2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Vessel Maneuvering Model Identification Using Multi-output Dynamic Radial-Basis-Function Networks

Ning Wang, Nuo Dong and Min Han

***Abstract*—In this paper, a vessel maneuvering model (VMM) based on multi-output dynamic radial-basis-function network (MDRBFN) is proposed. Data samples used for training and testing are obtained from the vessel maneuvering dynamics based on a group of nonlinear differential equations. In order to identify the vessel maneuvering model, the differential equations are transformed into nonlinear state-space form. Considering that the desired states are not only dependent on system inputs, *i.e.*, rudder defection and propeller revolution, but also previous states, the proposed MDRBFN is focus on the multi-input multi-output (MIMO) case. The structure of traditional fixed-size RBF networks is difficult to determine, so the growing and pruning algorithm is introduced to multi-output RBF networks to realize RBF networks with dynamic structure. The MDRBFN starts with no hidden neurons, and during the learning process, hidden neurons are recruited automatically according to hidden nodes generation criteria and parameters estimation. In addition, insignificant hidden nodes would be deleted if the node significance is lower than the predefined threshold. As a consequence, the proposed MDRBFN-based VMM (MDRBFN- VMM) reasonably captures the essential maneuvering dynamics with a compact structure. Finally, simulation results indicate that the proposed MDRBFN-VMM achieves promising performance in terms of approximation and prediction.**

I. INTRODUCTION

UP to now, varieties of vessel motion models have been proposed, mainly including Abkowitz model [1], MMG model [2] and response model [3]. In order to achieve high accuracy, Abkowitz model and MMG model are composed of dozens of hydrodynamic derivatives which are difficult to get. On the contrary, the response model is simple and convenient for control system design, but the model accuracy would reasonably decrease.

Based on the model frameworks mentioned above, identification investigations for hydrodynamic derivatives and input-output nonlinearities have been conducted by various methods, including recursive least square estimation [4], estimation-before-modeling technique [5], recursive maximum likelihood estimation [6], simplified linearization [7], and support vector regression method [8], *etc*. Due to the hydrodynamic nonlinearities related to the vessel dynamics, the resultant overall mathematical formulations of vessel maneuvering are usually complicated since previous traditional methods would inevitably lead to a dilemma between the accuracy and complexity of vessel motion models. Artificial neural networks (ANNs) in the field of artificial intelligence technology are introduced to overcome the problem mentioned above. ANNs are used to establish nonlinear input-output models for ship maneuvering motion effectively. In [9], the ANN and spectral analysis methods were adopted to identify the hydrodynamic derivatives in the mathematical model of marine vehicle motions. Rajesh *et al.* [10] identified an interesting nonlinear maneuvering model of large tankers based on BP neural networks. The system identification method based on ANNs can obtain considerable performance for approximation and generalization. However, BP neural networks have serious disadvantages, *i.e.*, slow convergence speed and easy to fail into local minimum value. In [11], Wang proposed generalized ellipsoidal basis function based fuzzy neural networks (GEBF-FNN). And GEBF-FNN-based vessel maneuvering model and controller were researched in [12-15]. Lately, a novel constructive multi-output extreme learning machine (CM-ELM) [16] was proposed to identify large tanker dynamics with fast speed.

Radial-basis-function (RBF) networks have gained much popularity due to their ability to approximate complex nonlinear mapping. And many works have been done to improve the performance of RBF neural networks. In [17], Platt proposed resource allocation network (RAN) in which hidden neurons were added sequentially based on the novelty of the new data. Lately, Kadirkamanathan and Niranjan [18] adopted extended Kalman filter (EKF) instead of least mean square (LMS) algorithm to update the network parameters, which is known as RANEKF. However, the RAN and RANEKF can only add neurons and cannot prune insignificant neurons from the network, and thereby leading to an extremely large network. To overcome the problem mentioned above, Lu *et al.* [19] proposed a neural network known as minimal resource allocating networks (MRAN), where a pruning strategy was introduced. A sliding data window in the growing and pruning criteria is used to identify the neurons which contribute relatively little to the network output. But choosing proper window size can only be done by trial and error based on exhaustive simulation studies. Lately, Huang *et al.* [20] proposed a simple sequential growing and pruning algorithm based on the relationship between the significance of a neuron and the required learning accuracy

N. Wang and N. Dong are with Marine Engineering College, Dalian Maritime University, Dalian 116026, China (e-mail: n.wang.dmu.cn@gmail.com). M. Han is with Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, P. R. China (email: minhan@dlut.edu.cn).

This work is supported by the National Natural Science Foundation of P. R. China (under Grants 51009017, 51379002 and 61074096), Applied Basic Research Funds from Ministry of Transport of P. R. China (under Grant 2012-329-225-060), China Postdoctoral Science Foundation (under Grant 2012M520629), Program for Liaoning Excellent Talents in University (under Grant LJQ2013055), and Fundamental Research Funds for the Central Universities of P. R. China (under Grants 2009QN025, 2011JC002 and 3132013025).

for RBF networks, referred to as GAP-RBF.

