Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 14, no. 2, February 2018, pp. 269-279
DOI: 10.23940/ijpe.18.02.p8.269279

Intelligent Identification of Ocean Parameters based on RBF Neural Networks

Li Yuan[a], Wei Wu[b,*], Chuan Tian[b], Wei Song[b], Xinghui Cao[b], Lixin Liu[b]

[a]*Department of Physical Science, Hainan Medical University, Haikou, 571179, China*
[b]*Institute of Deep-sea Science and Engineering, Chinese Academy of Sciences, Sanya, 572000, China*

Abstract

Ocean data assimilation is challenging because of interactive marine environmental parameters that are affected by macroscopic ocean dynamics. In order to overcome these challenges, a multi-variable assimilation scheme based on a Radial Basis Function (RBF) Neural Network is proposed in this paper. Relative influential parameters are considered as bounded time series variables so that they can be selected for nonlinear function approximating in the first stage. Then, a RBF Neural Network identification model is designed to simulate multiple interactive high-dimensional variables. This simulation is performed by applying proper hidden neurons. According to experimental results, this training method successfully approximates real circumstances. The identification accuracy and vibration are well constricted in the margin evaluated by 1.6×10^{-5}.

Keywords: intelligent identification; RBF neural networks; ocean data prediction; nonlinear modeling

(Submitted on November 2, 2017; Revised on December 29, 2017; Accepted on January 16, 2018)

1. Introduction

Often, data acquired from detection devices are unacceptable because the observation is insufficient or the device is incorrectly operated. Though global ocean observing systems provided by public organizations are generally powerful and timely [23], acquiring a comprehensive data set of the huge three-dimensional ocean is not easy to realize, and requires elaborate cooperation between ocean instruments and data processing technologies. The detectors should be placed in a more refined grid in order to provide elaborate three-dimensional data. Additionally, accurate data conversion operation in transducers and efficient transmission networks of detection arrays are helpful to enhance a good ocean observation system. However, major investment is required to develop these two aspects, making improvement difficult to justify financially. A proposed supplementary remedy is data assimilation, which remains to be a fundamental research before further digging. Using historical data to simulate ocean numerical models could reduce model error and significantly reduce prediction error. Since it was proposed, Optimal Interpolation (OI), which is based on statistical estimation, has been applied to many research prediction systems [13,16]. An important research outcome is to master the covariance matrix of background error [20,21]. If effective results are received from satellite altimeter data modeling [22,25], the OI method does better in horizontal fields than in vertical sections.

In this paper, we try to introduce another ocean modeling by another light AI function approximation, i.e. Radial Basis Function Neural Networks, to simulate ocean parameters processing focusing on how to reduce modeling cost including time and space, with a balanced efficiency. This result is helpful for further parameter settings, especially when those required areas are not refined.

* Corresponding author.
E-mail address: wuwei@idsse.ac.cn

This paper aims to realize an intelligent identification model based on RBF neural networks by assimilating ocean parameters in the Area. The paper is organized as follows: Section 2 is a brief survey about related works. Section 3 is a preliminary study of RBF neural networks and data assimilation, especially regarding temperature, salinity, current and wind. The model simulation and results are given in Section 4. The conclusion and future discussion are given in Section 5.

2. Related Works

The multi-scale dimensional variation scheme is a helpful improvement from OI. By introducing nonlinear functions into data approximation, dynamic conditions could globally control the data, in which three-dimensional variation schemes (3DVAR) and four-dimensional variation schemes (4DVAR) are common forms. Because it is computationally affordable and statistically reliable, the 3DVAR is widely used in ocean data assimilation [8,15,18]. When applied to time series observational data, the 4DVAR method can optimize model parameters in the initial phase, including prediction systems in meteorology [5,6,28] and marine physics data fitting [12,24]. Kalman filtering aimed at downsizing model error by adjusting background error covariance [9], and a correlation error reduction (CER) algorithm is proposed to deal with insufficient ensemble variance and spurious ensemble correlations between observations and state variables. Increasing the size of the ensemble at the analysis time in order to boost the rank of its background-error covariance [1], estimate could improve the performance of an Ensemble Kalman filter [14]. Similarly, D. Erdal applied EnKf as an inversion method to jointly estimate spatial variable recharge and conductivity fields from head observations, but the success of this approach strongly depends on the prior knowledge that is assumed [7]. In recent years, the simulation of ocean modeling has become more precise and regional. Many up-to-date, interdisciplinary methods have been developed to optimize data processing, such as genetic algorithms and artificial intelligence [10,19]. The key point of this research is to reveal correlation theories between the mechanisms of ocean dynamics and algorithm convergence.

Simulation on ocean dynamics is hot, but acceptable results highly depend on a coupled ocean model. However, it is costly to run a specialized ocean model because of complex parameters relations, massive data training/adjusting, and expertized design from ocean scientist [2,3]. For example, the current of seawater, wind and temperature are basic parameters for basic environment analyzing. In traditional methods, high order equations on revealing how they are working are required, and several necessary helps of nearby ocean circumstances (boundary conditions) are necessary [4,11].

3. Preliminary Study

3.1. Radial Basis Function Neural Networks

The basic form of the neural network is shown in Figure 1. It is a simplified three-layer structure. The number of neurons in the input layer is fixed according to the parameters obtained, the hidden layer is decided by the designer, and the output layer is determined by the system requirements.

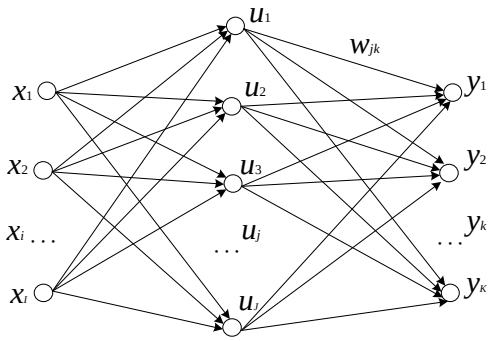

Figure 1. Simplified structure of RBF neural networks (3 layers with *I* input neurons, *J* hidden neurons and *K* output neurons)

As Figure 1 shows, assuming *M* nodes in the perception layer, $x = [x_1, x_2, \cdots, x_I]^T (x \in R^I)$, the number of mapping neurons in the hidden layer is *J* (only one hidden layer here), and the network output is $y = [y_1, y_2, \cdots, y_J]^T (y \in R^J)$.

