Yacht Roaming Simulation Algorithm based on ELM

Zhefu Yu and Tianyu Wang
Naval Architecture and Ocean Engineering College, Dalian Maritime University, 116026, China
Dalian the eighth high school, 116026, China
Email: yuzf629@sina.com; 38455607@qq.com

Abstract. This paper used Google earth as data source, used the extreme learning method to establish yacht motion equations, and used PC as a platform to build a yacht simulator. The motion equations could make the virtual scene in simulator conform to the actual motion state of yacht. This method provides a new idea to realize navigation simulator. That will make the ship driving simulation training more realistic, more efficient and low cost.

1. Introduction
Yacht sailing involves high-risk and extreme conditions. Enough training the drivers to deal with emergency situation before formal sailing is important. The real ship training is an irreversible, high cost and high consumption operation, so highly realistic simulation training based on virtual reality and human-computer interaction technology has become trend in current [1]. Many maritime academies utilized marine simulator to train the crew [2,3,4], which was shown in fig.1. Although there have been many yacht simulators, but they primarily focus on the interaction between the yachts attitude and control instrument. In aspect of the virtual scene, now simulator can only present the virtual scene in a limited range, which was shown in fig.2. They can not present the virtual scene of real routes [5]. In most simulators, ship motion equations based on drainage vessels do not conform to yacht motion state, because yachts belong to waterskiing vessels [6, 7]. The motion equations of waterskiing ships are difficult to establish because wind and wave have great effect on it [8,9]. In currency, most of driving ship simulation systems are bridge simulators or console simulators. Their costs are high, as much as several million dollars [10, 11].

Figure 1: Dynamic positioning (DP) simulation system in Dalian Maritime University.

Figure 2: Yacht simulator.

In 1998, the University of Plymouth in the UK held "the10th international navigation simulator lectures conference". The participants agreed that the simulator based on high-performance PC has
become the main trend of simulator research and development. British company PC Maritime successfully launched the first PC-based driving simulator software PC Navigator. In 1993, St. Petersburg water transport State University developed the ship maneuvering and berthing simulation software (MMSIM5.00) supported by a high level ship motion mathematical model. Through solving the simultaneous equations of ship hydrodynamics, hull and hosting, some ship motion information in different case of waves, wind, currents and water depths can be gotten. The results can be shown in the electronic river charts.

With the improvement of PC performance and the improvement of Google earth data, to build low cost simulators is possible [12]. Google earth software freely presents the worldwide satellite images, aerial photography data, three-dimensional terrain and building models. It can supply rich data to show global landscape efficiently. For cities, famous scenic spots, coastline and ports, it provides high precision images. Their resolution is about 1m or 0.5m. Furthermore, Google earth software will continuously improve and update its data. It can provide worldwide detailed map information for the maritime simulation training. In this paper, we used Google earth as data source, used the "black box" approach [13, 14] to establish the motion equations of yacht, and used PC as a platform to build a yacht simulator. Specifically, we used extreme learning method to establish the motion equations of yacht with yacht sailing data collected in different sea conditions. The equations were helpful to realize yacht real-time roaming. The virtual scene presented by PC-based simulator conformed to the actual motion state of yacht. This method provided a new idea to realize navigation simulator. The yacht simulators can simulate acceleration, deceleration, parking, sharp turn and other manipulation realistically. Drivers can operation a yacht simulator to complete around the mark, berthing pier, sea rescue boats and other key project training manipulation. The simulator will make the ship driving simulation training more realistic, more efficient and low cost.

2. Extreme Learning Method

A single hidden layer feedforward neural network includes an input layer, a hidden layer and output layer. Its unified model is shown in Fig.3.

A standard SLFN with $\tilde{N}$ hidden nodes and activation function $g(x)$ are mathematically modeled as Eq.1.

$$\sum_{i=1}^{\tilde{N}} \beta_i g_i(x) = \sum_{i=1}^{\tilde{N}} \beta_i g_i(a_i \cdot x_i + b_i) = y_j, \quad j = 1 \cdots N$$

Figure 3: Single hidden layer feedforward neural network.