This paper proposes to approximate vessel motion model (VMM) based on multi-output dynamic radial-basis-function network (MDRBFN). Training and checking samples are generated by the reference model. And then the training samples are used to identify vessel maneuvering dynamics by the MDRBFN. The system starts with no hidden nodes. During the online training process, the hidden neurons are generated according to its novelty and deleted according to significance to system performance, and the parameters are adjusted depending on the criteria. As a consequence, a powerful MDRBFN-VMM is obtained after online training process. In order to demonstrate the effectiveness of the proposed MDRBFN-VMM, simulation studies are conducted on typical benchmark scenarios, *i.e.* zig-zag maneuvering. The results indicated that the MDRBFN-VMM achieves promising identification and generalization performance.

The rest of this paper is organized as follows. Section II presents the ship motion dynamics which are composed of several differential equations. In Section III, the reference model of vessel maneuvering is described. The learning scheme of MDRBFN is presented in Section IV. Comprehensive simulation studies are conducted in Section V. Section VI is the conclusion of this paper.

II. VESSEL MANEUVERING DYNAMICS

The vessel motion dynamics considered in this paper can be represented by the following non-dimensional surge, sway, and yaw equations (Bis-system),

$$\begin{cases} \dot{u} - vr = gX'' \\ \dot{v} + ur = gY'' \\ (Lk_z'')^2\dot{r} + Lx_G''ur = gLN'' \\ \dot{x} = u\cos(\psi) - v\sin(\psi) \\ \dot{y} = u\sin(\psi) + v\cos(\psi) \\ \dot{\psi} = r \end{cases} \tag{1}$$

where,

$$\begin{aligned} gX'' = {} & X_{\dot{u}}''\dot{u} + L^{-1}X_{uu}''u^2 + L^{-1}X_{vv}''v^2 + L^{-1}X_{c|c|\delta\delta}''c|c|\delta^2 \\ & + L^{-1}X_{c|c|\beta\delta}''c|c|\beta\delta + gT''(1-\hat{t}) + X_{\dot{u}\zeta}''\dot{u}\zeta \\ & + L^{-1}X_{uu\zeta}''u^2\zeta + X_{ur\zeta}''ur\zeta + L^{-1}X_{vv\zeta\zeta}''v^2\zeta^2 \end{aligned} \tag{2}$$

$$\begin{aligned} gY'' = {} & Y_{\dot{v}}''\dot{v} + L^{-1}Y_{v|v|}''v|v| + L^{-1}Y_{ur}''ur + L^{-1}Y_{c|c|\delta}''c|c|\delta \\ & + L^{-1}Y_{|c|c|\beta|\beta|\delta|}'' + Y_T''gT'' + Y_{ur\zeta}''ur\zeta + L^{-1}Y_{uv\zeta}''uv\zeta \\ & + L^{-1}Y_{|v|v\zeta}''|v|v\zeta + L^{-1}Y_{|c|c|\beta|\beta|\delta|\zeta}''|c|c|\beta|\beta|\delta|\zeta \\ & + Y_{\dot{v}}''\dot{v} - Y_{\dot{v}\zeta}''\dot{v}\zeta \end{aligned} \tag{3}$$

$$\begin{aligned} gLN'' = {} & L^2(N_{\dot{r}}''\dot{r} + N_{\dot{r}\zeta}''\dot{r}\zeta) + N_{uv}''uv + LN_{|v|r}''|v|r \\ & + N_{|c|c\delta}''|c|c\delta + LN_{ur}''ur + N_{|c|c|\beta|\beta|\delta|}''|c|c|\beta|\beta|\delta| \\ & + LN_{ur\zeta}''ur\zeta + LN_{uv\zeta}''uv\zeta + LN_{|v|r\zeta}''|v|r\zeta \\ & + N_{|c|c|\beta|\beta|\delta|\zeta}''|c|c|\beta|\beta|\delta|\zeta + LN_T''gT'' \end{aligned} \tag{4}$$

$$\begin{cases} gT'' = L^{-1}T_{uu}''u^2 + T_{un}''un + LT_{|n|n}''|n|n \\ k_z'' = L^{-1}\sqrt{I_z''/m} \\ c^2 = c_{un}un + c_{nn}n^2 \\ \zeta = d/(h-d) \\ \beta = v/u \end{cases} \tag{5}$$

where u, v and x, y are the velocities and positions along X axis (toward forward) and Y axis (toward starboard), ψ is the yaw angle in the horizontal plane, and its derivative r is the yaw rate. L, d, and m are the length, draft and mass of the ship, I_z'' is its mass moment of inertia about Z axis (vertically downward with axis origin at free surface), x_G'' is the non-dimensional X coordinate of ship's center of gravity, g is acceleration due to gravity, X'', Y'', and N'' are nonlinear non-dimensional surge force, sway force, and yaw moment respectively, δ is the rudder angle, c is the flow velocity past rudder, ζ is the water depth parameter, c_{un} and c_{nn} are constants, T'' is the propeller thrust, h is the water depth, $\hat{t}$ is the thrust deduction factor and n is the rpm of the propeller shaft. All other quantities are constant hydrodynamic derivatives are given by [21].

