IEEE J. Oceanic Eng*., Vol. 24, No. 4, pp.507-513, 1999.

[4] Burns, R.S.,"The use of artificial neural networks for the intelligent optimal control of surface ships", *IEEE J. Oceanic Eng*., Vol. 20, No.1, pp.65–72, 1995.

[5] Leonessa, A., VanZwieten, T., Morel, Y.,"Neural network model reference adaptive control of marine vehicles", *Current Trends in Nonlinear Systems and Control*, IV, pp.421–440, 2006.

[6] Tee, K P., Ge, S.S.,"Control of fully actuated ocean surface vessels using a class of feedforward approximators", *IEEE Trans. on Control Syst. Technol*., 14(4), pp.750–756, 2006.

[7] Dai, S.-L., Wang, C., Luo, F., "Identification and learning control of ocean surface ship using neural networks", *IEEE Trans. on Industrial Informatics*, Vol. 8. No.4, pp.801-810, 2012.

[8] Kusumoputro, B., Priandana, K., Wahab, W.,"System Identification and Control of Pressure Process Rig System using Backpropagation Neural Networks", *ARPN Journal of Engineering and Applied Sciences*, Vol. 10, No. 6, pp.7190-7195, 2015.

[9] Kohonen, T.,"The self-organizing map",*Proceedings of the IEEE*, Vol. 78, No.9, pp.1464-1480, 1990.

[10] Barreto, G.de.A., Araujo, A.F.R.,"A self-organizing NARX network and its application to prediction of chaotic time series", *Proc. Int. Joint Conf. on Neural Networks*, Vol. 3, pp.2144-2149, 2001.

[11] Barreto, G.de.A., Araújo, A.F.R.,"Identification and Control of Dynamical Systems Using the Self-Organizing Map",*IEEE Trans. on Neural Networks*, Vol. 15, No. 5, pp.1244-1259, 2004.

[12] Sahoo, H.K., Dashb, P.K., Rathc, N.P.,"NARX model based nonlinear dynamic system identification using low complexity neural networks and robust H∞ filter", *Appl. Soft Comput. J*, Vol. 13, No. 7, pp.3324-3334, 2013.

[13] Narendra, K.S., Parthasarathy, K.,"Identification and control of dynamical systems using neural networks",*IEEE Trans. Neural Networks*, Vol. 1, pp.4-27, 1990.