Where $a_i = [a_{i1}, a_{i2}, \cdots, a_{in}]^T$ is the weight vector connecting the No. $i$ hidden node and the input nodes, $\beta_i = [\beta_{i1}, \beta_{i2}, \cdots, \beta_{im}]^T$ is the weight vector connecting the No. $i$ hidden node and the output nodes, and $b_i$ is the bias of the No. $i$ hidden node. $a_i \cdot x_j$ denotes the inner product of $a_i$ and $x_j$. The output nodes are chosen linear in this paper.
The above \( N \) equations can be written compactly as Eq.2.

\[
H\beta = Y
\]

\[
H\left( a_1 \cdots a_N, b_1 \cdots b_N, x_1 \cdots x_N \right) =
\begin{bmatrix}
g(a_1 \cdot x_1 + b_1) \\
\vdots \\
g(a_N \cdot x_N + b_N)
\end{bmatrix}
\]

\[
\beta = \begin{bmatrix}
\beta_1^T \\
\vdots \\
\beta_N^T
\end{bmatrix}, \quad
Y = \begin{bmatrix}
y_1^T \\
\vdots \\
y_N^T
\end{bmatrix}
\]

(2)

That standard SLFN with \( \tilde{N} \) hidden nodes with activation function \( g(x) \) can approximate these \( N \) samples with error \( E(W) \). Back Propagation method (BP) is used to find the optimal weights \( W = (a, b, \beta) \) to minimize error function \( E(W) \) shown as Eq.3.

\[
\min_{w=(a, b, \beta)} \| e \|^2,
\]

\[
s.t. \quad \sum_{i=1}^{\tilde{N}} \beta_i g(a \cdot x_j + b_j) - y_j = e_j
\]

(3)

Where \( e_j = [e_{j1}, e_{j2}, \ldots, e_{jM}] \) is the error of the No. \( j \) sample.

The training time of BP algorithm is very long. The initial parameters of BP algorithm are sensitive. Settings the initial parameters inappropriately can cause the training network failure. The BP algorithm is easy to fall into overfit situations. In order to solve the BP issue, extreme learning algorithm is proposed by Huang on the basis of the following theorem[15].

Theorem 1. Given a standard SLFN with \( N \) hidden nodes and activation function \( g : \mathbb{R} \rightarrow \mathbb{R} \) which is infinitely differentiable in any interval, for arbitrary distinct samples \( (x_i, y_i) \), \( i = 1, 2, \ldots, N \), where \( x_i = [x_{i1}, x_{i2}, \ldots, x_{in}] \in \mathbb{R}^n \) and \( t_i = [y_{i1}, y_{i2}, \ldots, y_{im}] \in \mathbb{R}^m \), for any \( a_i \in \mathbb{R}^n \) and \( b_i \in \mathbb{R} \), the hidden layer output matrix \( H \) of the SLFN is invertible and \( E(W) = 0 \).

Theorem 2. Given any small positive value \( \epsilon > 0 \) and activation function \( g : \mathbb{R} \rightarrow \mathbb{R} \) which is infinitely differentiable in any interval, there are \( \tilde{N} \leq N \) hidden nodes for \( N \) arbitrary distinct samples \( (x_i, y_i) \), where \( x_i = [x_{i1}, x_{i2}, \ldots, x_{in}] \in \mathbb{R}^n \) and \( t_i = [y_{i1}, y_{i2}, \ldots, y_{im}] \in \mathbb{R}^m \), for any \( a_i \in \mathbb{R}^n \) and \( b_i \in \mathbb{R} \), there are matrix \( \beta \in \mathbb{R}^{N \times m} \) to make \( E(W) \leq \epsilon \) with probability one.

Detailed proof of Theorem 1 and Theorem 2 can refer to the literature [16, 17]. Theorem states that: As long as a sufficient number of hidden layer nodes, SLFN can approximate any continuous function. But in order to make SLFN good generalization performance, usually \( \tilde{N} \ll N \). Different from the BP in neural networks that all the hidden nodes in SLFN need to be tuned, ELM learning theory shows that the hidden nodes can be randomly generated. All the hidden node parameters are independent from the target functions or the training datasets. ELM theories conjecture that this randomness may be true to biological learning in animal brains. When the hidden node parameters are randomly generated, the hidden layer matrix is a definite matrix \( H \). For the sake of efficiency, the output weights of ELM may be determined in different ways. The common way is transforming SLFN into the problem of least squares solution \( H\beta = Y \).
In order to make the simulator more realistic, the training samples contained tens thousands of motion states. ELM algorithm can speed up the computing and avoid local optimal [18].