In order to generate data samples conveniently, the equations above are transformed into the following nonlinear state-space equation:

$$\begin{cases} \dot{u} = \dfrac{1}{L(1 - X_{\dot{u}}'' - X_{\dot{u}\zeta}''\zeta)}((X_{uu}'' + X_{uu\zeta}''\zeta)u^2 \\ \quad + L(1 + X_{vr}'' + X_{ur\zeta}''\zeta)ur + X_{c|c|\delta\delta}''c|c|\delta^2 \\ \quad + (X_{vv}'' + X_{vv\zeta\zeta}''\zeta^2)v^2 + X_{c|c|\beta\delta}''c|c|\beta\delta + \\ \quad LgT''(1-\hat{t})) \\ \dot{v} = \dfrac{1}{L(1 - Y_{\dot{v}}'' - Y_{\dot{v}\zeta}''\zeta)}(Y_{uv}''uv + Y_{|v|v}''|v|v \\ \quad + Y_{|c|c\delta}''|c|c\delta + L(Y_{ur}'' - 1)ur + LY_{ur\zeta}''ur\zeta \\ \quad + Y_{|c|c|\beta|\beta|\delta|}''|c|c|\beta|\beta|\delta| + Y_{uv\zeta}''uv\zeta + Y_{|v|v\zeta}''|v|v\zeta \\ \quad + Y_{|c|c|\beta|\beta|\delta|\zeta}''|c|c|\beta|\beta|\delta|\zeta + LY_T''gT'') \\ \dot{r} = \dfrac{1}{L^2(k_z''^2 - N_{\dot{r}}'' - N_{\dot{r}\zeta}'')}(N_{uv}''uv \\ \quad + L(N_{ur}'' - x_G'')ur + N_{|c|c|\beta|\beta|\delta|}''|c|c|\beta|\beta|\delta| \\ \quad + LN_{|v|r}''|v|r + LN_{ur\zeta}''ur\zeta + N_{uv\zeta}''uv\zeta \\ \quad + LN_{|v|r\zeta}''|v|r\zeta + N_{|c|c|\beta|\beta|\delta|\zeta}''|c|c|\beta|\beta|\delta|\zeta \\ \quad + N_{|c|c\delta}''|c|c\delta + LN_T''gT'') \\ \dot{x} = u\cos(\psi) - v\sin(\psi) \\ \dot{y} = u\sin(\psi) + v\cos(\psi) \\ \dot{\psi} = r \end{cases} \tag{6}$$

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{7}$$

where $\mathbf{x} = [u, v, r, x, y, \psi]^T$ is the state vector, $\mathbf{u} = [\delta, n, h]^T$ is the input vector and the nonlinear function vector $\mathbf{f} = [f_u, f_v, f_r, f_x, f_y, f_\psi]^T$ represents the system (6).

And the resulting speed of the vessel can be given by

$$U = \sqrt{u^2 + v^2} \tag{8}$$

III. Reference Model of Vessel Maneuvering

As described in Section II, the vessel motion model consists of six nonlinear equations which are complicated due to varieties of hydrodynamics derivatives and complex structure. Obviously, the previous nonlinear equations are unsuitable for controller design.

According to the nonlinear dynamics mentioned above, we can intuitively develop a feasible reference model of vessel motion for system identification using data-driven methods. And the state variable can be estimated by (6). The corresponding differential equations are used as reference models for data-driven system identification which can be described as follows,

$$\begin{cases} u(i+1) = f_u(\mathbf{x}(i), \mathbf{u}(i)) \\ v(i+1) = f_v(\mathbf{x}(i), \mathbf{u}(i)) \\ r(i+1) = f_r(\mathbf{x}(i), \mathbf{u}(i)) \\ x(i+1) = f_x(\mathbf{x}(i), \mathbf{u}(i)) \\ y(i+1) = f_y(\mathbf{x}(i), \mathbf{u}(i)) \\ \psi(i+1) = f_\psi(\mathbf{x}(i), \mathbf{u}(i)) \end{cases} \tag{9}$$

In order to simplify the problem, we consider that the vessel sails under constant propeller rpm (n=constant) in deep water (h=$constant$>>d). So the equation (9) can be rewritten as:

$$\mathbf{x}(i+1) = \tilde{\mathbf{f}}(\mathbf{x}(i), \delta(i)) \tag{10}$$

where, i denotes the sample index, $\mathbf{x}$ is state vector including velocities, positions and so on, δ (rudder angle) is input variable and $\tilde{\mathbf{f}}$ is nonlinear function vector implemented by (9). As a consequence, the reference model (10) can be rationally used to generate data samples for training neural networks.

