SOM-Based Direct Inverse Trajectory Control System for Double-Propeller Boat Maneuvers

Karlisa Priandana¹,², Member, IEEE, Benyamin Kusumoputro¹

¹Department of Electrical Engineering, Faculty of Engineering, Universitas Indonesia, Depok 16424 Indonesia
²Department of Computer Science, Faculty of Mathematics and Natural Sciences, IPB University (Bogor Agricultural University), Bogor 16680 Indonesia

Corresponding author: Benyamin Kusumoputro (e-mail: kusumo@ee.ui.ac.id).

ABSTRACT This paper presents the development of neural-network-based control system using self-organizing-maps (SOMs) for the maneuvers of a double-propeller boat. The performance characteristics of the developed SOM controller system are compared with a widely-used supervised learning mechanism, the backpropagation neural network (BPNN) controller. The experimental results show that the proposed unsupervised SOM controller can control the boat model with very low error, although most artificial neural network (ANN)-based controllers are usually designed using supervised learning approaches. The important characteristic of the proposed SOM controller system is that it utilizes a mapping principle instead of an error calculation such as that in the BPNN controller system; consequently, the proposed SOM controller system is not very sensitive to non-ideal training data, which produces a low control error for the generated elliptical trajectory data. It is also confirmed in these experiments that when more mapping neurons are utilized in the SOM controller, a lower control error is achieved. It is expected that in a real implementation, the SOM controller could provide more robust control than the BPNN controller in handling small disturbances such as light winds and small waves.

INDEX TERMS Backpropagation, Computational and artificial intelligence, Control systems, Neural networks, Self-organizing maps.

I. INTRODUCTION

In recent years, the autonomous control system of unmanned surface vehicles (USVs) has gained significant attention due to the importance of USVs in various applications, such as military, marine mapping, sea surveillance and aquatic data acquisition. The challenges of USV autonomous control are due to the nonlinear dynamics of the USV and the unstructured working environment in which the USV operates [1]. In theory, the USV dynamics are highly coupled, time-varying and nonlinear, while winds, water currents and ocean waves create an unstable and unstructured USV working environment, adding more complexity to the problem.

Conventional proportional-integral-derivative (PID) controllers are used in many autonomous control applications, including USV applications, due to their simple structure, ease of design, and low cost. However, these controllers cannot provide perfect control performance and, at the same time, cannot guarantee operation with same level of accuracy for the entire operating range [2]. To address the nonlinear effects, various controllers based on other mathematical models, such as sliding-mode control [3] [4], back-stepping techniques [5] [6], linear quadratic control [7], surface control [8], fuzzy PID control [9], internal model control [10] [11], and active disturbance rejection control (ADRC) [12], have been developed. However, to reduce the complexity of the problem, these mathematical-model-based control systems, including PID controllers, require some considerable assumptions for the simplifications and linearization, as stated in [13] and [14]. Furthermore, the model may not be able to represent the entire plant characteristics and properties without correct tuning parameters, as explained in [15] and [16], so that the approximation results may exhibit un-modeled dynamics due to the changing parameters [17]. Therefore, the designed controller may not perform well when it is utilized to control the USV in a highly nonlinear environment.
The design of controller using supervised machine learning methods has been proposed to solve unstable ship dynamical problems. A method for controlling a USV using the concept of the support vector machine (SVM) was implemented in [18-20] based on the work proposed in [21]. Other increasingly popular and promising methods are based on the artificial neural network (ANN) approach. Compared to the SVM, the structure of an ANN is much simpler, so the online training of then ANN is easier and expected to be faster than the online training for the SVM. Furthermore, in some cases, an ANN-based controller is proven to be more successful in terms of disturbance rejection than other machine learning methods, such as fuzzy logic [22], although some more recent works have developed fuzzy control for nonlinear Markov jump systems [23] [24]. Some USV controllers based on ANNs include the backpropagation controller [25] [26], neural network model reference adaptive controller (NN-MRAC) [27] [28], radial-basis function neural network hybrid controller [29] [30], and back-stepping-ANN hybrid controller [31] [32].

