Neurocomputing 128 (2014) 59–72

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Constructive multi-output extreme learning machine with application to large tanker motion dynamics identification

Ning Wang [a,b,*], Min Han [b], Nuo Dong [a], Meng Joo Er [a,c]

[a] *Marine Engineering College, Dalian Maritime University, Dalian 116026, PR China*
[b] *Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, PR China*
[c] *School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore*

ARTICLE INFO

Article history:
Received 27 August 2012
Received in revised form
7 January 2013
Accepted 21 January 2013
Available online 25 October 2013

Keywords:
Extreme learning machine
Constructive method
Improved multi-response sparse regression
Multi-output regression
Tanker motion dynamics

ABSTRACT

In this paper, a novel constructive multi-output extreme learning machine (CM-ELM) is proposed to deal with a large tanker motion dynamics identification. The significant contributions are as follows. (1) Driven by generated tanker dynamics data from the reference model, the CM-ELM method is proposed to identify multi-output dynamic models. (2) The candidate pool for CM-ELM is randomly generated by the ELM strategy, and ranked chunk-by-chunk based on a novel improved multi-response sparse regression (I-MRSR) incorporated with λ weighting. (3) Consequently, the constructive model selection works with fast speed due to chunk-type training process, which also benefits stable hidden node selection and corresponding generalization. (4) Furthermore, output weight update on the final CM-ELM model randomly selected from the elite subset is conducted to enhance the overall performance of the resulting CM-ELM scheme. Finally, the convincing performance of the complete CM-ELM paradigm is verified by simulation studies on not only tanker motion dynamics identification but also benchmark multi-output regressions. Comprehensive comparisons of the CM-ELM with ELM and OP-ELM indicate the remarkable superiority in terms of generalization capability and stable compact structure. Conclusions are steadily drawn that the CM-ELM method is feasibly effective for tanker motion dynamics identification and multi-output regressions.

1. Introduction

On the one hand, the promising extreme learning machine (ELM) method for single-hidden layer feedforward networks (SLFNs) has attracted comprehensive and intensive research since Huang et al. [1] proposed the seminal work. The main idea of ELM strategy is intuitively realized as follows. Hidden nodes are randomly generated and output weights are analytically determined by pseudo-inverse technologies. It is evident that the ELM is an extremely fast batch learning algorithm and can provide good generalization performance [2]. As a consequence, the ELM does not need any iterations to determine the hidden node parameters, and dramatically reduces the computational time for training process. Actually, the randomness and diversity of hidden nodes should be guaranteed for high generalization performance. In this case, the determination for the suitable or optimal number of randomly generated hidden nodes becomes an interesting and critical issue to elaborate the ELM advantages. However, the original ELM [3] does not provide any effective solution for architectural design of the network. In most cases, the suitable number of hidden nodes is pre-defined by a trial and error method, which may be tedious in some applications.

In order to circumvent the above-mentioned problems, some improvements to the ELM for optimal structure have been proposed in two heuristic approaches, i.e., destructive and constructive methods, which have been effectively implemented in fuzzy neural networks [4–6]. For the former approach, Rong et al. [7] have presented a pruned ELM (P-ELM), for classification problems, which starts with a large network and then eliminates the hidden nodes having low relevance to the output. Miche et al. [8] have proposed an optimally pruned ELM (OP-ELM) by using the multi-response sparse regression (MRSR) algorithm [9] and leave-one-out (LOO) validation for pruning strategy. Evidently, these methods within destructive paradigms would face common difficulties that the algorithm starts with a large scale structure which inevitably increase the computational burden. For the latter approach, the incremental extreme learning machine (I-ELM) [2] and its variants [10,11] proposed by Huang et al. are proposed to add hidden nodes one-by-one to the hidden layer and incrementally update output weights. However, those algorithms cannot lead to an optimal network structure automatically, and hidden

[*] Corresponding author at: Marine Engineering College, Dalian Maritime University, Dalian 116026, PR China. Tel.: +86 134 785 10177.
E-mail address: n.wang.dmu.cn@gmail.com (N. Wang).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.01.062

nodes are added to the SLFN merely in one-by-one manner. Feng et al. [12] have proposed the error minimized extreme learning machine (EM-ELM), which can add random hidden nodes one-by-one or group-by-group. Unfortunately, the nodes added into the hidden layer are randomly generated and might deteriorate the performance with increasing hidden nodes since no generalization measure is guaranteed. Nevertheless, the resultant network structure would be much similar to original ELM if high prediction performance is required. A constructive hidden nodes selection of extreme learning machine termed as CS-ELM is proposed by Lan et al. [13], whereby the hidden nodes are selected by the MRSR and unbiased risk estimation based criterion C_p. However, the CS-ELM works for single-output regression and the hidden node selection conducts in one-by-one manner.

On the other hand, as the potential application in this paper, the large tanker maneuvering dynamics play a fundamental role in the whole guidance, navigation and control (GNC) system. Focusing on this essential issue, many researchers have proposed varieties of vessel motion models, mainly including Abkowitz model, MMG model and response model [14]. These three types of models preserve distinct features as follows. Abkowitz model pursues accurate hydrodynamic derivatives at the cost of clear presentation for variables, and therefore results in difficulties for control system design. To the contrary, MMG model and response model focus on analysis and synthesis of model based control systems while the model accuracy would be lower since MMG and response models could be considered as simplifications of the Abkowitz model to some extent.

Within the previous model frameworks of vessel motion, studies on system identification for hydrodynamic derivatives and input–output nonlinearities have been conducted by using various methods, i.e., simplified linearization [15], estimation-before-modeling technique [16], and support vector regression method [17], *etc*. However, the resultant overall mathematical formulation of vessel maneuvering is usually complicated due to the existence of hydrodynamic nonlinearities associated with the vessel dynamics. In this case, there exists a dilemma between the accuracy and interpretation of vessel motion models using traditional methods.

In addition, the use of large tankers becomes an important issue since the demand of transportation for crude oil has increased. System identification for large tanker motion dynamics becomes an involved task due to the maneuverability difficulties caused by their bulk. Unfortunately, comparing with the previous investigations of general vessels, promising results of large tanker maneuvering models are short of appearance and mainly focus on controller design rather than motion dynamics identification [18]. Typically, van Berlekom made an excellent seminal research on Esso Osaka tanker model, whereby the hydrodynamic derivatives have been proposed in detail [14].

Recently, in order to overcome the above-mentioned problems, researchers appeal to artificial neural networks (ANNs) in the field of artificial intelligence technology which can be used to establish nonlinear input–output models for ship maneuvering motion effectively. Mahfouz and Haddara [19] applied the ANN and spectral analysis methods to identify the hydrodynamic derivatives in the mathematical model of marine vehicle motions. Moreira and Guedes Soares [20] proposed a dynamic recurrent neural network (RNN) based maneuvering simulation model for surface ships. Rajesh et al. [18] identified an interesting nonlinear maneuvering model of large tankers based on back propagation (BP) neural networks. Certainly, the ANN based system identification method could obtain considerable performance for approximation and generalization. However, the nonlinearities underlying between input and output variables would also be folded into a “black box” which is difficult to be interpreted and understood.