3. The Yacht Motion Equations

The joystick in yacht controls the throttle of gasoline engine. When the throttle is adjusted to a high speed, the yacht can not reach the high speed immediately. There is a delay for acceleration. The deceleration also needs a reaction time. Similarly, there is a delay from rudder angle to the rotational angular velocity. In order to describe the relationship between manipulations and the ship motions, we established yacht motion equations based on the measured sailing data.

We established two right-handed Cartesian coordinate systems shown as Fig.4. The \( OX_1Y_1Z_1 \) coordinate system is fixed to the earth. The \( OXYZ \) coordinate system was fixed to the hull, and move with ship together. Coordinate origin is in amidships. The direction of \( X \) axis was towards the bow. The direction of \( Y \) axis was towards starboard. The direction of \( Z \) axis was towards the bow. \( V \) denotes the speed; \( \omega \) denotes angular velocity; Its positive direction was clockwise.

Generally, yacht steering system is relatively simple. Therefore, this article ignored the delay from steering device to actual rudder angle. The parameter \( \theta \) given by steering device was the actual rudder angle. Driving force \( N \) could be derived from host power and steering angle. In order to build yacht motion equations, we also ignored the influence of wind and sea current. We set four kinds of sensor on yacht to acquire velocity \( V \), angular velocity \( \omega \), Ship lateral tilt \( \beta \) and lengthwise tilt \( \alpha \). The data were recorded once every three seconds. Driving force \( N \) and steering angle \( \theta \) were recorded simultaneously. Thus, six chronological sequence parameters are collected, they are driving force \( N_i \), steering angle \( \theta_i \), velocity \( V_i \), angular velocity \( \omega_i \), ship lateral tilt \( \beta_i \) and lengthwise tilt \( \alpha_i \). According to weather forecast the wave height \( h_i \) can be gotten.

![Figure 4: Two right-handed Cartesian coordinate system](image)

We established a SLFN to predict the movement of the yacht, and used the measured data to form a training sample set \( \{(x_i, y_i)\}, i = 1, 2, \cdots, n \). The 7-dimension feature vectors shown as Eq.4 are input vectors. The 4-dimension vectors shown as Eq.5 are output vectors.

\[
\begin{align*}
    x_i &= [N_i, \theta_i, V_i, \omega_i, \beta_i, \alpha_i, h_i] \in \mathbb{R}^7, \\
    y_i &= [V_{i+1}, \omega_{i+1}, \beta_{i+1}, \alpha_{i+1}] \in \mathbb{R}^4,
\end{align*}
\]

We collected tens thousands of arbitrary distinct samples that contains various motion states, therefore, the simulator could simulate various driving states.

In SLFN, RBF was selected as the excitation function. The initial number of hidden nodes was 5. The number was gradually increased by an interval of 2, until the nearly optimal number of nodes for SLFN was found by cross-validation method. ELM algorithm is shown as follows:
Step 1: Give the initial number of hidden nodes and randomly set initial values for the parameters $(a_{ij}, b_i), i = 1, 2, \cdots, N$ in the hidden nodes.

Step 2: The training set was divided into four parts. The three parts were used to train the network. The remaining one part would be used to test the output error.

Step 3: Calculated the output matrix $H$ of hidden layer with the training set $(x_i, y_i), i = 1, 2, \cdots, n$, $x_i = [N_i, \theta_i, \phi_i, \beta_i, \alpha_i, h_i]^T \in \mathbb{R}^7$, $y_i = [\beta_{i,1}, \beta_{i,2}, \beta_{i,3}, \alpha_{i,1}]^T \in \mathbb{R}^4$. We should ensure that $H$ is full rank. Then get its Moore-Penrose generalized inverse matrix $H^+$ by optimizing methods. Output the optimal parameters $\beta$ on the basis of the equation $Y = H^+ Y$, where $Y = [y_1, y_2, \cdots, y_n]^T$.

Step 4: Input the remaining one part to the trained SLFN, then get its output error. Back to step 2, take turns to leave one part of the training set as the testing set, obtained the total output error.

Step 5: Adjust the number of hidden nodes, then returned to step 1, until the total output error was minimized, the optimal number of nodes was determined.

Step 6: Put the ship motion state vector $x_j = [V_j, \theta_j, \phi_j, \beta_j, \alpha_j, h_j]^T \in \mathbb{R}^7$ into the trained SLFN, then the output is the hull attitude vector $y_j = [\beta_{j,1}, \beta_{j,2}, \beta_{j,3}, \alpha_{j,1}]^T \in \mathbb{R}^4$ in the next time point. The output formed the next state of ship motion. The process followed by cycle.