Due to its simple yet powerful structure, the backpropagation controller (BPNN controller) is the most widely adopted neural network controller among all of the developed controllers based on ANNs. The learning concept is quite straightforward; it iteratively adjusts the neural connection weight to minimize the difference between the actual output vector and the desired output vector through the back-propagated error. A detailed analysis of BPNN controller was presented in [33], and it was proven that the proposed controller could produce a very low error. However, the iterative procedure during learning is quite time consuming, especially when the data and the chosen number of neurons in its network are large.

Kohonen self-organizing maps (SOMs) [34] have some advantages over the backpropagation neural network (BPNN) in terms of its simplicity, fast computation, and easy evaluation. Therefore, the use of an SOM controller may be able to overcome the computational cost problem associates with the BPNN controller. However, unlike the BPNN, the SOM is categorized as an unsupervised learning mechanism, and its main functions are to reduce the dimensions of the data and display the similarities of the data. Moreover, SOMs are commonly used for clustering and classification. The utilization of an SOM to approximate dynamical input–output mappings was first introduced in a vector-quantized temporal associative memory (VQTAM) model [35], and the approach was further developed in an autoregressive SOM (ARSOM) model [36]. However, the discussion on the VQTAM and ARSOM only focused on the possibilities of using methods based on mathematical models and simulations, resulting in a lack of empirical analysis of a real experimental system.

The authors have developed an SOM controller employing direct inverse control (DIC) schemes for the velocity and course control of a double-propeller boat model, and its performance characteristics have been compared with those of the backpropagation neural network controller (BPNN controller) [37] and [38]. The preliminary results revealed that the SOM controller was superior to the BPNN controller in terms of control error and training computational cost. However, the boat model movement was still limited to forward motions without any significant maneuvers, so the boat model could still be easily controlled.

This paper provides a more thorough evaluation and analysis of the proposed SOM controller. In contrast to the previous work described in [37] and [38], this paper proposes a system control for a boat model with the ability to perform a maneuvering trajectory that shows higher nonlinear characteristics. Therefore, the control becomes more challenging and complex. Furthermore, SOM controllers are directly implemented for boat model position control.

The remainder of the paper is organized as follows. Section 2 describes the developed boat model, data acquisition scheme, and ANN-based identification model. The related theories and background concepts of the BPNN and SOM as inverse controllers under the direct inverse control (DIC) scheme are explained in Section 3. Section 4 presents the simulation results based on real experimental data obtained from the developed boat model. This section also discusses the interpretation and implications of the results. Section 5 summarizes the important findings of the work.

II. THE BOAT MODEL

A. BOAT MODEL AND DATA ACQUISITION

In this work, a double-propeller boat model without a rudder (Fig. 1(a)) is designed and developed as a USV system to mimic the characteristics of a real double-propeller boat system. By using this boat model, the developed controller system can be analyzed by moving the model along the floor instead of in water, eliminating the effect of the ocean waves and currents. The boat model is assembled using a controller, two MT-BLDC motors that are connected to two T18A-ESC electric speed controllers and two 12.5 cm E-propellers, a compass sensor, a GPS sensor, a voltage regulator, and a Li-Po battery as the power system (Fig. 1(b)). This study focuses only on the trajectory control of the USV, i.e., boat model and not on the design of the boat model itself.

To design a suitable trajectory controller, the double-propeller boat model is considered as a multi-input-multi-output (MIMO) system that has two input control signals and three output parameters, which correspond to its position in the inertial frame. As shown in Fig. 2, the boat model is moved by two input control signals, i.e., PWM1 for the left motor and PWM2 for the right motor. Meanwhile, the output parameters are the boat’s direction or heading (yaw) and its coordinate position (longitude and latitude), which can be converted to a two-dimensional Cartesian coordinate system (x-axis and y-axis) by considering a predetermined point of reference. The boat movement is caused by the difference
between two thrust forces generated by the two pairs of motors and propellers on the back side of the boat model. The movements of the boat model are driven by the surge velocity \( v_x \) and the sway velocity \( v_y \), both in the body-fixed frame, and the heading angle or yaw \( \psi \) in the inertial frame [13].

FIGURE 1. The boat model: (a) real picture of the developed boat model; (b) block diagram of the main components.