IV. MDRBFN-based Vessel Maneuvering Model

Radial-basis-function neural networks are well suited for function approximation and pattern recognition due to their simple topological structure and their ability to reveal how learning proceeds in an explicit manner. However, the traditional fixed-size networks either use too few units in which case the network memorizes poorly or too many in which case the network generalizes poorly. And in this section, a growing and pruning algorithm is introduced to RBF networks to achieve high performance with compact structure.

A. Architecture of MDRBFN-based Vessel Maneuvering

The proposed MDRBFN is a three-layer network as the traditional RBF neural networks.

Layer 1: Each node in layer 1 represents an input variable.

Layer 2: The second layer consists of units that respond to a local region of the space of input values.

$$\varphi_k(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \boldsymbol{\mu}_k\|^2}{\sigma_k^2}\right) \tag{11}$$

where, $\varphi_k(\mathbf{x})$ is the response of the kth hidden neuron for an input vector $\mathbf{x}$. $\boldsymbol{\mu}_k = (\mu_{k,1}, \ldots, \mu_{k,7})^{\mathrm{T}} \in \mathbf{R}^7$ and σ_k are the center and width of the kth hidden neuron, respectively, k=1,…,K. K is the number of hidden neurons.

Layer 3: The inputs to the synapses of the third layer are the outputs of the second layer. The purpose of each third-layer synapse is to define the contribution of each second-layer unit to a particular output y of the network. The output of a RBF network with K hidden neurons for an input vector $\mathbf{x} = (x_1, \cdots, x_7)^T \in X \subseteq \mathbf{R}^7$

$$f(\mathbf{x}) = \sum_{k=1}^{K} \alpha_k \varphi_k(\mathbf{x}) \tag{12}$$

where α_k is the weight connecting the kth hidden neuron to the output neuron.

B. Learning Scheme for MDRBFN-VMM

For each observation ($\mathbf{X}^i$, $\mathbf{T}^i$), i=1,2,…, n, where n is the number of total training data pairs, $\mathbf{X}^i$□$\mathbf{R}^7$ and $\mathbf{T}^i = [t_1^i, t_2^i, ..., t_6^i]^T \in \mathbf{R}^6$ are the ith input vector and the desired output vector. The output vector of the MDRBFN-VMM, $\mathbf{Y}^i = [y_1^i, y_2^i, ..., y_6^i]^T$ of the existing structure could be obtained by (12). Before the first observation ($\mathbf{X}^1$, $\mathbf{T}^1$) arrives, the MDRBFN-VMM has no hidden neurons. It grows according to the criteria, including neuron generation criterion, allocation of RBF unit, weight adjustment and pruning strategy.

1) Criteria of Neuron Generation

(1) System Error

When the ith observation ($\mathbf{X}^i$, $\mathbf{T}^i$) arrives, calculate the system error as follows:

$$\|e^i\| = \|\mathbf{T}^i - \mathbf{Y}^i\|, i = 1, 2, \cdots, n \tag{13}$$

If

$$\|e^i\| > k_e, k_e = \max\{e_{\max}\beta^{i-1}, e_{\min}\} \tag{14}$$

a new RBF hidden neuron should be considered. Here, $e_{\max}$ is predefined maximum error, $e_{\min}$ is the desired accuracy and $\beta \in (0,1)$ is the convergence constant.

(2) Accommodation Boundary

The accommodation criterion is described as follows: for the ith observation ($\mathbf{X}^i$, $\mathbf{T}^i$), calculate the distance $d_i(k)$ between the observation $\mathbf{X}^i$ and the center $\mathbf{C}_k$ of the existing RBF units:

$$d_i(k) = \|\mathbf{X}^i - \mathbf{C}_k\| \tag{15}$$

$$d_{i,\min} = \arg\min_k (d_i(k)) \tag{16}$$

If

$$d_{i,\min} > k_d, k_d = \max\{d_{\max}\gamma^{i-1}, d_{\min}\} \tag{17}$$

a new RBF hidden neuron should be created. Here, $d_{\max}$ and $d_{\min}$ are the upper and lower bound of effectiveness radius of the accommodation boundary, and the term γ□(0,1) is the convergence constant.

2) Allocation of RBF Unit

The initial parameters of new hidden node are set as follows:

$$\mathbf{C}_k = \mathbf{X}^i \tag{18}$$

$$\sigma_u = \kappa \times d_{\min} \tag{19}$$

where □ (□ >1) is an overlap factor that determines the overlap between the RBF units.

When the first pattern ($\mathbf{X}^1$,$\mathbf{T}^1$) enters the MDRBFN, it is adopted as the first neuron: $\mathbf{C}_1=\mathbf{X}^1$, $\sigma_1=\sigma_0$, where σ_0 is a prespecified constant.