As above, we established the motion equations with black-box approach, if we maneuvered the joystick in simulator, an input vector was generated, then an output vector was generated. The output vectors formed a time series vectors by the time interval of three seconds. They are shown as Eq.6.

$$y_{j+1} = [V_j, \omega_j, \beta_j, \alpha_j] \in \mathbb{R}^4, \quad i = 1, \cdots, n$$

(6)

Every two adjacent hull attitudes $y_{j+1} = [V_j, \omega_j, \beta_j, \alpha_j] \in \mathbb{R}^4$ and $y_{i} = [V_i, \omega_i, \beta_i, \alpha_i] \in \mathbb{R}^4$ were linearly interpolated by the interval of one-third-second, then we obtained the hull attitude series shown as Eq.7.

$$y^j = [V^j, \omega^j, \beta^j, \alpha^j] \in \mathbb{R}^4, \quad j = 1, \cdots, 9n$$

(7)

Many arcs and straight lines forms real yacht route, the arcs could be divided into many short segments according to the limited differential thinking. In order to achieve yacht roaming picture, we got yacht route based on the hull attitude series. Yacht velocity multiplied with the interpolation interval, we got a segment length in roaming route. Yacht angular velocity multiplied by the interpolation interval, we got a turning angle in every interpolation point. After coordinate transformation, we calculated the longitude $L^j$, the latitude $A^j$ of the ship position, hull direction $\Phi^j$, and the hull attitude $\beta^j, \alpha^j$ for each interpolation point.

4. The realization of route roaming

Yacht route scene is consisted of complex natural terrain, environment and human activities together. The more details are shown in route scenario, the more workload is faced. The 3-D route scene in this simulator presented by means of Google Earth. Google Earth shows a digital terrain model DTM based on survey data, then the aerial photograph data or satellite image data were mapped onto the DTM to achieve the realistic display of the actual terrain. Piers, beacons and other ancillary facilities, shown as fig.5, and fig.6, were modeled by software Sketchup. After transforming SKB format to the KML format, the models were imported into Google Earth.

In order to display route scene, the space coordinate system should be transform to a viewpoint coordinate system. Viewpoint can be calculated from the parameters $[L^j, A^j, \Phi^j, \alpha^j]$. The viewpoint position decides which model to be drawn. The viewpoint can be obtained from the parameters
The viewport decides the cutting-off range, which can reduce the possession of computer resources and improve running efficiency [19].

We developed the simulator in the VC ++ environment. The interface of IapplicationGE is an important COM API. It is used to open or close Google Earth program, to acquire and operate the handles of client frame window, to set the view high, view point and other relevant information and to move the viewpoint to specified locations [20, 21]. We can realize the route roaming of yacht by the SetCameraParams function in this interface. SetCameraParams function can help us get the pictures observed from a static view point. In order, we moved the viewpoint in a very short time to achieve the roaming view. After acquiring hosts sound in various conditions, the sounds were broadcasted in simulator in accord with the joystick states.

5. Conclusions
The simulation system in this article can simulate acceleration, deceleration, parking and sharp from a driver's point of view. It even can simulate yacht roll and pitch caused by outside interference. With this simulation system, we also can carry out the training of roaming around navigation mark, berthing pier, sea rescue and other key project. The system also accessed the automatic identification system (AIS) data, so it has a standard electronic charts platform. Drivers can realistically experience each stage of yachting. But there are still shortcomings in the simulator, it can not adjust weather (wind strength, direction and variation); It does not have the function of smart target ship, the target vessel can automatically avoid or sailing in accordance with customary rules of maritime navigation [22]. We will continue to develop this simulator to satisfy the simulator technical standards in Manila amendments to the STCW convention.