FIGURE 2. Boat model as a MIMO system.

The purpose of the designed controller is to control the position of the boat model, that is, to define the correct input signals of the MIMO system and obtain the desired output signals. In this study, the controller is designed using the neural network approach, which requires sufficient data for training and testing. Here, to obtain the data for both the neural network identification and inverse controller trainings, the boat is manually controlled using a radio controller to move forward with the desired maneuvering trajectory, as shown in Fig. 3. The concerned movements are those with three degrees of freedom (3DOF), i.e., yaw, surge, and sway [1]. In this study, the boat model is controlled to form clockwise and a counterclockwise motion so that the trajectory forms a figure 8 shape. The data were recorded on a smooth ground with minimum friction between the boat model wheels and ground. Considering the data updating speed of the GPS sensor, all the data were sampled every 200 ms. Then, the obtained data were used for the neural network training. The obtained raw data consist of the two control signals of the left and right propellers, PWM1 and PWM2, the heading/yaw of the boat in the inertial frame obtained from the compass sensor, and the longitude and latitude coordinate positions of the boat obtained from the GPS sensor. These coordinate positions were then converted into \( x \) and \( y \) Cartesian positions using east north up (ENU) navigational coordinate systems and taking the initial boat position as the origin of the Cartesian coordinate.

FIGURE 3. Data acquisition with manual control.

Fig. 4 shows the 216 samples of the learning data obtained for the neural-networks-based identification and controller systems. Fig. 4(a) shows the Cartesian position of the boat model with respect to the \( x \)-axis and \( y \)-axis of the Earth, where the initial position of the boat model is set to \((0, 0)\). The graph in Fig. 4(b) shows the headings or yaw positions of the boat model obtained from the compass sensor. The lower graph in Fig. 4(c) shows the corresponding input parameters of the boat, e.g., PWM1 and PWM2, used to drive the boat model to move according to the desired maneuvering trajectory.

B. NN-BASED IDENTIFICATION FOR THE BOAT MODEL

Plant identification is performed by adopting the nonlinear autoregressive exogenous model (NARX) approximation [39], expressed as:

\[
y[k] = f(y[k-1], ..., y[k-n_y], u[k-1], ..., u[k-n_u]),
\]

(1)

where \( y \) is the plant output, \( u \) is the plant input, and \( n_y \) and \( n_u \) are the delay or memory operators for the plant output and input, respectively. In this work, the neural-network-based system identification is a multilayer perceptron neural network that consists of one input layer with 15 neurons, one hidden layer with 30 neurons, and one output layer with 3 neurons. The configuration for the boat model system identification is shown in Fig. 5. All of the neurons other than those in the input layers use a bipolar sigmoid activation function.
The boat model is identified with a neural network that can mimic the boat model behavior using the acquired training data. A backpropagation learning mechanism is adopted to train the neural network configurations at a learning rate of 0.1 and momentum of 0.7. After 2,000,000 epochs, the learning process converged with a training mean-square error (MSE) of $5.6458 \times 10^{-6}$ as shown in Fig. 6. The MSE of each output parameter is $3.335 \times 10^{-6}$ for yaw, $7.629 \times 10^{-6}$ for the $x$-position, and $5.974 \times 10^{-6}$ for the $y$-position.

Using the same control signal as shown in Fig. 4(c), the comparison of the real boat model behavior and the designed neural network-based identification system is shown in Fig. 7. The total testing MSE is $5.6458 \times 10^{-6}$, the MSE for the $x$-position is $7.629 \times 10^{-6}$, the MSE for the $y$-position is $5.974 \times 10^{-6}$, and the MSE for the yaw is $3.335 \times 10^{-6}$. A validation is performed by taking a small set of testing data as shown by the zoom in the figure; the validated MSE value for the $x$-position is $11.9 \times 10^{-6}$, the validated MSE value for the $y$-position is $5.8 \times 10^{-6}$, and the validated MSE value for the yaw is $3.5 \times 10^{-6}$. These MSE values show that the NN identification system can successfully model the real transfer function of the boat model with a very high degree of approximation.