Motivated by the previous reviews, we present a novel constructive multi-output extreme learning machine (CM-ELM) for large tanker motion dynamics identification in this paper. The underlying main idea could be implemented as follows. A group of well established nonlinear differential equations for tanker motion dynamics are used as the reference model for training and testing data generation. In this case, the dynamics identification is equal to a multi-output regression problem that the states, i.e., surge, sway, yaw speed and rudder angle (u, v, r, δ), are input variables, and state derivatives (Δu, Δv, Δr) are taken as multiple outputs. With data samples at hand, it is followed by the promising data-driven learning method term as CM-ELM for multi-output regression. A candidate pool of hidden nodes in the SLFN is randomly generated by the ELM strategy in the initial phase, and then hidden node ranking and model selection from the candidate pool is implemented by a novel improved MRSR (I-MRSR) method and generalization measure. In the last phase, from the elite subset of model selections, the resulting CM-ELM model is randomly selected to update the output weight based on the whole training data. It should be noted that the proposed CM-ELM method ranks and adds candidate hidden nodes chunk-by-chunk, rather than one-by-one, which would reasonably reduce computational burden and accelerate learning speed. Simulation studies on benchmark multi-output regression datasets validate the effectiveness and superiority of the CM-ELM compared with ELM and OP-ELM, *etc*. In order to evaluate the CM-ELM application performance of large tanker motion dynamics identification, comprehensive simulations and comparisons of typical maneuvering scenarios are conducted on sine rudder angle input and zigzag maneuvers. The results demonstrate that compared with the ELM, the CM-ELM based tanker motion model with parsimonious structure achieves much promising identification and generalization performance in terms of both moderate and extreme maneuvering.

The rest of this paper is organized as follows. Section 2 briefly presents preliminary formulations of related works. The main idea and contributions to the CM-ELM including candidate pool generation, I-MRSR based ranking, constructive model selection and generalization measure are unfolded in Section 3. Section 4 implements simulation studies on the CM-ELM for benchmark dataset experiments and applications to tanker motion dynamics identification in detail. In Section 5, conclusions are drawn.

2. Preliminary formulation

In this section, the preliminary formulation of related works, i.e., extreme learning machine (ELM) and multi-response sparse regression (MRSR), will be briefly presented to enhance the foundation knowledge of our proposed learning scheme, which is applied to large tanker motion dynamics identification.

2.1. Extreme learning machine (ELM)

The ELM [1] was originally proposed for the single-hidden layer feedforward neural networks (SLFNs) and was then extended to the generalized SLFNs where the hidden layer need not be neuron alike. In the ELM, the hidden layer need not be tuned. The overall description of ELM for generalized SLFNs is given by

$$f_L(\mathbf{x}) = \sum_{i=1}^{L} \boldsymbol{\beta}_i G(\mathbf{a}_i, b_i, \mathbf{x}), \quad \mathbf{x}, \mathbf{a}_i \in \mathbf{R}^n \tag{1}$$

where $\mathbf{a}_i$ and b_i are the learning parameters of hidden nodes and $\boldsymbol{\beta}_i$ the weight connecting the ith hidden node to the output node. $G(\mathbf{a}_i, b_i, \mathbf{x})$ is the output of the ith hidden node with respect to the input.

Huang et al. [2] have proved that SLFNs with a wide type of random computational hidden nodes have the universal approximation capability. For a given set of training examples $\{(\mathbf{x}_i,\mathbf{t}_i)\}_{i=1}^{N} \subset \mathbf{R}^n \times \mathbf{R}^m$, if the outputs of the network are equal to the targets, we have the compact formulation as follows:

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{T} \tag{2}$$

where

$$\mathbf{H}(\mathbf{a}_1,\ldots,\mathbf{a}_L,b_1,\ldots,b_L,\mathbf{x}_1,\ldots,\mathbf{x}_N) = \begin{bmatrix} G(\mathbf{a}_1,b_1,\mathbf{x}_1) & \cdots & G(\mathbf{a}_L,b_L,\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ G(\mathbf{a}_1,b_1,\mathbf{x}_N) & \cdots & G(\mathbf{a}_L,b_L,\mathbf{x}_N) \end{bmatrix}_{N\times L} \tag{3}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1^T \\ \vdots \\ \boldsymbol{\beta}_L^T \end{bmatrix}_{L\times m} \quad \text{and} \quad \mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_N^T \end{bmatrix}_{N\times m} \tag{4}$$

Here, $\mathbf{H}$ is called the hidden-layer output matrix of the network, whereby the ith column is the ith hidden node's output vector with respect to inputs $\mathbf{x}_1,\ldots,\mathbf{x}_N$ and the jth row is the output vector of the hidden layer with respect to input $\mathbf{x}_j$. $\boldsymbol{\beta}$ and $\mathbf{T}$ are corresponding matrices of output weights and targets, respectively.

After the hidden nodes are randomly generated and the training data are given, the hidden-layer output matrix $\mathbf{H}$ is known and need not be tuned. Thus, training SLFNs simply amounts to getting the solution of a linear system (5) of output weights $\boldsymbol{\beta}$. Under the constraint of minimum norm least square, i.e., $\min\|\boldsymbol{\beta}\|$ and $\min\|\mathbf{H}\boldsymbol{\beta}-\mathbf{T}\|$, a simple representation of the solution of the system (5) is given explicitly as follows:

$$\hat{\boldsymbol{\beta}} = \mathbf{H}^{\dagger}\mathbf{T}, \quad \mathbf{H}^{\dagger} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T \tag{5}$$

where $\mathbf{H}^{\dagger}$ is the Moore–Penrose generalized inverse of the hidden-layer output matrix $\mathbf{H}$. Huang et al. [11] have further shown that if the N training data are distinct, $\mathbf{H}$ is column full rank with probability one when $L \le N$. In real applications, note that the number of hidden nodes is always less than the number of training data, i.e., $L < N$.

It follows that these hidden-layer parameters of the ELM can be randomly generated instead of being iteratively tuned. One just needs to simply calculate the output weights using the least square method in one step.

2.2. Multi-response sparse regression (MRSR)

The multi-response sparse regression (MRSR) method [9] is proposed by Similä and Tikka for ranking the regressor candidates in multi-dimensional scaling. The main idea of the algorithm is as follows. Suppose that the targets are denoted by a matrix $\mathbf{T} = [\mathbf{t}_1,\ldots,\mathbf{t}_m]_{N\times m}$ and the regressors are denoted by a matrix $\mathbf{X} = [\mathbf{x}_1,\ldots,\mathbf{x}_L]_{N\times L}$. The MRSR algorithm adds sequentially one active regressor column to the model $\mathbf{Y}^k = \mathbf{X}\mathbf{W}^k$, where $\mathbf{Y} = [\mathbf{Y}_1,\ldots,\mathbf{Y}_m]_{N\times m}$ is the approximation to the target of the model, $\mathbf{W}^k$ is the weight matrix including k nonzero rows at the kth step of the MRSR algorithm. Thus, a new nonzero row and a new column of the regressor matrix are added to the model in each new step. More details of the MRSR algorithm could be referred to the original work [9].

Note that the MRSR algorithm is actually a variable ranking technique, rather than a selection method. The important characteristic is that the ranking obtained is exact if the problem is linear. Fortunately, if we consider the hidden node output matrix $\mathbf{H}$ of the SLFN as the regressors in the system to be modeled, the issue of structure identification for the ELM would be turned to the parameter-in-linear problem which coincides with the underlying idea of the MRSR. In this case, the MRSR would provide an exact ranking of the hidden nodes for the SLFN. Based on the exact hidden node ranking provided by the MRSR, randomly generated hidden nodes could be reasonably selected to construct the ELM with high performance and compact structure. It follows that the MRSR is a promising candidate for dealing with the model selection of the multi-output ELM.