The instant output of node *j* is acquired by computing each connected neuron x_i. The basic function of the RBF Neural Network is a distance metric, which is expected to be activated by the Gaussian function, in which the input data indicates radial symmetry. The data center c_j of each neuron enables this neuron to be responsive to an input numerical data.

If the activation function in *Gaussian* form, then the output of node *j* is achieved by Equation (1):

$$u_{ji}|_{x_i} = \exp(-\left\|x_i - c_j\right\|^2 / 2\sigma_j^2) \tag{1}$$

This data processing is nonlinear function mapping, and σ_j is the spread width of *j*, c_j is the normalized center. The nonlinear mapping process from $u_j(x)$ to y_k is accomplished through a linear sum by Equation (2):

$$y_k = \sum_{j=1}^{J} w_{jk} u_j + \theta_k, u_j = \sum_{i=1}^{I} u_{ji} \tag{2}$$

where w_{jk} is the adjusting weights from node *j* to the node *k*, and θ_k is the bias of output node *k*. y_k is the mapping results as they to the input data, and is to be transferred to a workspace depending on the application requirements.

The neuron weight is adjusted by Equation (3)

$$w_j(t+1) = w_j(t) - \eta \cdot \frac{\partial \varepsilon(t)}{\partial w_j(t)} \tag{3}$$

where *t* is the current operation, and η is the learning speed. The training target ε is calculated by Equation (4)

$$\varepsilon(t) = \alpha \cdot \sum_{k=1}^{L} (T_{ori}(t) - T_{out}(t))^2 \tag{4}$$

where α is the adjusted constant, and T_{ori}, T_{out} represent the real output and the model result, respectively.

3.2. Data assimilation foundations

The fundamental equation of statistical interpolation with least-squares estimation is a linear analysis method. Letting the dimension of the model state be *n* and the dimension of the observation vector be *p*, the optimal least-squares estimator is defined by the following interpolation Equations [5]:

$$x_a = x_b + K(y - H[x_b]) \tag{5}$$

where x_t is the true model state, x_b is the background model state, x_a is the analysis model state, *y* is the vector of observations, *H* is the observation operator, *B* is the covariance matrix of the background errors (x_b-x_t), *R* is the covariance matrix of observation errors (y-$H[x_t]$), and *A* is the covariance matrix of the analysis errors. Hereby, K = $BH^T\{(HBH^T + R)^{-1}$. For any given *K as* Equation (6),

$$A = (1 - KH)B(1 - KH)^T + KRK^T \tag{6}$$

If *K* is the optimal least-square gain, then *A* could be simplified to *A* = (1 - *KH*)*B*. Therefore, the least squares estimation is expressed by Equation (7)

$$x_a = arg\{minJ\} \tag{7}$$

where *J* is called the cost function of the analysis, in optimization algorithms, it is often achieved by error updating from iterations, which here could be expressed as Equation (8)

$$J(x) = (x - x_b)^T B^{-1}(x - x_b) + (y - H[x])^T R^{-1}(y - H[x]) \\ = J_b(x) + J_o(x) \tag{8}$$

where J_b is the background term, and J_o is the observation term.

4. RBFNN Identification Modeling

Regarding the observational data of various ocean parameters (temperature, salinity, current, wind), an identification model based on RBF neural networks is discussed in this section. In the first part, ocean data adaptively is explained as it is applied into a discrete time series normalization. The local vertical function that approximates certain positions based on RBFNN is established in the second part to test if it is a stationary time series. Finally, a regional intelligent simulation model is built to identify the selected parameters of the three-dimensional ocean system.

4.1. Data Description

The observational data ranging from January 1 to March 29 in 2016 is considered, including Salinity, U-component of current, V-component of current, Temperature, u-component of wind and v-component of wind. These are distributed within a bounding box by 4 degrees square from latitude to longitude. The data accumulates in the grid of 0.5 deg×0.5deg from 0.25E to 359.5E and 89.75N to 89.75S (720×361 Longitude/Latitude). A brief data description is given below in Table 1.

The first required operation for data exploration is not building corresponding models, but analyzing data quality. Data quality control is a preliminary procedure that must take place before further data mining. The dataset given in Table 1 contains too many missing values in levels 13 to 16 of the wind parameters, and in levels 30 to 40 of the current in the deep ocean. In normal procedures, interpolation methods would be used to fill in the missing points in these levels such as Lagrange interpolation and Newton interpolation. However, in this case, because those levels that are missing data are not critical for the model, the deep-sea levels from 30 to 40, ranging from -747m to -4478m, are too high to affect sea surface directly. Moreover, because the data of the wind from levels 13 to 16 are all missing beyond interpolation operations, those insignificant data were removed to form a new set (Table 2).

Table 1. Original observational dataset description

Data range	January 1, 2015 to March 30, 2016
Parameters	16 layers: Temperature, u-component of wind, v-component of wind 40 layers: Salinity, U-component of current, V-component of current
Grid	0.5deg×0.5deg from 0.25E to 359.75E and 89.75N to 89.75S
Time	0, 6, 12, 18h per day
Output format	netCDF
Bounding box	4 degrees span

Table 2. Controlled dataset description

Data range	January 1, 2015 to March 30, 2016
Wind	Temperature, u-component of wind, v-component of wind 12 layers: 10m to 1500m, 12 cells with 8×8×1460 data in each level
Current	Salinity, U-component of current, V-component of current 29 layers: -5m to -584m, 29 cells with 8×8×1460 data in each level
Grid	0.5deg×0.5deg from 0.25E to 359.75E and 89.75N to 89.75S
Time	0, 6, 12, 18h per day
Output format	netCDF
Bounding box	4 degrees span

It is not reasonable to identify all data sets through the timeline. Therefore, represented levels are selected instead. The wind and current distribution characteristic diagram is given below in Figure 2. The wind at 10m above sea level and the current at -5m were selected. The directions are shown at day 1, day 10, day 20, day 30, day 40, day 50, day 60, day 70 and day 80.