**III. POSITION CONTROL USING AN UNSUPERVISED DIRECT INVERSE NEURAL CONTROLLER**

**A. DIRECT INVERSE NEURAL NETWORK CONTROL SCHEME**

The proposed trajectory controller system for the boat model employs open-loop direct inverse control (DIC), as illustrated in Fig. 8. To achieve an identity mapping between the desired output or signal reference $r(k)$ and the actual output $y(k)$, the plant is directly cascaded with the neural network inverse controller. The open-loop DIC structure shows that the system relies on the fidelity of the inverse model as the controller. For the open-loop DIC structure shown in Fig. 8(a), problems such as a lack of robustness may occur due to the absence of feedback signals. Recent ideas proposed to optimize the neural network DIC open-loop system include using feedback signals from the system output to be inputted backward to the neural inverse
controller, as shown in Fig. 8(b). Since some feedback signals from the plant output are used as some of the inputs to the neural inverse controller, the system may also respond to the errors of the system identification, e.g., the simulated plant or discrepancies of the plant in a real application. Moreover, the feedback signals in the open-loop DIC system with feedback may decrease the performance of the DIC controller system, as proven in [37]; therefore, this paper only consider an open-loop DIC system without feedback.

The basic idea of the proposed inverse controller is to utilize a neural network to approximate the inverse function of the plant. The inverse-model-based control strategy is chosen because this nonlinear control method is the most promising method among all of the existing nonlinear control approaches [40]. In this scheme, the controller simply acts as the inverse of the plant, with the following NARX equation:

\[ u[k] = f^{-1}(u[k-1], \ldots, u[k-n_u + 1], y[k+1], \ldots, y[k-n_y + 1]). \]  

(2)

where \( y \) is the vector of the plant outputs, \( u \) is the vector of the plant inputs, and \( n_y \) and \( n_u \) are the delay or memory operators of the plant output and input, respectively. In our developed system, the inverse transfer function of the plant, \( f^{-1} \), is replaced by the artificial neural networks.

![Figure 7](image1.png)

**FIGURE 7.** ANN-based identification for the boat model and its error values for (a) position and (b) yaw.
(6)

\[ \mathbf{u}[k] \equiv \mathbf{v}^{\text{out}}_{j^*}[k]. \]

Fig. 9 illustrates the architectural structure of the SOM controller as the ANN-based inverse controller in the DIC system of Fig. 8(a).

**IV. RESULTS AND DISCUSSIONS**

In this section, the open-loop direct inverse control scheme (Fig 8(a)) is implemented by using the ANN-based identification model as described in Section III-A as the plant block, and using SOM controller as described in Section III-B as the ANN-based inverse controller. In this study, to empirically analyze the effect of the size of the neurons on the control performance, three SOM controllers were developed by using a quarter, half, and all of training data, i.e., 54, 108 and 216 mapping neurons. The overall parameters utilized in the SOM controller design are depicted in Table 1. The initial learning rate was set to 0.9 with a reduction factor of 0.9. The minimum learning rate value was set to $10^{-6}$, since the effect of a learning rate value below this threshold could be neglected. A simulation was carried out using a computer with an Intel Core i5-3339Y processor, 4 GB memory RAM, and 20 GB SSD storage. The obtained training MSEs were $3.6 \times 10^{-3}$ for 54 mapping neurons, $2.55 \times 10^{-3}$ for 108 mapping neurons, and 0 for 216 mapping neurons. The required training times for the SOM learning processes were 0.88 seconds, 1.16 seconds, and 1.69 seconds, respectively. It should be noted that the SOM controller basically uses the mapping neurons to directly map the input to the correct output; thus, the use of 216 mapping neurons, which is the same as the size of the training data, produces zero error, since all of the input data are exactly mapped to the corresponding output data.