The discussion above focuses on only one case *i.e.*, $\|e_i\|>k_e$ and $d_{\min}>k_d$, where, a new hidden neuron should be generated. For the other three cases, the algorithm is as follows:

If $\|e_i\|\le k_e$ and $d_{\min}\le k_d$, MDRBFN can accommodate the observation completely. Nothing need be done or only the weights should be updated.

If $\|e_i\|\le k_e$ and $d_{\min}>k_d$, the system has good generalization and only weights should be adjusted.

If $\|e_i\|>k_e$ and $d_{\min}\le k_d$, the nearest RBF unit is not so good for generalization, and the width of the nearest RBF node and all the weights should be updated simultaneously.

For the nearest kth RBF unit clustering ($\mathbf{X}^i$, $\mathbf{T}^i$):

$$\sigma_k^i = k_w \times \sigma_k^{i-1} \tag{20}$$

where k_w is a predefined constant.

3) Weight Adjustment

Assume the nth training pattern enters the MDRBFN, all these n data pairs are memorized in the input matrix $\mathbf{P}$□$\mathbf{R}^{7\times n}$ and the output matrix $\mathbf{T}\in\mathbf{R}^{6\times n}$, respectively, upon which the weights are determined.

Considering the nth training pattern for the MDRBFN, K hidden neurons are generated according to the criteria stated above. The hidden neurons output can be obtained according to (11). Let

$$\mathbf{\Phi} = \begin{pmatrix} \varphi_{11} & \cdots & \varphi_{1n} \\ \vdots & \ddots & \vdots \\ \varphi_{K1} & \cdots & \varphi_{Kn} \end{pmatrix} \tag{21}$$

The output $\mathbf{Y}$ and error $\tilde{\mathbf{E}}$ of the system are as follows:

$$\mathbf{Y} = \mathbf{W\Phi} \tag{22}$$

$$\tilde{\mathbf{E}} = \left\| \mathbf{T} - \mathbf{Y} \right\| \tag{23}$$

In order to minimize the error energy $\tilde{\mathbf{E}}^T\tilde{\mathbf{E}}$ and find the optimal coefficient vector $\mathbf{W}^*$, the well-known linear least squares (LLS) method is employed as follows:

$$\mathbf{W}^* = \mathbf{T\Phi}^+ \tag{24}$$

where $\mathbf{\Phi}^+$ is the pseudoinverse of $\mathbf{\Phi}$ as,

$$\mathbf{\Phi}^+ = (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T \tag{25}$$

4) Criteria of pruning nodes

In order to get the parsimonious architecture of the MDRBFN, a pruning technology called error reduction ratio (ERR) method [22] is adopted as follows:

$$\mathbf{D} = \mathbf{H\Theta} + \mathbf{E} \tag{26}$$

where

$$\mathbf{D}=\mathbf{T}^T$$
$$\mathbf{H}=\mathbf{\Phi}^T$$
$$\mathbf{\Theta}=\mathbf{W}^T$$

$\mathbf{H}$ can be decomposed into

$$\mathbf{H}=\mathbf{W_2A} \tag{27}$$

where $\mathbf{W_2}$ is an $n\times K$ matrix with orthogonal columns and $\mathbf{A}$ is a $K\times K$ upper triangular matrix. Substituting (27) into (26) yields

$$\mathbf{D}=\mathbf{W_2A\Theta}+\mathbf{E}=\mathbf{W_2G}+\mathbf{E} \tag{28}$$

The LLS solution of $\mathbf{G}$ is given by $\mathbf{G}=(\mathbf{W_2}^T\mathbf{W_2})^{-1}\mathbf{W_2}^T\mathbf{D}$, or

$$g_i = \frac{\omega_i^T D}{\omega_i^T \omega_i} \tag{29}$$

where ω_i is the ith column of $\mathbf{W}_2$.

As ω_i and ω_j are orthogonal for $i \neq j$, the sum of squares or energy of $\mathbf{D}$ is as follows:

$$\mathbf{D}^T\mathbf{D} = \sum_{i=1}^{K} g_i^2 \omega_i^T \omega_i + \mathbf{E}^T\mathbf{E} \tag{30}$$

If $\mathbf{D}$ is the desired output vector after its mean has been removed, the variance of $\mathbf{D}$ is given by

$$n^{-1}\mathbf{D}^T\mathbf{D} = n^{-1}\sum_{i=1}^{K} g_i^2 \omega_i^T \omega_i + n^{-1}\mathbf{E}^T\mathbf{E} \tag{31}$$

Note that $\sum_{i=1}^{K} g_i^2\omega_i^T\omega_i / n$ is the part of the desired output variance. Thus, $g_i^2\omega_i^T\omega_i / n$ can be regarded as the increment to the explained desired output variance introduced by ω_i and an ERR due to ω_i can be defined as,

$$err_i = \frac{g_i^2\omega_i^T\omega_i}{\mathbf{D}^T\mathbf{D}} = \frac{(\omega_i^T\mathbf{D})^2}{\omega_i^T\omega_i\mathbf{D}^T\mathbf{D}} \qquad 1 \le i \le K \tag{32}$$

The ERR in (32) offers a simple and effective means of seeking a subset of significant regressor.