As illustrated in Figure 2, the direction of the wind and current may not kept the same, but the angle between them is connected at a constant angle, 45 degrees in this case. The wind variation fits the seasonal vibration of approximately 50 days (from day 1 to day 50). Meanwhile, the current change from day 1 to day 50 indicates similar trends. The direction between the current and the wind was not constant. This phenomenon is caused by the delay between active wind (with different strength) movements and passive current response. Normally, the current is derived from persistent wind excitation of immediate wind (although there may be some delays transferred to the deep levels of less than 2 weeks).

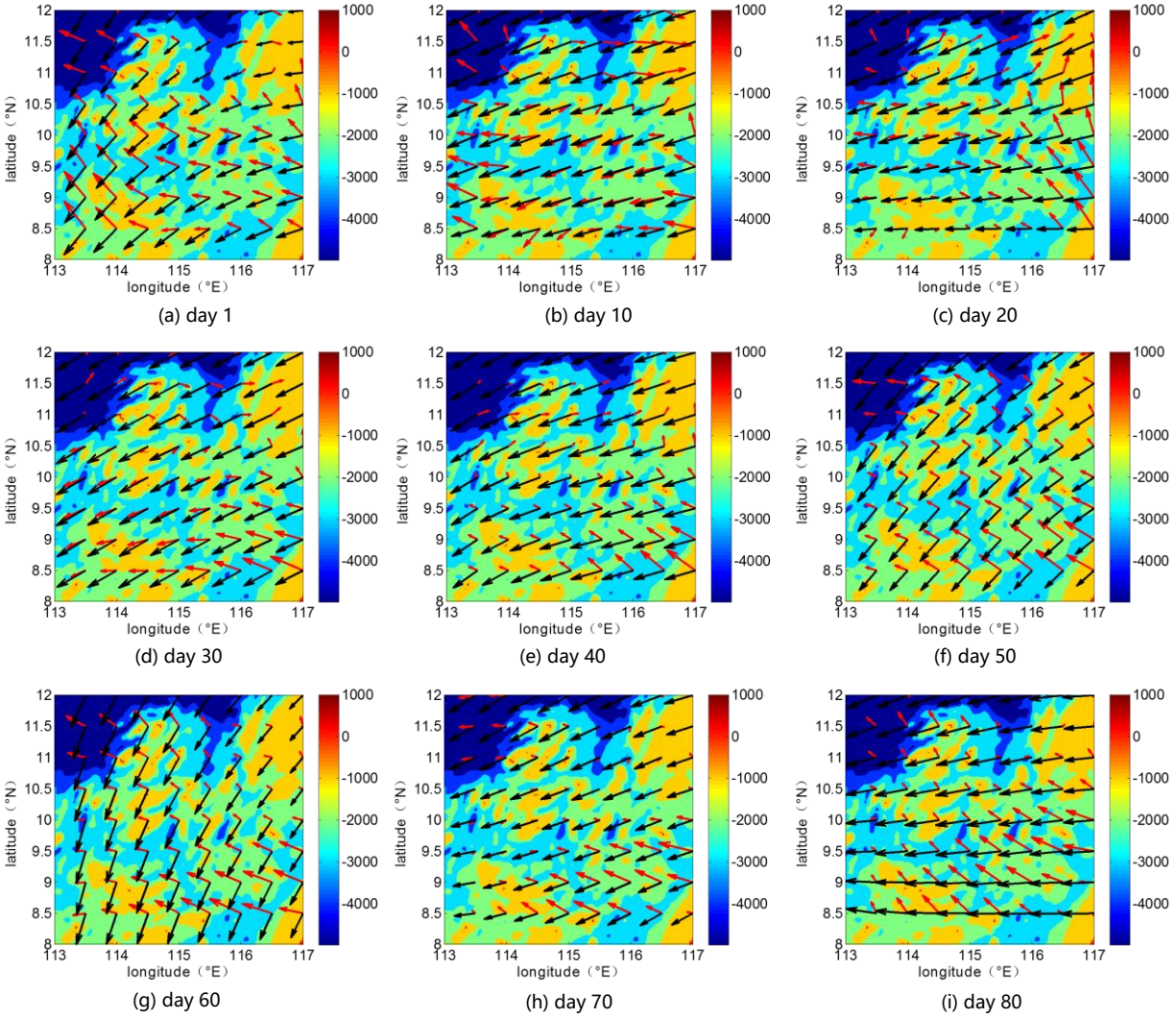

Figure 2. Daily average parameters distribution diagram (January 1, 2016 to March 20, 2016): Wind distribution at level 1 (height=10m) in black color; Current distribution at level 29 (depth=-5m) in red color.

As proposed in part 1 of section 3, the current and wind are expected to be stable time series data under the conditions of persistent excitation, which can guarantee the further identification modeling. However, if it is found to be a stochastic process, it may be impossible to abstract the underlying message in the discrete data set. ADF testing is a necessary method to test the data characteristics. The stationary time series could be achieved by $EX_t = \mu$, and EX_tX_{t+k} is not affected by t. Where t and $k=0, \pm1, \pm2,\ldots$, the cross-correlation function ρ is calculated by Equation (9)

$$\rho_{XY} = \frac{Cov(X,Y)}{\sqrt{DX}\sqrt{DY}} \tag{9}$$

Table 3. Pearson correlation coefficient test between u component of wind and current.
(**.α_1=0.01, **.α_2=0.05, double significance correlation)