2.3. Large tanker motion model

Mathematical models describing the manoeuvrability of large tankers in deep and confined waters could be found in [14]. The motion model considered in this paper presents the heading and propulsion dynamics of a large tanker described by the Bis-system which normalizes coefficients and derivatives by corresponding terms [14]. The resulting model can be represented by the following non-dimensional surge, sway and yaw equations:

$$\begin{cases} \dot{u} - vr = gX'' \\ \dot{v} + ur = gY'' \\ (Lk_z'')^2\dot{r} + Lx_G''ur = gLN'' \\ \dot{x} = u\ \cos(\psi) - v\ \sin(\psi) \\ \dot{y} = u\ \sin(\psi) + v\ \cos(\psi) \\ \dot{\psi} = r \end{cases} \tag{6}$$

where k_z'' is the non-dimensional radius of gyration of the ship in yaw, $x_G'' = L^{-1}x_G$, and X'', Y'' and N'' are nonlinear non-dimensional functions:

$$\begin{cases} gX'' &= X_{\dot{u}}''\dot{u} + L^{-1}X_{uu}''u^2 + L^{-1}X_{vv}''v^2 + L^{-1}X_{c|c|\delta\delta}''c|c|\delta^2 \\ & \quad + L^{-1}X_{c|c|\beta\delta}''c|c|\beta\delta + gT''(1-\hat{t}) + X_{\dot{u}\zeta}''\dot{u}\zeta + L^{-1}X_{uu\zeta}''u^2\zeta \\ & \quad + X_{ur\zeta}''ur\zeta + L^{-1}X_{vv\zeta\zeta}''v^2\zeta^2 \\ gY''\hat{t} &= Y_{\dot{v}}''\dot{v} + L^{-1}Y_{v|v|}''v|v| + L^{-1}Y_{ur}''ur + L^{-1}Y_{c|c|\delta}''c|c|\delta + Y_T''gT'' \\ & \quad + L^{-1}Y_{|c|c|\beta|\beta|\delta|}''|c|c|\beta|\beta|\delta| + Y_{ur\zeta}''ur\zeta + L^{-1}Y_{uv\zeta}''uv\zeta \\ & \quad + L^{-1}Y_{|v|v\zeta}''|v|v\zeta + L^{-1}Y_{|c|c|\beta|\beta|\delta|\zeta}''|c|c|\beta|\beta|\delta|\zeta + Y_{\dot{v}}''\dot{v} - Y_{\dot{v}\zeta}''\dot{v}\zeta \\ gLN'' &= L^2(N_{\dot{r}}''\dot{r} + N_{\dot{r}\zeta}''\dot{r}\zeta + N_{uv}''uv + LN_{|v|r}''|v|r + N_{|c|c\delta}''|c|c\delta \\ & \quad + LN_{ur}''ur + N_{|c|c|\beta|\beta|\delta|}''|c|c|\beta|\beta|\delta| + LN_{ur\zeta}''ur\zeta + N_{uv\zeta}''uv\zeta \\ & \quad + LN_{|v|r\zeta}''|v|r\zeta + N_{|c|c|\beta|\beta|\delta|\zeta}''|c|c|\beta|\beta|\delta|\zeta + LN_T''gT'' \end{cases} \tag{7}$$

where

$$\begin{cases} gT'' = L^{-1}T_{uu}''u^2 + T_{un}''un + LT_{|n|n}''|n|n \\ k_z'' = L^{-1}\sqrt{I_z''/m} \\ c^2 = c_{un}un + c_{nn}n^2 \\ \zeta = d/(h-d) \\ \beta = v/u \end{cases} \tag{8}$$

Here, u, v and x, y are the velocities and positions along X- (towards forward) and Y-axes (towards starboard), respectively, $r = \dot{\psi}$ is the yaw rate (where ψ is the yaw angle in the horizontal plane), L, d and m are the length, draft and mass of the ship, respectively, I_z'' is its mass moment of inertia about Z-axis (vertically downward with axis origin at free surface), $x_G'' = L^{-1}x_G$ is the non-dimensional X coordinate of ship's center of gravity (Y coordinate of ship's center of gravity y_G'' is taken as zero), g is acceleration due to gravity, X'', Y'' and N'' are the 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. All quantities in the above equations are non-dimensional by using Bis-system given in [14].

Substituting (7) and (8) into (6), we obtain the following nonlinear state space equation:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\mathbf{u}) \tag{9}$$

where $\mathbf{x}=[u,v,r,x,y,\psi]^{\mathrm{T}}$ is the state vector and $\mathbf{u}=[\delta,n,h]^{\mathrm{T}}$ is the input vector, and the nonlinear function vector $\mathbf{f}$ is comprised by

$$\left\{\begin{array}{lcl}
\dot{u} & = & \dfrac{1}{L(1-X''_{\dot{u}}-X''_{\dot{u}\zeta}\zeta)}((X''_{uu}+X''_{uu\zeta}\zeta)u^2+L(1+X''_{vr}+X''_{vr\zeta}\zeta)ur \\
 & & +(X''_{vv}+X''_{vv\zeta\zeta}\zeta^2)v^2+X''_{c|c|\delta\delta}c|c|\delta^2+X''_{c|c|\beta\delta}c|c|\beta\delta+LgT''(1-\hat{t})) \\
\dot{v} & = & \dfrac{1}{L(1-Y''_{\dot{v}}-X''_{\dot{v}\zeta}\zeta)}(Y''_{uv}uv+Y''_{|v|v}|v|v+Y''_{|c|c\delta}|c|c\delta+L(Y''_{ur}-1)ur \\
 & & +Y''_{|c|c|\beta|\beta|\delta|}|c|c|\beta|\beta|\delta|+LY''_{ur\zeta}ur\zeta+Y''_{uv\zeta}uv\zeta+Y''_{|v|v\zeta}|v|v\zeta \\
 & & +Y''_{|c|c|\beta|\beta|\delta|\zeta}|c|c|\beta|\beta|\delta|\zeta+LY''_{T}gT'') \\
\dot{r} & = & \dfrac{1}{L^2(k''_z2-N''_{\dot{r}}-N''_{\dot{r}\zeta}\zeta)}(N''_{uv}uv+N''_{|c|c\delta}|c|c\delta+L(N''_{ur}-x''_G)ur \\
 & & +N''_{|c|c|\beta|\beta|\delta|}|c|c|\beta|\beta|\delta|+LN''_{|v|r}|v|r+LN''_{ur\zeta}ur\zeta+N''_{uv\zeta}uv\zeta \\
 & & +LN''_{|v|r\zeta}|v|r\zeta+N''_{|c|c|\beta|\beta|\delta|\zeta}|c|c|\beta|\beta|\delta|\zeta+LN''_{T}gT'') \\
\dot{x} & = & u\ \cos(\psi)-v\ \sin(\psi) \\
\dot{y} & = & u\ \sin(\psi)+v\ \cos(\psi) \\
\dot{\psi} & = & r
\end{array}\right. \tag{10}$$

And, the resulting advance speed of the tanker could be given by

$$U=\sqrt{u^2+v^2} \tag{11}$$

3. Constructive multi-output extreme learning machine (CM-ELM)

In this section, a novel constructive method for multi-output ELM, termed as CM-ELM, will be presented in detail. Generally speaking, hidden node candidates are randomly generated via original ELM methodology in the initialization phase, and then an improved MRSR (I-MRSR) method is proposed to select potential regressor nodes chunk by chunk, rather than one by one, to accelerate the selection procedure. Finally, the validation and retraining process will be conducted on the selected subset of hidden nodes for realizing the CM-ELM with high performance and parsimonious structure.