**TABLE I**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of data samples</td>
<td>216</td>
</tr>
<tr>
<td>Number of input neurons</td>
<td>21</td>
</tr>
<tr>
<td>Number of output neurons</td>
<td>2</td>
</tr>
<tr>
<td>Number of mapping neurons</td>
<td>54, 108, 216</td>
</tr>
<tr>
<td>Initial learning rate</td>
<td>0.9</td>
</tr>
<tr>
<td>Learning rate reduction factor</td>
<td>0.9</td>
</tr>
<tr>
<td>Minimum value of learning rate</td>
<td>$10^{-6}$</td>
</tr>
</tbody>
</table>
Using the optimum weights of the neurons obtained from the SOM training, the direct inverse control (DIC) system was then tested with real boat model data. The DIC simulation results of the SOM controller with 54, 108, and 216 neurons are shown in Figs. 10, 11, and 12, respectively. As shown in Fig. 10, the resulting MSE value for the SOM controller with 54 mapping neurons is \(5.3431 \times 10^{-5}\) with an MSE for the \(x\)-position of \(1.115 \times 10^{-4}\) and an MSE for the \(y\)-position of \(1.3 \times 10^{-5}\) and an MSE for the \(\text{yaw}\) of \(3.57 \times 10^{-5}\). As shown in Fig. 11, the DIC simulation results for the SOM controller with 108 neurons resulted in a total MSE value of \(3.5687 \times 10^{-5}\) with an MSE for the \(x\)-position of \(6.251 \times 10^{-5}\), an MSE for the \(y\)-position of \(8.94 \times 10^{-6}\) and an MSE for the \(\text{yaw}\) of \(3.561 \times 10^{-5}\). Meanwhile, the DIC simulation results for the SOM controller with 216 neurons (Fig. 12) show that the total MSE result is \(5.6458 \times 10^{-6}\) with an MSE for the \(x\)-position of \(7.629 \times 10^{-6}\), an MSE for the \(y\)-position of \(5.974 \times 10^{-6}\) and an MSE for the \(\text{yaw}\) of \(3.335 \times 10^{-6}\).

The overall simulation results are shown in detail in Table 2. It can be observed from this table that there is a trade-off between the number of SOM mapping neurons and the resulted errors, where a larger number of SOM mapping neurons produces a lower control error. However, the computational cost of the training for the controller with more mapping neurons is higher than that of the controller with fewer mapping neurons. Furthermore, the calculation time in a real implementation may also be higher, since each reference signal has to be correctly calculated and mapped by a large numbers of neurons.
The SOM-DIC system was further tested with a generated elliptical trajectory, as shown in Fig. 13. The trajectory data points were generated by using mathematical equations to guarantee an ideal shape. Then, these data are used as reference signals, in which the trained boat controller system is expected to be able to control the boat according to this reference signal.

The DIC test results for the SOM controller with 54, 108 and 216 mapping neurons are shown in Fig. 14, Fig. 15, and Fig. 16 respectively. As can be observed from Fig. 14(a), Fig. 15(a), and Fig. 16(a), the deviations in the DIC system responses from the reference signals for the first few samples were still quite large. These phenomena occurred because the DIC system required some settling time to find the correct values of the control signal, as in this case the first values of the unknown control signals were set to 0.

The resulting MSE for SOM-DIC with 54 mapping neurons is 0.0024 with an MSE for the x-position of 0.0044, an MSE for the y-position of 0.0008 and an MSE for the yaw of 0.0020. The MSE for SOM-DIC with 108 mapping neurons is 0.0023 with an MSE for the x-position of 0.0039, an MSE for the y-position of 0.0008 and an MSE for the yaw of 0.0021. Meanwhile, the MSE for SOM-DIC with 216 mapping neurons is 0.0022 with an MSE for the x-position of 0.0039, an MSE for the y-position of 0.0008 and an MSE for the yaw of 0.0020.

The overall training and testing results for the SOM-DIC are shown in Table 3. The table shows that the x-position control produced a higher error than the control of the y-position and yaw. This is related to the data generation where the x-position is modified to form an elliptical trajectory as shown in Fig. 13. Nonetheless, the boat model could still follow the generated elliptical trajectory with a low MSE on the order of 10⁻². Here, the trade-off between the numbers of SOM mapping neurons and the produced errors is also shown.
FIGURE 14. SOM-DIC testing results for an elliptical trajectory with 54 mapping neurons. (a) Normalized control performance and the error values. (b) Position control without the first 10 samples.