Test		uwind01	uwind08	uwind16	uwind24	uwind32	uwind40	uwind48	uwind56	uwind64
Ucur01	Corr.	.311**	.148**	.085	.030	-.016	-.056	-.088	-.107*	-.115*
	sig.	.000	.005	.107	.567	.765	.288	.097	.044	.030
Ucur08	Corr.	-.099	-.167**	-.198**	-.226**	-.249**	-.258**	-.257**	-.248**	-.236**
	sig.	.062	.001	.000	.000	.000	.000	.000	.000	.000
Ucur16	Corr.	-.091	-.163**	-.189**	-.215**	-.237**	-.246**	-.246**	-.238**	-.229**
	sig.	.086	.002	.000	.000	.000	.000	.000	.000	.000
Ucur24	Corr.	-.098	-.171**	-.194**	-.215**	-.236**	-.244**	-.244**	-.238**	-.229**
	sig.	.064	.001	.000	.000	.000	.000	.000	.000	.000
Ucur32	Corr.	-.097	-.169**	-.190**	-.210**	-.229**	-.237**	-.238**	-.233**	-.224**
	sig.	.067	.001	.000	.000	.000	.000	.000	.000	.000
Ucur40	Corr.	-.042	-.112*	-.138**	-.163**	-.186**	-.198**	-.204**	-.202**	-.195**
	sig.	.433	.034	.009	.002	.000	.000	.000	.000	.000
Ucur48	Corr.	.063	.019	-.015	-.049	-.081	-.101	-.114*	-.119*	-.118*
	sig.	.234	.716	.774	.352	.127	.056	.031	.024	.026
Ucur56	Corr.	.144**	.149**	.110*	.071	.035	.008	-.011	-.022	-.024
	sig.	.006	.005	.038	.180	.512	.876	.838	.682	.654
Ucur64	Corr.	.195**	.240**	.210**	.178**	.148**	.124*	.107*	.096	.093
	sig.	.000	.000	.000	.001	.005	.019	.043	.069	.079
Vcur01	Corr.	.075	.219**	.217**	.212**	.198**	.180**	.164**	.151**	.152**
	sig.	.159	.000	.000	.000	.000	.001	.002	.004	.004
Vcur08	Corr.	-.330**	-.315**	-.324**	-.328**	-.332**	-.332**	-.331**	-.334**	-.334**
	sig.	.000	.000	.000	.000	.000	.000	.000	.000	.000
Vcur16	Corr.	-.176**	-.159**	-.177**	-.190**	-.202**	-.210**	-.216**	-.222**	-.220**
	sig.	.001	.003	.001	.000	.000	.000	.000	.000	.000
Vcur24	Corr.	-.038	-.032	-.063	-.086	-.103	-.119*	-.133*	-.143**	-.143**
	sig.	.472	.545	.239	.106	.051	.024	.012	.007	.007
Vcur32	Corr.	-.029	-.050	-.090	-.119*	-.140**	-.160**	-.179**	-.194**	-.197**
	sig.	.582	.351	.090	.024	.008	.002	.001	.000	.000
Vcur40	Corr.	-.088	-.147**	-.196**	-.232**	-.259**	-.284**	-.307**	-.327**	-.336**
	sig.	.097	.005	.000	.000	.000	.000	.000	.000	.000
Vcur48	Corr.	-.086	-.176**	-.232**	-.278**	-.313**	-.344**	-.371**	-.394**	-.407**
	sig.	.104	.001	.000	.000	.000	.000	.000	.000	.000
Vcur56	Corr.	-.032	-.116*	-.180**	-.233**	-.276**	-.309**	-.338**	-.361**	-.378**
	sig.	.544	.029	.001	.000	.000	.000	.000	.000	.000
Vcur64	Corr.	-.038	-.062	-.115*	-.157**	-.190**	-.215**	-.237**	-.257**	-.271**
	sig.	.480	.244	.030	.003	.000	.000	.000	.000	.000

Table 4. Pearson correlation coefficient test between v component of wind and current.
(**.α_1=0.01, **.α_2=0.05, double significance correlation)

Test		vwind01	vwind08	vwind16	vwind24	vwind32	vwind40	vwind48	vwind56	vwind64
Ucur01	Corr.	.639**	.597**	.599**	.594**	.583**	.564**	.541**	.519**	.502**
	sig.	.000	.000	.000	.000	.000	.000	.000	.000	.000
Ucur08	Corr.	.413**	.502**	.500**	.495**	.488**	.477**	.463**	.446**	.428**
	sig.	.000	.000	.000	.000	.000	.000	.000	.000	.000
Ucur16	Corr.	.379**	.468**	.469**	.466**	.462**	.453**	.441**	.425**	.407**
	sig.	.000	.000	.000	.000	.000	.000	.000	.000	.000
Ucur24	Corr.	.351**	.443**	.446**	.445**	.441**	.431**	.417**	.398**	.379**
	sig.	.000	.000	.000	.000	.000	.000	.000	.000	.000
Ucur32	Corr.	.367**	.453**	.455**	.452**	.445**	.432**	.415**	.393**	.372**
	sig.	.000	.000	.000	.000	.000	.000	.000	.000	.000
Ucur40	Corr.	.434**	.494**	.494**	.488**	.479**	.463**	.444**	.423**	.403**
	sig.	.000	.000	.000	.000	.000	.000	.000	.000	.000
Ucur48	Corr.	.495**	.519**	.519**	.515**	.509**	.497**	.484**	.471**	.460**
	sig.	.000	.000	.000	.000	.000	.000	.000	.000	.000
Ucur56	Corr.	.480**	.478**	.482**	.485**	.488**	.487**	.486**	.487**	.488**
	sig.	.000	.000	.000	.000	.000	.000	.000	.000	.000
Ucur64	Corr.	.391**	.376**	.380**	.385**	.394**	.400**	.407**	.418**	.426**
	sig.	.000	.000	.000	.000	.000	.000	.000	.000	.000
Vcur01	Corr.	.211**	.181**	.184**	.188**	.194**	.198**	.202**	.207**	.218**
	sig.	.000	.001	.000	.000	.000	.000	.000	.000	.000
Vcur08	Corr.	-.017	.066	.079	.090	.095	.098	.094	.080	.063
	sig.	.746	.213	.137	.089	.072	.065	.077	.134	.236
Vcur16	Corr.	.150**	.191**	.203**	.214**	.222**	.230**	.233**	.229**	.220**
	sig.	.004	.000	.000	.000	.000	.000	.000	.000	.000
Vcur24	Corr.	.229**	.226**	.242**	.258**	.274**	.291**	.304**	.310**	.308**
	sig.	.000	.000	.000	.000	.000	.000	.000	.000	.000
Vcur32	Corr.	.195**	.183**	.207**	.232**	.253**	.274**	.289**	.294**	.290**
	sig.	.000	.001	.000	.000	.000	.000	.000	.000	.000
Vcur40	Corr.	.182**	.177**	.206**	.233**	.252**	.268**	.275**	.270**	.257**
	sig.	.001	.001	.000	.000	.000	.000	.000	.000	.000
Vcur48	Corr.	.236**	.235**	.262**	.287**	.304**	.315**	.318**	.309**	.295**
	sig.	.000	.000	.000	.000	.000	.000	.000	.000	.000
Vcur56	Corr.	.255**	.263**	.292**	.319**	.342**	.360**	.371**	.373**	.370**
	sig.	.000	.000	.000	.000	.000	.000	.000	.000	.000
Vcur64	Corr.	.128*	.136**	.166**	.195**	.220**	.242**	.259**	.269**	.273**
	sig.	.016	.010	.002	.000	.000	.000	.000	.000	.000