It should be noted that the available observations for constructing the CM-ELM are randomly divided into three separate parts, i.e., training, validation and testing samples, which are not overlapped. Without loss of generality, the numbers of training, validation and testing samples, denoted as N, N_v and N_t, take 1/4, 1/4 and 1/2 of the whole available observations, respectively.

3.1. Hidden node candidate generation

The potential candidates for hidden nodes of the multi-output ELM (M-ELM) are randomly generated according to the original ELM strategy. Denote the number of candidates by L_{can} which generally satisfies that $L_{can}<N$, where N is the number of arbitrary distinct training samples $\{(\mathbf{x}_i,\mathbf{t}_i)\}_{i=1}^{N}\in\mathbf{R}^n\times\mathbf{R}^m$. Accordingly, the output of the M-ELM with full candidates (L_{can} hidden nodes) could be represented by

$$\mathbf{f}_{L_{can}}(\mathbf{x}_j)=\sum_{i=1}^{L_{can}}\boldsymbol{\beta}_i G(\mathbf{a}_i,b_i,\mathbf{x}_j),\quad \mathbf{x}_j,\mathbf{a}_i\in\mathbf{R}^n,\ \ \mathbf{f}_{L_{can}},\boldsymbol{\beta}_i\in\mathbf{R}^m \tag{12}$$

where $\mathbf{a}_i$ and b_i are the hidden node parameters, $\boldsymbol{\beta}_i$ is the output weight, and $G(\mathbf{a}_i,b_i,\mathbf{x}_j)$ denotes the output of the ith hidden node with respect to the input $\mathbf{x}_j, j=1,2,\ldots,N$. Given the randomly generated hidden node parameters $\mathbf{a}_i$ and b_i, the compact matrix description with respect to N data samples is as follows:

$$\mathbf{T}=\mathbf{H}_{L_{can}}\boldsymbol{\beta}+\mathbf{E} \tag{13}$$

where $\mathbf{T}=[\mathbf{t}_1,\ldots,\mathbf{t}_N]^T_{N\times m}$ the target matrix, $\mathbf{H}_{L_{can}}=[\mathbf{h}_1,\ldots,\mathbf{h}_{L_{can}}]_{N\times L_{can}}$ (here, $\mathbf{h}_i=[G(\mathbf{a}_i,b_i,\mathbf{x}_1),\ldots,G(\mathbf{a}_i,b_i,\mathbf{x}_N)]^T, i=1,\ldots,L_{can}$ denotes the ith hidden node's output vector with respect to inputs $\mathbf{x}_1,\ldots,\mathbf{x}_N$) the hidden node output matrix, $\boldsymbol{\beta}=[\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_{L_{can}}]^T_{L_{can}\times m}$ the weight matrix and $\mathbf{E}=[\mathbf{e}_1,\ldots,\mathbf{e}_N]^T_{N\times m}$ the error matrix.

It follows that given $\mathbf{h}_i$ randomly generated as regressor vector in the linear regression model (13), weight matrix $\boldsymbol{\beta}$ could be considered as the least-square solution derived from (14) and the corresponding residual error vectors $\mathbf{e}_1,\ldots,\mathbf{e}_N$ are assumed to be independent random variables with mean zero and unknown common variance:

$$\hat{\boldsymbol{\beta}}=\mathbf{H}^{\dagger}_{L_{can}}\mathbf{T},\quad \mathbf{H}^{\dagger}_{L_{can}}=(\mathbf{H}^T_{L_{can}}\mathbf{H}_{L_{can}})^{-1}\mathbf{H}^T_{L_{can}} \tag{14}$$

As a consequence, the candidate pool set of initial hidden nodes and corresponding optimal weight matrix in the least-square sense could be described as follows:

$$\mathcal{H}_{P_{can}}=\{(\mathbf{a}_i,b_i,\hat{\boldsymbol{\beta}}_i), i\in P_{can}\}_{ord},\quad P_{can}=\{1,2,\ldots,L_{can}\}_{ord} \tag{15}$$

where the set $\{\bullet\}_{ord}$ denotes the ordered set in which all the elements are ordered in the sequence with corresponding indices. Here, $\mathcal{H}_{P_{can}}$ is the ordered set of all the initial hidden nodes, and P_{can} denotes corresponding ordered set of indices.

3.2. Improved MRSR (I-MRSR)

In order to conveniently utilize the proposed I-MRSR method, the candidate node output matrix $\mathbf{H}_{L_{can}}$ and the target matrix $\mathbf{T}$ are normalized to standard normal distribution and zero mean, respectively, i.e.,

$$\mathbf{h}_i\leftarrow\frac{\mathbf{h}_i-\mathrm{mean}(\mathbf{h}_i)}{\mathrm{var}(\mathbf{h}_i)},\quad \mathbf{t}_j\leftarrow\mathbf{t}_j-\mathrm{mean}(\mathbf{t}_j),$$
$$i=1,2,\ldots,L_{can},\ \ j=1,2,\ldots,m \tag{16}$$

where $\mathrm{mean}(\cdot)$ and $\mathrm{var}(\cdot)$ denote mean value and variance of corresponding vectors, respectively.

Note that the hidden node candidate selection is simply to measure the significance of each candidate node output vector $\mathbf{h}_i$, and select the $\mathbf{h}_i$ with high significance from candidate pool P_{can}. Motivated by MRSR methodology, the selection of hidden nodes is regarded as a problem of subset model selection or regressor selection in linear regression. Intuitively, the hidden node output vectors could be ranked and selected chunk by chunk by a stepwise forward selection method based on a novel improved MRSR (I-MRSR) presented as follows.

Without loss of generality, the kth step is considered to unfold the main idea of the I-MRSR algorithm since the foregoing iterations add sequentially potential regressors selected from the candidate pool, and thereby resulting in the following model:

$$\mathbf{Y}^{(k)}=\mathbf{H}_{P(k)}\hat{\boldsymbol{\beta}}^{(k)} \tag{17}$$

where $P(k)$ is a set that consists of all the ordered indices of selected hidden node output vectors by the first k steps and $\mathbf{H}_{P(k)}$ contains hidden node output vectors selected from the candidate pool by the first k steps (i.e., $\mathbf{H}_{P(k)}=[\cdots\mathbf{h}_i\cdots]_{N\times|P(k)|}, i\in P(k)$).