FIGURE 15. SOM-DIC testing results for an elliptical trajectory with 108 mapping neurons: (a) Normalized control performance and the error values. (b) Position control without the first 10 samples.

As a comparison, a backpropagation inverse controller with the same numbers of input neurons and output neurons was also investigated. As shown in Fig. 17, the architectural configuration of the backpropagation-based neural network controller (BPNN-controller) consists of one input layer with 21 neurons, one hidden layer with 15 neurons, and one output layer with 2 neurons. A backpropagation learning mechanism was utilized to train the neural network controller with a learning rate equal to 0.01 without momentum. The learning continued until it reached 186,000 epochs, and the resulting training MSE was $5.5539 \times 10^{-4}$ as shown in Fig. 18. The required time for this training iteration was 4824 seconds. It is obvious that the backpropagation training could produce a very low error; however, it requires a high computational cost.
Using the weights of the neurons obtained from the backpropagation training, the BPNN-DIC system was also tested with real boat model data. The same testing scenario as that used in the previous SOM-DIC simulation was adopted to guarantee a valid comparison with the SOM-DIC system. The BPNN-DIC simulation results for real boat model trajectory data are shown in Fig. 19. The corresponding MSE value is $1.5059 \times 10^{-5}$ with an MSE for the $x$-position of $3.072 \times 10^{-5}$, an MSE for the $y$-position of $8.55 \times 10^{-6}$ and an MSE for the yaw of $5.9 \times 10^{-6}$. The BPNN-DIC system was then further tested with the same generated elliptical trajectory (Fig. 13). The experiment results are plotted on Fig. 20, and the corresponding MSE value is $0.0039$ with an MSE for the $x$-position of $0.0053$, an MSE for the $y$-position of $0.0010$ and an MSE for the yaw of $0.0055$.

The overall comparison of the training and testing results of the SOM-DIC and BPNN-DIC systems is shown in Table 4. It is obvious that the SOM-DIC controllers required much fewer training iterations, and hence a lower computational cost than the BPNN-DIC controller. The resulting training MSEs for the SOM controllers vary, where a lower training MSE is achieved for a higher number of mapping neurons. Testing the DIC systems using real boat model trajectory data, the MSE of SOM-DIC with 216 mapping neurons is much lower than that of BPNN-DIC, whereas the MSEs of SOM-DICs with 54 and 108 mapping neurons are comparable with the MSEs of the BPNN-DIC. When the DIC systems were tested with the generated elliptical trajectory data, the proposed SOM-DIC systems were able to produce a lower MSE than the well-known BPNN controller. This result reveals that the SOM controllers are less sensitive to noisy training data and can adapt and work better than the BPNN controller when applied to a new ideal trajectory.
V. CONCLUSIONS

A neural network controller based on an unsupervised learning algorithm with self-organizing maps (SOMs) was successfully designed and implemented as a controller for a double-propeller boat model. The validity of this proposed controller was confirmed in a test with real boat model experimental data despite the fact that most ANN-based controllers cannot be designed via unsupervised learning approaches. One of the important characteristics of the proposed SOM controller is that it utilizes a mapping principle instead of an error calculation such as that used in the backpropagation (BPNN) controller. Thus, the proposed SOM controller is not very sensitive to nonideal training data with many disturbances; therefore, the controller can work successfully when utilized to control a boat model with an ideal reference trajectory. Another important characteristic of the SOM controller is the trade-off between the number of SOM mapping neurons and the produced error, where a larger number of SOM mapping neurons leads to a lower control error. Comparing the performance of the SOM controller with the well-known controller based on a backpropagation neural network (BPNN controller), it was revealed that the proposed SOM controller is superior to the BPNN controller in terms of control error and has a shorter training period. The proposed SOM controller was proven to produce a lower control error.
than the BPNN controller for the generated trajectory data due to its lower sensitivity to noisy training data. Based on this evaluation, in a real implementation of a double-propeller boat, the SOM controller is expected to be more robust than the BPNN controller in handling small disturbances that may interfere with the training data, such as a light wind and small waves. Further research will design and implement the SOM controller for use with a real double-propeller boat.

### REFERENCES