$$Cov(X,Y) = E[X - E(X)][Y - E(Y)] \tag{10}$$

where $\sigma = \sqrt{D}$, series X_t and Y_t satisfy the stationary conditions as Equation (10). The Pearson correlation coefficient is introduced to reveal the correlation between the wind and the current. Regarding one level of wind or current, 64 points would be superfluous for a double significance check. Instead, points 1, 8, 16, 24, 32, 40, 48, 56, 64 were chosen, and the results are shown in Table 3 and Table 4.

As Table 3 shows, the u component of the wind and the U component of the current were closely related in most positions except Ucur48. This may have been caused by the reef islands in the vulnerable shallow water, which are easily affected by instant variations from all directions. Thus, the impact from the wind to the u current is large-scale and influential. At the same time, Table 4 shows that the influence of the v wind on the current was much more significant than that of the u wind. Still, the u and v components of the current were both affected by the entire wind level, as represented by correlation functions in both α_1=0.01 and α_2=0.05.

4.2. Modeling Simulation

From the aforementioned discussions, the u and v components of the current were closely related to the u and v components of the wind at 10m. This correlation mode ensures the basic foundations of the neural network identification; i.e., there is a certain identification mapping process for current derived from wind actions. Let a temporal data sequence S_ζ be produced by system Y_ζ, where Y_ζ is measurable and bounded, $Y_\zeta \in \Omega$, Ω is a compact set, and the system status is a regression deterministic tracking [17,26].

$$\begin{aligned} &S_\xi \to Y_\xi, S_\xi = [S(1), S(2), \cdots, S(n)]^T, \\ &Y_\xi = [y_1, y_2, \cdots, y_L]^T \in R \end{aligned} \tag{11}$$

With regard to the aforementioned recorded time series, the current \$T\$ is a bounded series, and all variables are varying in limited bounds as Equation (11), i.e. letting $T_{\zeta \to T}$, $T = [t(1), t(2), ..., t(n)] \in R^{m \times n}$, where $T_\xi = [t_\xi(1), t_\xi(2), ..., t_\xi(m)]$, and $t_\xi(i) = [t_{\xi i}(1), t_{\xi i}(2), ..., t_{\xi i}(n)]^T$.

Let the *u* and *v* components of the wind form a parameter matrix $P_{(128\times1460)}$, and let the current be target matrix $T_{(128\times1460)}$. Then, the system function is determined by Equation (12)

$$\begin{aligned} &F : P \to T, \\ &F_\xi : P\xi \to T_\xi, \lim_{t \to t_0} f(T_\xi, T) = \epsilon \end{aligned} \tag{12}$$

System *F* is the given form of true transmission from *P* to *T* (data pairs). The goal is to build an approximate and robust F_ζ to replace the unknown real system with *n* acceptable error ε. This error could be calculated as Mean Squared Error. All processes are expected to finish in limited time t_0 as Equation (13).

$$MSE = f(T_\xi, T) = \frac{\left\| T_\xi - T \right\|}{T} \times 100\% \tag{13}$$

Model prediction methods by offline training gain solid ground in the area of online forecasting. Online prediction models are often derived from offline ones, and rely heavily on training and preliminary results [25]. However, the goal of this new model is to predict current with a given dataset and an offline model design. This can provide a useful structure for further, more advanced models.

The entire process includes 3 steps (shown in Figure 3). In the initial phase, the network structure is determined by the variables of input, output and system complexity. The input layer is finalized by the relative parameters extracted from the rough data set, and the output layer is confirmed by the workspace of real requirements. The hidden layer is not limited as a constant number because it increases with larger data sets and more uncertain system disturbances. Previous researchers

have found that an increased number of hidden neurons improves the accuracy of the system, though this comes at the expense of more computation time.

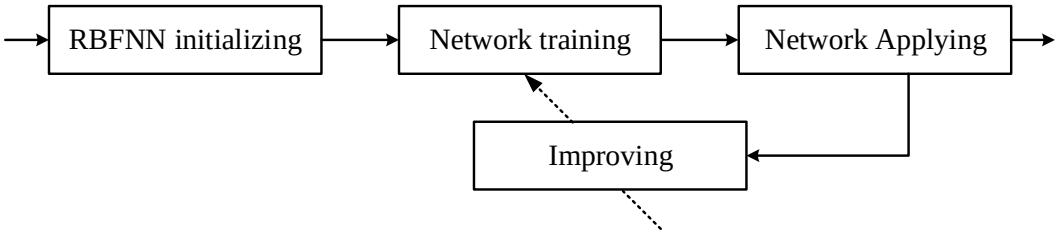

Figure 3. RBFNN training and applying processing

As long as the network structure is initialized, the second step uses the given data set to train the network until it is converged to a small error (initialized by the designer). For example, the margin between the real observational data and the simulation results is acceptable. In normal situations, the training sample should not be small, in order to make sure that the network is error-tolerant and widely applicable. The network application is the target of the network design. Therefore, a specific network was trained according to real circumstances. The trained network is saved from the finalized neurons, weights, and connections. The network can be tested by applying it to real known data inputs, and compare the results between the network output(s) and the reference data. If the compared results are acceptable, the model can be applied in other similar circumstances.