At the beginning of the kth step, new possible hidden node output vectors are required to add to the matrix $\mathbf{H}_{P(k-1)}$. Accordingly, a cumulative correlation between the ith regressor $\mathbf{h}_i$ and the current residuals is explicitly defined as follows:

$$c_i^{(k)}=\|(\mathbf{T}-\mathbf{Y}^{(k)})^T\mathbf{h}_i\|_1=\sum_{j=1}^{m}|(\mathbf{t}_j-\mathbf{y}_j^{(k)})^T\mathbf{h}_i| \tag{18}$$

It is obvious that the larger the cumulative correlation $c_i^{(k)}$ is, the higher contribution of the candidate hidden node output vector (regressor) $\mathbf{h}_i$ would make to approximation error reduction. In this case, the candidates $\mathbf{h}_i\in P^c(k-1)=P_{can}-P(k-1)$ having larger cumulative correlation would have high priority to be recruited by the selection set P_k. We define the kth index subset of the candidates satisfying selection priority as follows:

$$P_k = \{i | c_i^{(k)} \ge \lambda c_{\max}^{(k)} + (1-\lambda) c_{\text{mean}}^{(k)}, i \in P^c(k-1)\}_{ord} \tag{19}$$

where

$$c_{\max}^{(k)} = \max_i \{c_i^{(k)}\}, \quad c_{\text{mean}}^{(k)} = \operatorname*{mean}_i \{c_i^{(k)}\} \tag{20}$$

Here, $c_{\max}^{(k)}$ and $c_{\text{mean}}^{(k)}$ are the maximum and mean values of the cumulative correlation for candidates considered in $P^c(k-1)$. $\lambda \in [0,1]$ denotes the weight parameter defining the dynamic scale of selection priority. It means that the candidates are selected chunk by chunk whereby the size is dependent on the dynamic distribution of cumulative correlation between candidate regressors and residuals. Accordingly, the chunk size would be adaptively scaled although the parameter λ is predefined. If set $P(0) = P_0 \equiv \varnothing$ denoting the empty initial selection, the selected hidden node output matrix $\mathbf{H}_{P(k)}$ and index set $P(k)$ could be updated as follows:

$$\mathbf{H}_{P(k)} = [\mathbf{H}_{P(k-1)}, \mathbf{H}_{P_k}], \quad \mathbf{H}_{P_k} = [\cdots \mathbf{h}_i \cdots]_{i \in P_k} \tag{21}$$

$$P(k) = \bigcup_{i=0}^{k} P_k, \quad P^c(k) = P_{can} - P(k) \tag{22}$$

With iterations of candidate selection, the matrix $\mathbf{H}_{P(k)}$ would be augmented by adding $\mathbf{H}_{P_k}$ with $|P_k|$ columns in the kth step. Accordingly, the computational burden for calculating the generalized inverse $\mathbf{H}_{P(k)}^{\dagger}$ of the regressor matrix $\mathbf{H}_{P(k)}$, according to (14), would inevitably become increasingly high with enlarging dimensions. On the contrary, the error minimized method [12] is used, in this paper, as the incremental updating mode for the estimate of the weight matrix $\hat{\boldsymbol{\beta}}^{(k+1)}$. In this case, the corresponding computational complexity would be dramatically reduced by the following recursive updating formulations:

$$\hat{\mathbf{Y}}^{(k+1)} = \mathbf{H}_{P(k)} \hat{\boldsymbol{\beta}}^{(k+1)} \tag{23}$$

$$\hat{\boldsymbol{\beta}}^{(k+1)} = \begin{bmatrix} \mathbf{U}^{(k)} \\ \mathbf{D}^{(k)} \end{bmatrix} \mathbf{T} \tag{24}$$

$$\mathbf{U}^{(k)} = \mathbf{H}_{P(k)}^{\dagger} (\mathbf{I} - \mathbf{H}_{P_k}^T \mathbf{D}^{(k)}) \tag{25}$$

$$\mathbf{D}^{(k)} = ((\mathbf{I} - \mathbf{H}_{P(k)} \mathbf{H}_{P(k)}^{\dagger}) \mathbf{H}_{P_k})^{\dagger} \tag{26}$$

Motivated by the less greedy algorithm in the MRSR [9], the largest step possible is taken in the direction of $\hat{\mathbf{Y}}^{(k+1)} - \mathbf{Y}^{(k)}$ until some $\mathbf{h}_i (i \in P^c(k))$ has as large cumulative correlation with the current residuals as the already added regressors. The I-MRSR estimate $\mathbf{Y}^{(k+1)}$ is updated by

$$\mathbf{Y}^{(k+1)} = \mathbf{Y}^{(k)} + \gamma^{(k)} (\hat{\mathbf{Y}}^{(k+1)} - \mathbf{Y}^{(k)}) \tag{27}$$

where $\gamma^{(k)}$ is the step size to find a new hidden node satisfying the maximum correlation in the direction of $\hat{\mathbf{Y}}^{(k+1)} - \mathbf{Y}^{(k)}$.

Note $\mathbf{H}_{P(k)}^T (\mathbf{T} - \mathbf{Y}^{(k)}) = \mathbf{H}_{P(k)}^T (\hat{\mathbf{Y}}^{(k+1)} - \mathbf{Y}^{(k)})$, the cumulative correlations in the next step become a function of γ,

$$\begin{cases} c_i^{(k+1)}(\gamma) = |1-\gamma| c_i^{(k)}, & i \in P(k) \\ c_i^{(k+1)}(\gamma) = \|\mathbf{a}_i^{(k)} - \gamma \mathbf{b}_i^{(k)}\|_1, & i \in P^c(k) \end{cases} \tag{28}$$

where $\mathbf{a}_i^{(k)} = (\mathbf{T} - \mathbf{Y}^{(k)}) \mathbf{h}_i$ and $\mathbf{b}_i^{(k)} = (\hat{\mathbf{Y}}^{(k+1)} - \mathbf{Y}^{(k)}) \mathbf{h}_i$. A new regressor with index $i \in P^c(k)$ will be recruited into the model when the following condition can be satisfied:

$$|1-\gamma| c_{\text{lower}}^{(k)} = \|\mathbf{a}_i^{(k)} - \gamma \mathbf{b}_i^{(k)}\|_1 \tag{29}$$

where

$$c_{\text{lower}}^{(k)} = \lambda c_{\max}^{(k)} + (1-\lambda) c_{\text{mean}}^{(k)} \tag{30}$$

The solution for γ could be found by

$$\gamma^{(k)} = \min\{\gamma | \gamma \in \Gamma_i \text{ and } \gamma \ge 0, i \in P^c(k)\} \tag{31}$$

where

$$\Gamma_i = \left\{ \frac{c_{\text{lower}}^{(k)} + \mathbf{s}^T \mathbf{a}_i^{(k)}}{c_{\text{lower}}^{(k)} + \mathbf{s}^T \mathbf{b}_i^{(k)}} \right\}_{\mathbf{s} \in S} \tag{32}$$

Here, S is the set of all 2^m sign vectors of size $m \times 1$, i.e., the elements of the vector $\mathbf{s}$ may be either 1 or -1.