6. References
[1] J. A. H. Paffett, Recent Developments in Marine Simulation, Journal of Navigation, 1981, Vol. 34, No. 2, pp165-181.
[2] S. D. Aizinov, A. V. Orekhov, Simulator training for the high technology ship crews, Gyroscope and navigation, 2010, Vol.1, No.4, pp258-262.
[3] Aizinov, S.D., Analyzing the Efficiency of Marine Simulators, Morsk. Flot. (Navy), 2006, no. 6, pp. 18–23.
[4] Sazonov, A.E. and Aizinov, S.D., Methodical Estimation of Simulator Training Efficiency, Ekspлуatatsiya morskogo transporta (Marine Transport Operation), Yemeliantsev, P.S., Ed., St. Petersburg: Nauka, 2005, no. 44, pp. 38–40.
[5] Sun Jifeng, Nie Weiguo, Su Zejian. Manipulation ship training system and relevant technical based on virtual reality [J]. South China university of technology: Natural science edition, 2004, 32 (2): pp67-71. (In Chinese)
[6] Cheng Qiming, Sun Ganchao, Wang Zhihong, Liu Qiming. The Design and implement of hardware simulator for ship maneuvering [J]. Systems engineering and electronics, 2001, 23 (7): pp4-8. (In Chinese)
[7] Hu Yaohua. A simulation system for ship motion control [J]. Journal of Dalian maritime university, 2000, 26 (2): pp25-28. (In Chinese)

[8] Hu Shengwu, Pan Zhengfei. Application of virtual reality technology and problems required to be solved [J]. Mine surveying, 2004 (3): pp13-17. (In Chinese)

[9] Ren J S, Yin Y, Zhang X F (2006) Study on maneuvering model of kamewa type waterjet ship for ship handling simulator. In: Proceedings of International Conference on Marine Simulation and Ship Manoeuvrability (Marsim2006), Terschelling, Netherlands, June 2006, pp M-20 -1–M-20 -5.

[10] Fang Quangen, Yang Zaili, Hu Shenping, Wang Jin, Formal safety assessment and application of the navigation simulators for preventing human error in ship operations, Journal of Marine Science and Application, 2005, September, Vol.4, Issue 3, pp 5-12.

[11] Guo Chen, Yang Guoxun, Sun Jianbo, and so on. Virtual reality based interactive simulation system for ship motion control [J]. Journal of system simulation, 2004, 16 (5): pp942-940. (In Chinese)

[12] T. Degre, X. Lefevre Micro-computerized maritime Traffic simulator, Maritime Simulation, In: Proceedings of the First Intercontinental Symposium, Munich, 1985, June, pp 233-241.

[13] Yong Yin, Yicheng Jin, Hongxiang Ren, Xufeng Zhang, Xiuwen Liu, Junsheng Ren. Research on key technologies of full mission navigation simulation systems [J]. Virtual Reality & Augmented Reality in Industry, 2011, pp133-146. (In Chinese)

[14] Zhang X F, Jin Y C, Yin Y et al (2004) Ship simulation using virtual reality technique. In: Proceedings VRCAI 2004 ACM SIGGRAPH International Conference on Virtual Reality Continuum and Its Applications in Industry, 2004, pp 282-285.

[15] Deng Wanyu, Zheng Qinghu, Chen Lin, Xu Xubin Research on extreme learning of neural networks[J], Chinese journal of computers. 2010, Vol.33 No.2: pp279-287. (In Chinese)

[16] G. B. Huang, Q. Y. Zhu, C. K. Siew, Extreme learning machine: theory and applications [J], Neurocomputing, 2006, Vol.70, No.1-3: pp489-501.

[17] G. B. Huang. Learning capability and storage capacity of two hidden-layer feedforward networks [J]. IEEE Transaction on Neural Networks, 2003, 14 (2): pp274-281.

[18] Vapnik V.N. The nature of statistical learning theory [M]. New York: Springer, 1995.

[19] Jiang Wengang, Shang Jie, Deng Zhili, Wang Jianming, Li Shaopeng, Research and design of steering simulation system for high speed yacht, Journal of jiangsu university of science and technology(Natural Science Edition) 2005, 19 (6): pp12-15. (In Chinese)

[20] Liu Benyong, Xu Goubiao, Li Xin, Xiang RuiLing, The design and implementation of plane route roaming based on Google earth[J], Technology in west China, 2011,10 (29): pp7-9. (In Chinese)

[21] Sun T D, Yin Y, Jin Y C (2002) Realization of viewpoint tracking in visual system of navigation simulator (In Chinese). Journal of Dalian Maritime University 28(1): pp26-30.

[22] J. N. J. Moon, D. S. Tudhope, An Agent-Directed Marine Navigation Simulator, Journal of Navigation, 2006, September, Vol. 59, Issue 3, pp 461-475.