Define the ERR matrix $\mathbf{\Delta}=(err_1,err_2,\ldots,err_K)\in \mathrm{R}^K$, where err_i represents the error reduction ratio due to the ith column vector of $\mathbf{W_2}$. And err_i is defined as the significance of ith RBF unit.

If $err_i<k_{err}$ (a prespecified threshold), the ith RBF neuron is considered insignificant and will be removed from the system. Otherwise, no RBF neurons will be deleted.

V. Simulation Studies

Comprehensive simulation studies are conducted in this part. The Esso Osaka 190,000 dwt tanker is used to generate the training and checking data and the model's parameters can be found in [21]. All the simulation studies are carried out in the unified environment running in Matlab R2011a version with an Intel Core i3, 2.27 GHz CPU.

To identify the tanker model, the external input signal (rudder angle $\delta(i)$) is governed by the following sine function under rudder rate limitation,

$$\begin{cases} \delta(i) = A\sin(\omega i\Delta T), i\in[0,n] \\ |\delta| \le \delta_{\max} \\ |\dot{\delta}| \le \dot{\delta}_{\max} \end{cases} \tag{33}$$

where, A and ω are amplitude and frequency of rudder angle generation function. ΔT and n are sampling period and data number. These parameters are chosen as follows: A=35(deg.), ω=9(deg./sec.), n=3000, ΔT=1(sec.), $\delta_{\max}$=35(deg.) and $\dot{\delta}_{\max}$=2.33(deg./sec.).

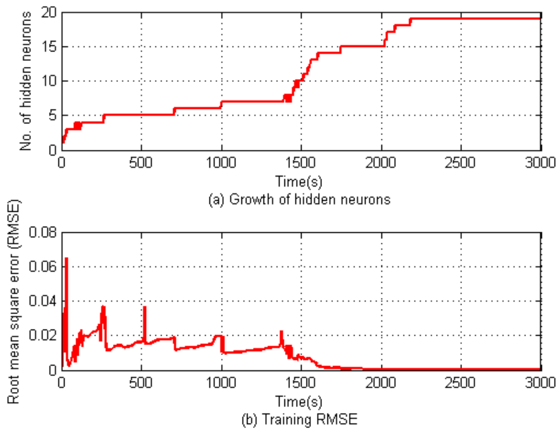

Fig. 1. Hidden node growth & RMSE during online training.

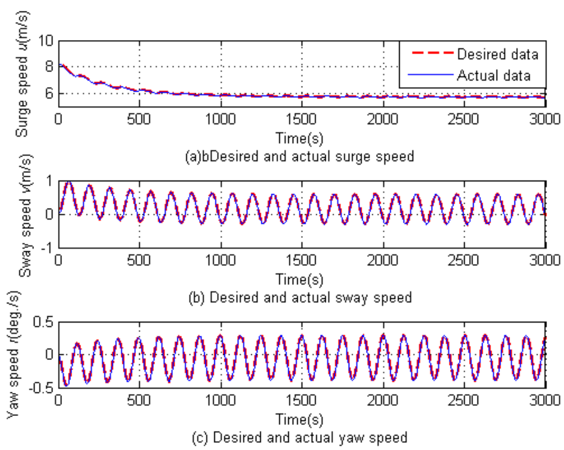

Fig. 2. Identification results of u, v, r during online training.

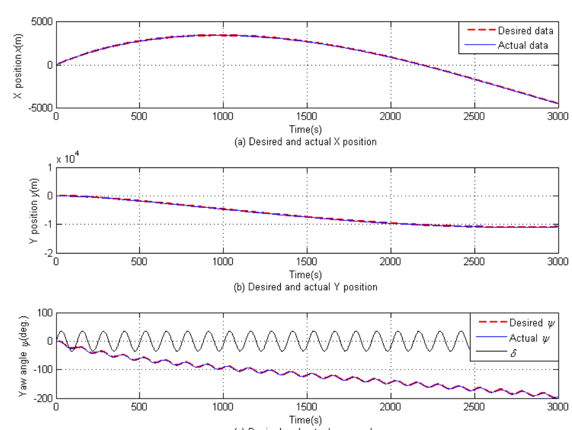

Fig. 3. Identification results of x, y, ψ during online training.

Simulation results in Fig. 1 indicate that the MDRBFN-VMM can identify well the original model since the training error index (RMSE) tends to zero. Finally, the MDRBFN-VMM employs 19 hidden neurons for the resulting system which obtains remarkable identification results for variables u, v, r, x, y, ψ, U shown in Fig.2-Fig.4, respectively.