3.3. Constructive model selection

Suppose that the final ranked candidate pool denoted by $\mathcal{H}_{P(L_{can})}$ is obtained by the I-MRSR method and described as follows:

$$\mathcal{H}_{P(L_{can})} = \{(\mathbf{a}_i, b_i, \hat{\boldsymbol{\beta}}_i), i \in P(L_{can})\}_{ord}, \ P(L_{can}) = \{i_1, i_2, \ldots, i_{L_{can}}\}_{ord} \tag{33}$$

In order to search for the optimal model selection for the CM-ELM by using validation data, motivated by the C_p criterion [21], a generalization measure (GM) is proposed for the CM-ELM candidate with $p(1 \le p \le L_{can})$ hidden nodes. The main idea for this generalization measure is that both residual sum of squares of prediction and hidden node numbers are considered to contribute to the minimum if the model selected obtains the best generalization which means best balance between accuracy and structure. To be specific, the generalization measure (GM) is defined as follows:

$$GM(p) = \left| \frac{\sum_{i=1}^{N_v} \|\mathbf{t}_i^{vld} - \mathbf{y}_i^{(p)}\|_2^2}{\left(\sum_{i=1}^{N} \|\mathbf{t}_i^{trn} - \mathbf{y}_i^{(L_{can})}\|_2^2\right)/N} - N_v + 2p \right| \tag{34}$$

where p is the number of hidden nodes in selected models, N_v and N are validation and initial training data numbers, $\mathbf{t}_i^{vld}, \mathbf{t}_i^{trn} \in \mathbf{R}^m$ are validation and initial training targets, $\mathbf{y}_i^{(p)}, \mathbf{y}_i^{(L_{can})} \in \mathbf{R}^m$ are actual outputs of corresponding models with p hidden nodes and L_{can} full selections, respectively.

With the hidden nodes being ranked by the descending significance via the foregoing I-MRSR method, the generalization capability of the model including the first p hidden nodes is iteratively measured by the above-mentioned $GM(p)$ criterion while the number p varies from 1 to L_{can}. It is obvious that the models with $p^* \in P^* \subset \{1, 2, \ldots, L_{can}\}$ hidden nodes would be considered as the most competitive ones if the number p^* satisfies the following condition:

$$\left| GM(p^*) - \min_{1 \le p \le L_{can}} \{GM(p)\} \right| \le \varepsilon \tag{35}$$

where $\varepsilon \ge 0$ is a user defined tolerance which could be intuitively chosen as follows:

$$\varepsilon = \eta \left(\max_{1 \le p \le L_{can}} \{GM(p)\} - \min_{1 \le p \le L_{can}} \{GM(p)\} \right), \quad 0 \le \eta \le 1 \tag{36}$$

Here, η is usually chosen to be far less than the unity and actually determines the optimal hidden node numbers p^* which contribute to the subset of hidden node numbers, i.e., $P^* = \{p_1^*, p_2^*, \ldots, p_C^*\}$, where C denotes the number of model selection candidates. Accordingly, resultant constructive model selections could be described as follows:

$$\mathcal{H}_c = \mathcal{H}_{P(p_c^*)} = \{(\mathbf{a}_i, b_i, \hat{\boldsymbol{\beta}}_i), i \in P(p_c^*)\}_{ord}, \ P(p_c^*) = \{i_1, \ldots, i_{p_c^*}\}_{ord} \tag{37}$$

where $c = 1, 2, \ldots, C$ denote indices of model selections.

3.4. Output weight update

The model selection phase is followed by the output weight update stage which aims to enhance the performance of the resulting CM-ELM model. To be specific, from the elite subset of model selection $\{\mathcal{H}_C\}$, the final CM-ELM model, denoted by $\mathcal{H}_{c^*}$, is randomly selected from the C candidates in the subset of model selections with superior generalization. In order to sufficiently extract informative knowledge from the whole training samples available, i.e., $\{(\mathbf{x}_i,\mathbf{t}_i)\}_{i=1}^{N}\bigcup\{(\mathbf{x}_i,\mathbf{t}_i)\}_{i=1}^{N_v}\in\mathbf{R}^n\times\mathbf{R}^m$ including initial training and validation data, the output weight of the final candidate model $\mathcal{H}_{c^*}$ is analytically determined by using pseudo-inverse technique based on the whole training data as follows:

$$\hat{\boldsymbol{\beta}}^{c^*}=\mathbb{H}^{\dagger}_{P(p^*_{c^*})}\mathbb{T},\quad \mathbb{H}^{\dagger}_{P(p^*_{c^*})}=(\mathbb{H}^{T}_{P(p^*_{c^*})}\mathbb{H}_{P(p^*_{c^*})})^{-1}\mathbb{H}^{T}_{P(p^*_{c^*})} \tag{38}$$

where subscript c^* denotes the c^*th CM-ELM model randomly selected from the elite subset of model selections, $\hat{\boldsymbol{\beta}}^{c^*}\in\mathbf{R}^{p^*_{c^*}\times m}$ denotes the output weight matrix estimate, $\mathbb{H}_{P(p^*_{c^*})}\in\mathbf{R}^{(N+N_v)\times p^*_{c^*}}$ and $\mathbb{T}\in\mathbf{R}^{(N+N_v)\times m}$ are the hidden node output matrix and target matrix with respect to $p^*_{c^*}$ original unnormalized hidden nodes and the whole training data, respectively, and $\mathbb{H}^{\dagger}_{P(p^*_{c^*})}$ is the corresponding generalized inverse. It follows that the resulting CM-ELM model with output weight updated can be given by

$$\mathcal{H}_{c^*}=\{(\mathbf{a}_i,b_i,\hat{\boldsymbol{\beta}}_i^{c^*}),i\in P(p^*_{c^*})\}_{ord},\;\; P(p^*_{c^*})=\{i_1,\ldots,i_{p^*_{c^*}}\}_{ord} \tag{39}$$

It should be noted that the final output weight update based on the whole training data would hopefully enhance the performance of approximation and generalization since additional information underlying in the validation data can be incorporated into the output weight, although the output weights have been incrementally updated by (23)–(26) based on initial training samples. Actually, performance evaluations in the section of simulation studies illustratively support the foregoing statement.

3.5. Complete algorithm of CM-ELM

Given initial training data $\{(\mathbf{x}_i,\mathbf{t}_i)\}_{i=1}^{N}\subset\mathbf{R}^n\times\mathbf{R}^m$, validation data $\{(\mathbf{x}_i,\mathbf{t}_i)\}_{i=1}^{N_v}$ and testing data $\{(\mathbf{x}_i,\mathbf{t}_i)\}_{i=1}^{N_t}$, the complete algorithm of the CM-ELM, shown in Fig. 1, is summarized in the phase-wise form as follows.

`Phase 1 Initial Training`: The original ELM algorithm is used to randomly generate L_{can} hidden nodes $\{(\mathbf{a}_i,b_i)\}_{i=1}^{L_{can}}$. The output matrix estimate $\hat{\boldsymbol{\beta}}$ is obtained by (14), and initial hidden node candidate pool $\mathcal{H}_{P_{can}}$ and corresponding ordered index set P_{can} are derived from (15).

`Phase 2 I-MRSR Based Hidden Node Ranking`: This phase can be divided into two steps as follows.

`Step 2.1`: According to (16), normalize the candidate node output matrix $\mathbf{H}_{L_{can}}$ and the target matrix $\mathbf{T}$.

`Step 2.2`: Based on the I-MRSR algorithm (18)–(32), do the candidate selection loop for ranking hidden nodes. The loop of this step is terminated till the L_{can} hidden nodes have been selected into $\mathcal{H}_{P_{L_{can}}}$ and $P_{L_{can}}$ given by (33).

`Phase 3 Constructive Model Selection`: Based on the ranked hidden node candidates (33), the generalization measure $GM(p)$ (34)–(36) is proposed to validate the generalization performance by selecting the first p hidden nodes from the ranked candidate pool (33) by using the validation data $\{(\mathbf{x}_i,\mathbf{t}_i)\}_{i=1}^{N_v}$. As a consequence, the resultant $C\ll L_{can}$ constructive model selections with $p_1^*,p_2^*,\ldots,p_C^*$ hidden nodes would be obtained by (37).