4.3. Modeling evaluation

In the previous part, the given data set is simulated with an increasing number of hidden neurons in three cases. The network structure is given as 128 inputs I=128 because the correlation between the u/v component of the wind and the U/V component of the current is closely related in Table 3 and 4. The hidden neurons were unutilized at J=1200, 1300, 1400, and the output was K=128. The relative error was introduced to evaluate the system performance as the error controlled at $\varepsilon=1\times10^{-4}$. The experiment is conducted by the Lenovo think station A8000F series and IBM server X3950 X6.

The simulation results are shown in Figure 4. The simulation results in each column run a case of current data, including the comparison between the real data and the network output of the U/V component of the current, and the relative error. Meanwhile, data from each row provides a comparison of the same parameters in different cases. The third row clearly shows that the simulation error decreased from both the relative error value and the vibration frequency. To be specific, in (d), (e) and (f), the relative error of the u component of the current at sample axis (200, 700, 900 and 1250) gradually decreased as the amount of hidden neurons increased. More obviously, from (j), (k) and (l), the relative error vibration magnitude increased by a small margin, though the vibration frequency significantly decreased. The relative error increased at time line 600 to 1200, though it is satisfied from May to October. This indicates that the current and wind characteristics in the selected Area are not always stable because of extra strong influences during this period.

A more visualized current result is shown in Figure 5(a), and the relative simulation error of the three different scales of neurons are shown in Figure 5 (b), (c) and (d). The direction of the current is represented by a value above or below zero, and the normalized current is smoothed in Figure 5 (a). The relative error of the network output is given in the rest subfigures from Figure 5 (b) to (d). The maximum error is whittled down to zero, and the vibration frequency has been cut down gradually. The MSE is given in Table 5. As the neural network became more accurate, it decreased from 1.039×10^{-2} to 1.676×10^{-3}.

Table 5. MSE comparison among different number of hidden neurons

case	hidden neurons	MSE	Network training time (sec.)
1	1200	1.039×10^{-2}	1143.17
2	1300	5.734×10^{-3}	1235.22
3	1400	1.676×10^{-3}	1321.52

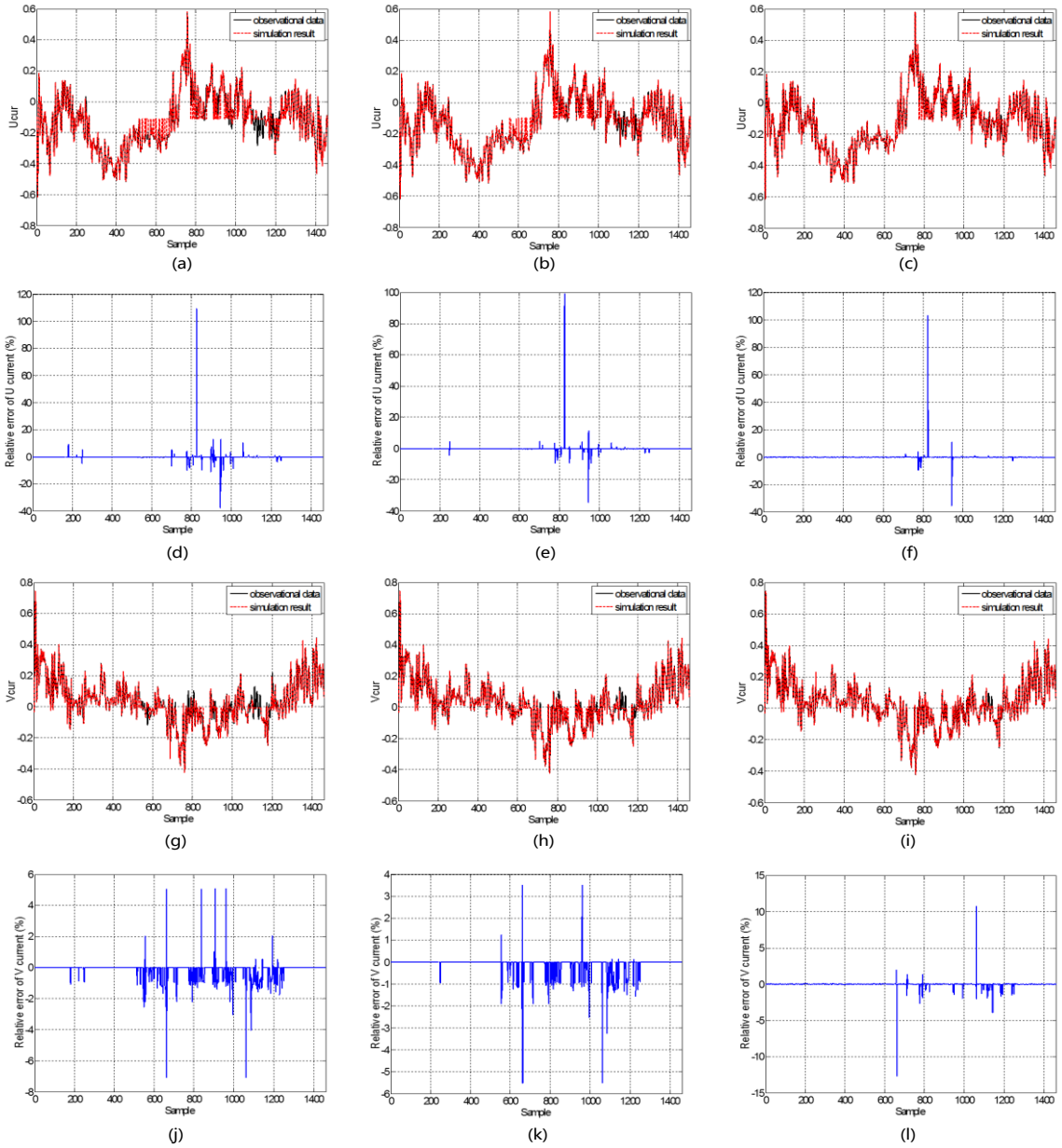

Figure 4. System identification modeling based on RBF Neural Networks: $\varepsilon=1\times10^{-4}$, *spread*=1, I=128. case 1: J_1=1200, (a)Comparison of U component of current between real data and network results, (d) Relative error of U component of current, (g)Comparison of V component of current between real data and network results, (j) Relative error of V component of current; case 2: J_2=1300, (b)Comparison of U component of current between real data and network results, (e) Relative error of U component of current, (h)Comparison of V component of current between real data and network results, (k) Relative error of V component of current; case 3: J_3=1400, (c)Comparison of U component of current between real data and network results, (f) Relative error of U component of current, (i)Comparison of V component of current between real data and network results, (l) Relative error of V component of current.