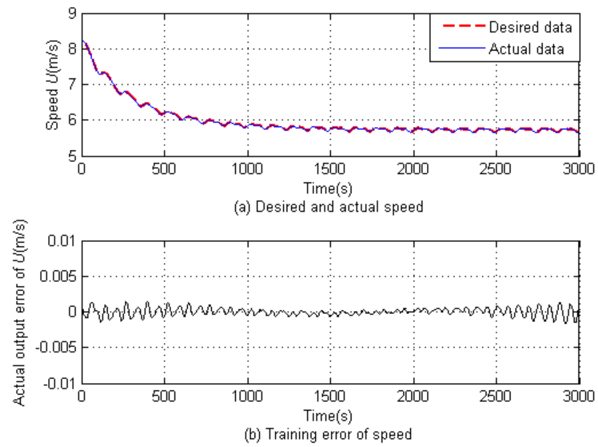

Fig. 4. Identification results of x, y, ψ during online training.

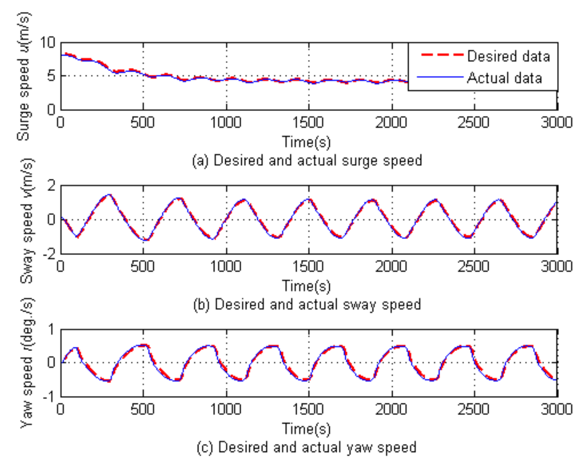

Fig. 5. Prediction results of u, v, r while zz-20deg-20deg.

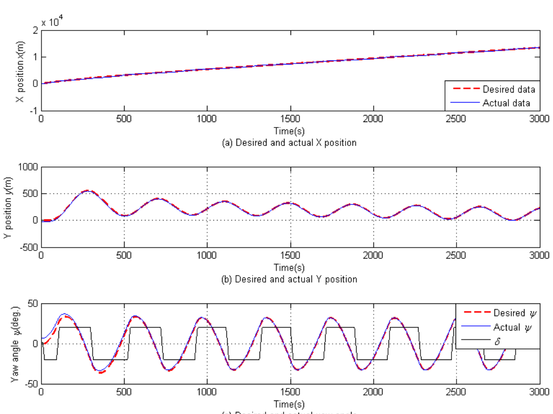

Fig. 6. Prediction results of x, y, ψ while zz-20deg-20deg.

In order to demonstrate the generalization and prediction capabilities of the proposed MDRBFN-VMM, zig-zag maneuvers are used to generate checking data. The typical zig-zag maneuver denoted by zz-(heading angle)-θ(rudder angle) with the design speed U_0=8.2311m/s(16 knot) as the initial speed is used to conduct the checking process and comparisons with original model given by (7).

Simulation results of zz-20deg-20deg are depicted in

Fig.5-Fig.7, where x, y, ψ, U can be predicted accurately during the whole maneuvering since the maximal relative errors uniformly lie within 1% except that those of variables u, v and r are under 6%, 8% and 9%, respectively. It follows that the proposed MDRBFN-VMM is able to identify and predict the original vessel motion dynamics.

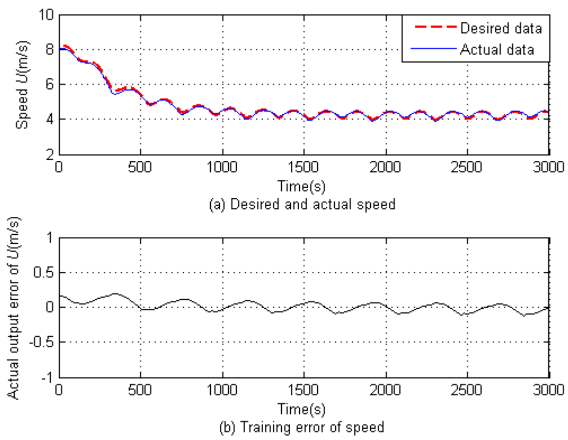

Fig. 7. Prediction results of U while zz-20deg-20deg.

VI. Conclusion

In this paper, we present a system identification method for vessel maneuvering models by using the promising Multi-output Dynamic Radial-Basis-Function Networks (MDRBFN) algorithm. To be specific, training and checking data are generated by a group of well established nonlinear differential equations for maneuvering dynamics. The generated data is used to identify the MDRBFN-based vessel maneuvering model (MDRBFN-VMM). The promising MDRBFN method is used to train data samples by using the generating and pruning criteria of hidden nodes and parameter update strategies. As a consequence, a powerful MDRBFN- VMM with superior performance of approximation and prediction is obtained after online learning. In order to demonstrate the effectiveness of the proposed scheme for vessel maneuvering dynamics identification, simulation studies are conducted on typical benchmark scenarios, *i.e.* zig-zag maneuvers. The results indicate that the MDRBFN-VMM achieves remarkable performance.