`Phase 4 Output Weight Update`: As the final phase of the CM-ELM algorithm, the resulting candidate model $\mathcal{H}_{c^*}$ is randomly selected from the elite subset of model selections $\{\mathcal{H}_C\}$ defined by (37). Initial training data together with validation samples are used to update the output weight of the resulting CM-ELM model via pseudo-inverse technique without any iterations given by (38). In this case, the performance of the final CM-ELM model (39) can be enhanced by incorporated the collective knowledge of validation data samples into the output weights.

4. Simulation studies

In this section, the performance evaluation and promising application of our proposed CM-ELM algorithm will be carried out. On the one hand, the effective performance and superiority will be evaluated by conducting simulation studies on multi-output regression benchmark datasets. On the other hand, the innovative application of the CM-ELM to large tanker motion dynamics identification will be presented by considering Esso Osaka tanker [14] as the reference model. All the simulations are carried out in the Matlab R2011b environment running on a PC with CPU 2.0 GHz and 2GB RAM. For all the regression applications, the input data have been normalized into $[-1, 1]$, while the output data have been normalized into [0,1]. The common parameter in the CM-ELM is simply chosen as follows: $\lambda=0.9$ and $\eta=0.1$. In order to obtain average results rather than the best one, 30 trials of simulations are conducted for each data case.

4.1. Performance evaluation on benchmark datasets

In order to validate the effectiveness and superiority of the CM-ELM method, simulation studies are conducted on multi-output regression problems which are listed in Table 1, whereby the training and testing samples are not overlapped and randomly selected from original datasets. Furthermore, comprehensive comparisons of our proposed CM-ELM using MRSR and I-MRSR, i.e., CM-ELM(MRSR) and CM-ELM(I-MRSR), with those counterparts without output weight update denoted by CM-ELM(MRSR)$_S$ and CM-ELM(I-MRSR)$_S$, as well as original ELM [1] and OP-ELM [8] are carefully presented. For fair comparisons, the sigmoidal additive activation function is chosen for all the above-mentioned algorithms, where the input weights and biases are randomly generated from the range $[-1, 1]$ and [0,1],

Table 1
Specification of benchmark multiple-output regressions.

Datasets	# Attributes	# Outputs	# Training data	# Testing data
Concrete slump	7	3	50	53
Abalone[a]	6	3	2000	2177
Wine (red)[a]	10	2	800	799
Wine (white)[a]	10	2	2400	2498

[a] The last 2 or 3 variables in original datasets are considered as outputs.

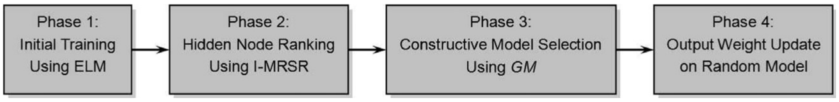


Fig. 1. The overall algorithm scheme of the CM-ELM.

Table 2
Performance comparisons on benchmark multiple-output regressions.

Datasets	Algorithms	Training RMSE		Testing RMSE		# Hidden nodes		
		Mean	Dev.	Mean	Dev.	$L_{\max}$	Mean	Dev.
Concrete slump	ELM	*0.0000*	*0.0000*	*9.4353*	*7.2264*	*50*	*50*	–
		0.1120	0.0078	0.2857	0.0426	25	25	–
		0.1461	0.0079	0.2462	0.0194	15	15	–
	OP-ELM	0.1893	0.0131	0.2572	0.0181	40	18.6667	7.0629
	CM-ELM(MRSR)$_S$	0.1275	0.0115	0.2564	0.0242	50	10.3667	0.8899
	CM-ELM(I-MRSR)$_S$	0.1359	0.0156	0.2491	0.0239	50	10.0667	1.7407
	CM-ELM(MRSR)	0.1485	0.0095	0.2313	0.0109	50	10.3667	0.8899
	CM-ELM(I-MRSR)	0.1516	0.0102	0.2311	0.0096	50	10.3000	1.4657
Abalone	ELM	0.0485	0.0001	0.0852	0.0177	100	100	–
		0.0520	0.0002	0.0546	0.0006	20	20	–
	OP-ELM	0.0559	0.0007	0.0569	0.0008	100	76.5000	10.5985
	CM-ELM(MRSR)$_S$	0.0522	0.0006	0.0564	0.0026	100	19.2000	3.7729
	CM-ELM(I-MRSR)$_S$	0.0523	0.0006	0.0561	0.0020	100	19.2000	4.6416
	CM-ELM(MRSR)	0.0517	0.0005	0.0554	0.0019	100	19.2000	3.7729
	CM-ELM(I-MRSR)	0.0517	0.0005	0.0551	0.0016	100	19.2000	4.0971
Wine (red)	ELM	0.0984	0.0007	0.1242	0.0071	100	100	–
		0.1145	0.0015	0.1156	0.0019	20	20	–
	OP-ELM	0.1225	0.0018	0.1207	0.0016	100	55.0000	10.0801
	CM-ELM(MRSR)$_S$	0.1138	0.0021	0.1194	0.0029	100	17.3333	3.4475
	CM-ELM(I-MRSR)$_S$	0.1146	0.0024	0.1185	0.0026	100	18.3000	4.0866
	CM-ELM(MRSR)	0.1142	0.0018	0.1145	0.0015	100	17.3333	3.4475
	CM-ELM(I-MRSR)	0.1143	0.0019	0.1144	0.0018	100	18.5333	4.2242
Wine (white)	ELM	0.0851	0.0003	0.1453	0.0449	200	200	–
		0.0923	0.0005	0.1045	0.0045	60	60	–
	OP-ELM	0.1039	0.0008	0.1073	0.0010	200	147.8333	13.1263
	CM-ELM(MRSR)$_S$	0.0924	0.0007	0.1036	0.0076	200	57.9667	6.0997
	CM-ELM(I-MRSR)$_S$	0.0929	0.0009	0.1046	0.0051	200	59.1667	7.6116
	CM-ELM(MRSR)	0.0914	0.0005	0.1013	0.0049	200	57.9667	6.0997
	CM-ELM(I-MRSR)	0.0917	0.0007	0.1044	0.0064	200	59.1000	7.9626

respectively. In addition, the numbers of candidate hidden nodes ($L_{\max}$) for the CM-ELM and OP-ELM are set equal except for the concrete slump dataset since the OP-ELM requires the number of initial full hidden nodes should be strictly less than that of data points. Considering the dilemma of the ELM between approximation and generalization, we implement two typical situations for each dataset whereby the numbers of hidden nodes are selected to be equal to $L_{\max}$ and approximately identical to that of the CM-ELM.