5. Conclusions

Using RBF Neural Networks to simulate ocean data is not new, but the technique is still far from fully realized. In this paper, the correlation method was used to analyze the connections between current data and wind data. They were shown to be highly interactive, especially in some specific areas. The simulation results that used RBFNN with different numbers of hidden neurons provided a small relative error between real data and network outputs. However, the model still needs to be tested and improved by more real circumstances. For example, the accuracy and adaptive capacity of using a year data to build a model have not yet been confirmed, and it may be necessary to consider more influential factors in order to obtain the target.

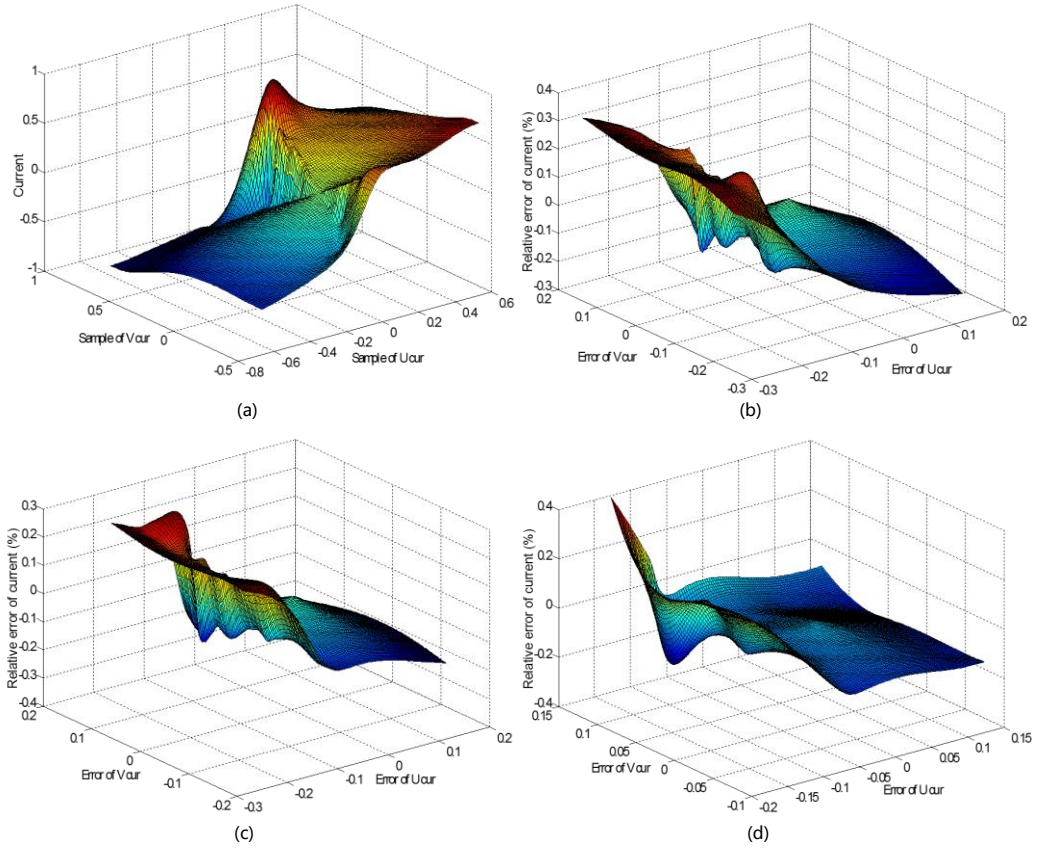

Figure 5. System identification modeling based on RBF Neural Networks: ε=1×10^{-4}, *spread*=1, *I*=128. (a) Current (real normalized data, both *u* component and *v* component are included); (b) J_1=1200, simulation error; (c) J_2=1300, simulation error; (d) J_3=1400, simulation error.

Acknowledgements

This research was supported by the Project of CAS grant NO. Y610141BRC, Innovation Foundation of CAS grant NO. CXJJ-17-M126, Major science and technology projects of Hainan Province NO. ZDKJ2016014, and Natural Science Foundation of Hainan Province grant NO. 417211.

References

1. J.L. Anderson, “Reducing Correlation Sampling Error in Ensemble Kalman Filter Data Assimilation,” Monthly Weather Review, vol. 144, no. 3, pp. 913–925, 2015.
2. F. Bouttier and P. Courtier, “Data Assimilation Concepts and Methods March 1999,” Meteorological training course lecture series. ECMWF, 2002.
3. J.S. Brain, Isaac M. Held. “An Assessment of Climate Feedbacks in Coupled Ocean–Atmosphere Models,” Journal of Climate, Vol. 19, pp. 3354-3357, 2006.
4. L. Bopp, L. Resplandy, et al. “Multiple Stressors of Ocean Ecosystems in the 21st Century: Projections with CMIP5 models,” Biogeosciences, 10, 6225 -6245, 2013.
5. M. Bocquet, H. Elbern, et al. “Data Assimilation in Atmospheric Chemistry Models: Current Status and Future Prospects for Coupled Chemistry Meteorology Models,” Atmospheric chemistry and physics, vol. 15, no. 10, pp. 5325–5358, 2015.
6. M. Buehner, J. Morneau, and C. Charette, “Four-dimensional Ensemble-variational Data Assimilation for Global Deterministic Weather Prediction,” Nonlinear Processes in Geophysics, vol. 20, no. 5, pp. 669–682, 2013.
7. D. Erdal and O. Cirpka, “Joint Inference of Groundwater-recharge and Hydraulic-conductivity Fields from Head Data Using the Ensemble Kalman filter,” Hydrology and Earth System Sciences, vol. 20, no. 1, pp. 555–569, 2016.
8. C. A. Edwards, A. M. Moore, I. Hoteit, and B. D. Cornuelle, “Regional Ocean Data Assimilation,” Annual review of marine science, vol. 7, pp. 21–42, 2015.
9. G. Evensen, “Data assimilation: the ensemble Kalman filter,” springer Science & Business Media, 2009.
10. G. Ferri, M. Cococcioni, and A. Alvarez, “Mission Planning and Decision Support for Underwater Glider Networks: A sampling on-demand approach,” Sensors, vol. 16, no. 1, p. 28, 2015.
11. S.M. Griffies, Michael Winton et al. “Impacts on Ocean Heat from Transient Mesoscale Eddies in a Hierarchy of Climate