References

[1] M. A. Abkowitz, "Measurement of hydrodynamic characteristic from ship maneuvering trial by system identification," *SNAME Trans.*, vol. 80, 283–318, 1980.

[2] A. Ogawa and H. Kasai, "On the mathematical model of maneuvering motion of ship," *Int. Shipbuild. Prog.*, vol. 25, pp. 306–319, 1978.

[3] K. Nomoto, T. Taguchi, K. Honda and S. Hirano, "On the steering qualities of ships," *Int. Shipbuild. Prog.*, vol. 4, no. 35, pp. 354–370, 1957.

[4] K. J. Astrom and C. J. Kallstrom, "Identification of ship steering dynamics," *Int. Shipbuild. Prog.*, vol. 12, no. 35, pp. 9–22, 1976.

[5] H. K. Yoon and K. P. Rhee, "Identification of hydrodynamic coefficients in ship maneuvering equations of motion by estimation-before-modeling technique," *Ocean Eng.*, vol. 30, no. 18, pp. 2379–2404, 2003.

[6] C. G. Kallstrom and K. J. Astrom, "Experiences of system identification applied to ship steering," *Automatica.*, vol. 17, no. 1, pp. 187–189, 1981.

[7] S. Sutulo, L. Moreiar and C. Guedes Soares, "Mathematical models of ship path prediction in maneuvering simulation systems," *Ocean Eng.*, vol. 29, no. 1, pp. 1–19, 2002.

[8] X. G. Zhang and Z. J. Zou, "Identification of Abkowitz model for ship maneuvering motion using ε-support vector regression," *J. Hydrodynamics*, vol. 23, no. 3, pp. 353–360, 2011.

[9] A. B. Mahfouz and M. R. Haddara, "Effects of the damping an excitation on the identification of the hydrodynamic parameters for an underwater robotic vehicle," *Ocean Eng.*, vol. 30, no. 8, pp. 1005–1025, 2003;

[10] G. Rajesh and S. K. Bhattacharyya, "System identification for nonlinear maneuvering of large tankers using artificial neural networks," *Appl. Ocean Res.*, vol. 30, pp. 256–263, 2008;

[11] N. Wang, "A generalized ellipsoidal basis function based online self-constructing fuzzy neural network," *Neural Process. Lett*., vol. 34, pp. 13–37, 2011;

[12] N. Wang, D. Wang and T. Li, "A novel vessel maneuvering model via GEBF based fuzzy neural networks," in *Proc. 31th Chinese Control Conf.*, Hefei, 2012, pp. 7226–7031.

[13] N. Wang, D. Wang and T. Li, "Dynamic response model for tanker steering using GEBF based fuzzy neural networks," in *Proc. 31th Chinese Control Conf.*, Hefei, 2012, pp. 7032–7037.

[14] N. Wang, Z. Wu, C. Qiu and T. Li, "Vessel steering control using generalized ellipsoidal basis function based fuzzy neural networks," in *Proc. 9th International Symposium on Neural Networks*, Shenyang, 2012, pp. 515–524.

[15] N. Wang, X. Wang, Y. Tan, P. Shao and M. Han, "An improved Learning scheme for extracting T-S fuzzy rules from data samples," in Proc. 10th International Symposium on Neural Networks, Dalian, 2013, pp. 53-60.

[16] N. Wang, M. Han, N. Dong, and M. J. Er, "Constructive multi-output extreme learning machine with application to large tanker motion dynamics identification," *Neurocomputing*, [Online], Available: http://dx.doi.org/10.1016/i.neucom.2013.01. 062

[17] J. Platt, "A resource allocating network for function interpolation," *Neural Computa.*, vol 3, pp. 213-225, 1991.

[18] V. Kadirkamanathan and M. Niranjan, "A function estimation approach to sequential learning with neural network," *Neural Computa.*, vol. 5, pp. 954975, 1993.

[19] L. Yingwei, N. Sundararajan, and P. Saratchandran, "A sequential learning scheme for function approximation by using minimal radial basis function neural networks," *Neural Computa.* ,vol. 9, pp. 461-478, 1997.

[20] G. Huang, P. Saratchandran, and N. Sundararajan, "An efficient sequential learning algorithm for growing and pruning RBF (GAP-RBF) networks," *IEEE Trans. Syst., Man, Cybern. B*, vol. 34, no. 6, pp. 2284-2292, Dec. 2004.

[21] W. B. Van Berlekom and T. A. Goddard, "Maneuvering of large tankers," *SNAME Trans.*, vol. 80, pp. 264-298, 1972.

[22] S. Wu and M. J. Er, "Dynamic fuzzy neural networks: A novel approach to function approximation," *IEEE Trans. Syst. Man, Cybern, B*, vol. 30, pp.358-364, 2000.