Models," Journal of Climate, Vol. 28, pp. 952-970, 2015.
12. I. Iermano, A. Moore, and E. Zambianchi, "Impacts of a 4-dimensional Variational Data Assimilation in a Coastal Ocean Model of Southern Tyrrhenian sea," Journal of Marine Systems, vol. 154, pp. 157–171, 2016.
13. C. James, C. Gennady, C. Xianhe, and G. Benjamin, "A Simple Ocean Data Assimilation Analysis of the Global Upper Ocean 1950-95. part i: Methodology," Journal of Physical Oceanography, vol. 30, no. 2, pp.294–309, 2000.
14. M. Kretschmer, B. R. Hunt, and E. Ott, "Data Assimilation Using a Climatologically Augmented Local Ensemble Transform Kalman filter," Tellus A, vol. 67, pp. 1–5, 2015.
15. Z.J. Li, James C. McWilliams, et al. "Coastal Ocean Data Assimilation Using a Multi-scale Three-dimensional Variational Scheme," Ocean Dynamics, 65(7), 1001-1015, 2015.
16. Y. Liu, H. Meier, and L. Axell, "Reanalyzing Temperature and Salinity on Decadal Time Scales Using the Ensemble Optimal Interpolation Data Assimilation Method and a 3D Ocean Circulation Model of the Baltic sea," Journal of Geophysical Research: Oceans, vol. 118, no. 10, pp. 5536–5554, 2013.
17. S. Mohanty, A. Chattopadhyay, P. Peralta, and S. Das, "Bayesian Statistic Based Multivariate Gaussian Process Approach for Offline/online Fatigue Crack Growth Prediction," Experimental mechanics, vol. 51, no. 6, pp.833–843, 2011.
18. P.A. Muscarella, M. Carrier, and H. Ngodock, "An Examination of a Multi-scale Three-dimensional Variational Data Assimilation Scheme in the Kuroshio Extension Using the Naval Coastal Ocean Model," Continental Shelf Research, vol. 73, pp. 41–48, 2014.
19. A. Mozaffari, K. A. Scott, S. Chenouri, and N. L. Azad, "A Modular Ridge Randomized Neural Network with Differential Evolutionary Distributor Applied to the Estimation of Sea Ice Thickness," Soft Computing, pp. 1–25, 2016.
20. P.R. Oke, G. B. Brassington, D. A. Griffin, and A. Schiller, "The Bluelink Ocean Data Assimilation System (bodas)," Ocean Modelling, vol. 21, no. 1, pp. 46–70, 2008.
21. Q. Peng and C. Lei., "The Assimilation of Jason-2 Significant Wave Height Data in The North Indian Ocean Using the Ensemble Optimal Interpolation," IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no. 1, pp. 287–297, August 2016.
22. D. Paiva, R. C., M. T. Durand, and F. Hossain, "Spatiotemporal Interpolation of Discharge Across a River Network by Using Synthetic Swot Satellite Data," Water Resources Research, vol. 51, no. 1, pp. 430–449, 2015.
23. S. Suranjana, Moorthi, et al., "The Ncep Climate Forecast System Reanalysis," Bulletin of the American Meteorological Society, vol. 91, no. 8, pp. 1015–1057, August 2010.
24. N. Usui, Y. Fujii, K. Sakamoto, and M. Kamachii, "Development of a Four-dimensional Variational Assimilation System for Coastal Data Assimilation around Japan," Monthly Weather Review, vol. 143, no. 10, pp. 3874–3892, 2015.
25. C. Ubelmann, P. Klein, and L. L. Fu, "Dynamic Interpolation of Sea Surface Height and Potential Applications for Future High-resolution Altimetry Mapping," Journal of Atmospheric and Oceanic Technology, vol. 32, no. 1, pp. 177–184, 2015.
26. C. W. and D. Hill, "Deterministic Learning and Rapid Dynamical Pattern Recognition," IEEE Transactions on Neural Networks, vol. 18, no. 3, pp. 617–630, 2007.
27. W.Z. Zhang, M. Hao, M. Snir. Predicting HPC Parallel Program Performance Based on LLVM compiler. Cluster Computing, 20(2), 1179-1192, 2017.
28. M. Zhang and F. Zhang, "E4dvar: Coupling an Ensemble Kalman filter with Four-dimensional Variational Data Assimilation in a Limited-area Weather Prediction Model," Monthly Weather Review, vol. 140, no. 2, pp. 587–600, 2012.

Li Yuan graduated from the School of Physics, Hunan Normal University, for the degree of Master, her research interest covers Artificial Intelligence, data and knowledge engineering. Now she is an associate professor of the School of Physics, Hainan Medical University, China.

Wei Wu received his Master degree from the College of Information Science and Technology, Hainan University. He joined the Institute of Deep-sea Science and Engineering, Chinese Academy of Sciences since 2015. His research interests include Ocean Engineering, Artificial Intelligence and machine learning.

Chuan Tian received his M.S. and the Ph.D. degree from Tianjin University, he is an associate researcher of the Institute of Deep-sea Science and Engineering, Chinese Academy of Sciences His research interests include Ocean Science and Technology.