Simulation results for benchmark multi-output regression cases are presented in Table 2, where the hidden node numbers of the OP-ELM are the maximum of resulting individual models since the original OP-ELM works on single-output regressions, i.e., separated single-output models with different hidden nodes estimate individual outputs. It should be noted that the ELM using 25 hidden nodes is also included to balance the performance of approximation and generalization since the ELM using 50 hidden nodes can exactly approximate 50 distinct data points while the generalization validated by testing samples is extremely poor. For fairly intensive comparisons with the OP-ELM, the CM-ELMs without the final output weight update, i.e., CM-ELM (MRSR)$_S$ and CM-ELM(I-MRSR)$_S$, are also considered. In addition, comprehensive comparisons on training time of the above-mentioned methods are recorded simultaneously in Table 3. Generally comparing with the ELM and OP-ELM, the proposed CM-ELMs, including CM-ELM (MRSR)$_S$, CM-ELM(I-MRSR)$_S$, CM-ELM(MRSR) and CM-ELM(I-MRSR), tend to select the optimal subset of hidden nodes from the set of candidate hidden nodes since the CM-ELMs adopt the novelly defined generalization measure (34) which can achieve preferable generalization with parsimonious network structure. Due to the randomness of initial candidate hidden nodes in each simulation trial, the selection number of the optimal subset would be deviated around the mean values. It can be seen, from Tables 2 and 3, that the CM-ELM using MRSR is likely to obtain slightly compact model since the hidden nodes are ranked and selected in one-by-one manner which would dramatically increase computation burdens and thereby sacrificing learning speed. On the contrary, the CM-ELM using I-MRSR ranks hidden nodes in chunk-by-chunk mode which realizes not only comparable training time with the OP-ELM but also promising generalization performance. In addition, from Table 2, the final output weight update apparently enhances the performance of approximation and generalization with tiny increase of training time. In this case, among the foregoing variants of the CM-ELM, the one using I-MRSR and output weight update, i.e., CM-ELM(I-MRSR), realizes preferable performance in terms of generalization, parsimonious structure and comparative learning speed.

Table 3
Training time comparisons on benchmark multiple-output regressions.

Algorithms	Training time (s)			
	Concrete slump	Abalone	Wine (red)	Wine (white)
ELM	0.0026	0.0203	0.0088	0.0807
OP-ELM	0.0608	4.2775	0.9038	22.3901
CM-ELM(MRSR)$_S$	0.0738	15.3604	2.9042	64.0177
CM-ELM(I-MRSR)$_S$	0.0484	7.8027	1.4602	22.9187
CM-ELM(MRSR)	0.0754	15.5819	2.9557	64.0187
CM-ELM(I-MRSR)	0.0608	7.7428	1.4628	23.0234

Comparing with the ELM, from Table 2, the CM-ELM performs remarkable generalization with automatically created compact network. Especially, the CM-ELM(I-MRSR) achieves identical superiority to the ELM using full and comparable numbers of hidden nodes. Comparing with the OP-ELM, the significant characteristics of the CM-ELM are as follows. Owing to the chunk-by-chunk or one-by-one hidden node ranking and model selection, the CM-ELM can be able to constructively build the SLFN by employing the subset of hidden nodes with superior significance since the hidden nodes are well ranked in descent contribution. However, as a destructive approach, the OP-ELM starts with a large structure and then optimally prunes the insignificant hidden nodes one by one based on LOO or MRSR methods. In this case, the CM-ELM recruits significant hidden nodes from the candidate pool while the OP-ELM removes insignificant hidden nodes from the initial network. The situation occurs that moderate (slightly significant) hidden nodes cannot be removed from the redundant network. As a consequence, from Table 2, the SLFN trained by the CM-ELM is likely to be much more parsimonious than the one learned by the OP-ELM. Besides, from Table 2, it can be seen that the CM-ELM preserves reliably stable model scale with promising performance which indicates insensitivities to the randomness of hidden nodes in the initial candidate pool. However, the model structure from the OP-ELM varies critically with the initial candidate reservoir of random hidden nodes.

Note that the remained output weights in the OP-ELM are not updated after the optimal pruning. Approximation and generalization performance comparisons with the OP-ELM are fairly conducted on the CM-ELM without output weight update. From Table 2, the superiority of the CM-ELM to the OP-ELM is steadily demonstrated in terms of accuracy and network structure. As a consequence, the CM-ELM can be able to develop a reasonable balance between training and testing accuracy by using a parsimonious network, which results in much higher generalization capability. It is also evident, from Table 2, that the CM-ELM using output weight update can undoubtedly be able to obtain enhanced approximation and generalization accuracy by employing comparatively parsimonious network structure.

In light of training time shown in Table 3, the ELM runs very fast since no iterations and extra optimizations besides the pseudo-inverse computation have been conducted. Inevitably, as shown in Table 2, parsimonious network and convincing generalization of the ELM cannot be obtained simultaneously. On the contrary, as constructive and destructive methods, the CM-ELM and OP-ELM can be able to automatically select optimal subset of hidden nodes, whereby the approximation and generalization performance shown in Table 2 is convincingly guaranteed. However, the CM-ELM and OP-ELM would inevitably spend much more training time for model selections. It should be noted that the CM-ELM using I-MRSR in chunk-by-chunk mode can reasonably accelerate the learning speed which is comparable with that of the OP-ELM.

In order to visualize the approximation and generalization capabilities of CM-ELM, ELM and OP-ELM, the concrete slump regression is taken to exemplify the comparisons of training and testing RMSE over 30 evaluation trials, shown in Fig. 2. It is obvious, from Fig. 2(a), that the ELM using 15 hidden nodes and the CM-ELMs with approximate network scale achieve comparable training accuracy superior to that of the OP-ELM in that no extra output weight update on the remained network after the hidden node pruning is conducted. From Fig. 2(b), the CM-ELM(I-MRSR) possesses the preferable testing accuracy with small deviation over all the evaluations while the CM-ELM(MRSR) takes slightly higher prediction error. As the counterparts without output weight update, the CM-ELM(MRSR)$_S$ and CM-ELM(I-MRSR)$_S$ might perform degraded generalization capabilities which tend to be identical to those of the OP-ELM and ELM. Moreover, the prediction error deviations of the CM-ELM(MRSR) and CM-ELM(I-MRSR) among randomly generated candidate hidden nodes are also prone to be much smoother than those of CM-ELM(MRSR)$_S$, CM-ELM (I-MRSR)$_S$, ELM and OP-ELM. In addition, the hidden node selections of CM-ELM, ELM and OP-ELM versus evaluation trails are depicted in Fig. 3. Comparing with the OP-ELM, we can steadily found that the subsets of hidden nodes selected by the CM-ELMs are conducted in a remarkably stable process versus evaluation trials. It should be noted that the hidden node number deviations of the CM-ELMs using I-MRSR are slightly larger than those of the CM-ELMs using MRSR. Moreover, the hidden nodes selected by the CM-ELM are much parsimonious than the OP-ELM. It follows that the CM-ELM method achieves superior performance to the OP-ELM and ELM in terms of generalization and compact structure.

4.2. Application to large tanker motion dynamics identification

In this section, the effectiveness and superiority of the proposed CM-ELM based tanker motion dynamics identification will be demonstrated on typical maneuvering scenarios, i.e., sine rudder angle input and zigzag maneuvering, by using the benchmark model, ESSO OSAKA 190,000 dwt tanker [14]. The principal particulars of the tanker are as follows: length $L = 304.8$ m, breadth $B = 47.17$ m, draft $T = 18.46$ m, displacement $\nabla = 220,000\ \mathrm{m}^3$, block coefficient $C_B = 0.83$, design speed $U_0 = 16$ kn, nominal propeller speed $n = 80$ rpm, rudder rate limitation $\dot{\delta} = 2.33\ ^{\circ}/\mathrm{s}$. The values of hydrodynamic coefficients and other parameters can be referred to [14]. To be specific, simulation studies are presented in two phases, i.e., tanker motion dynamics identification using the CM-ELM algorithm and the CM-ELM based model validation. Moreover, comparisons with the ELM method are also comprehensively carried out.

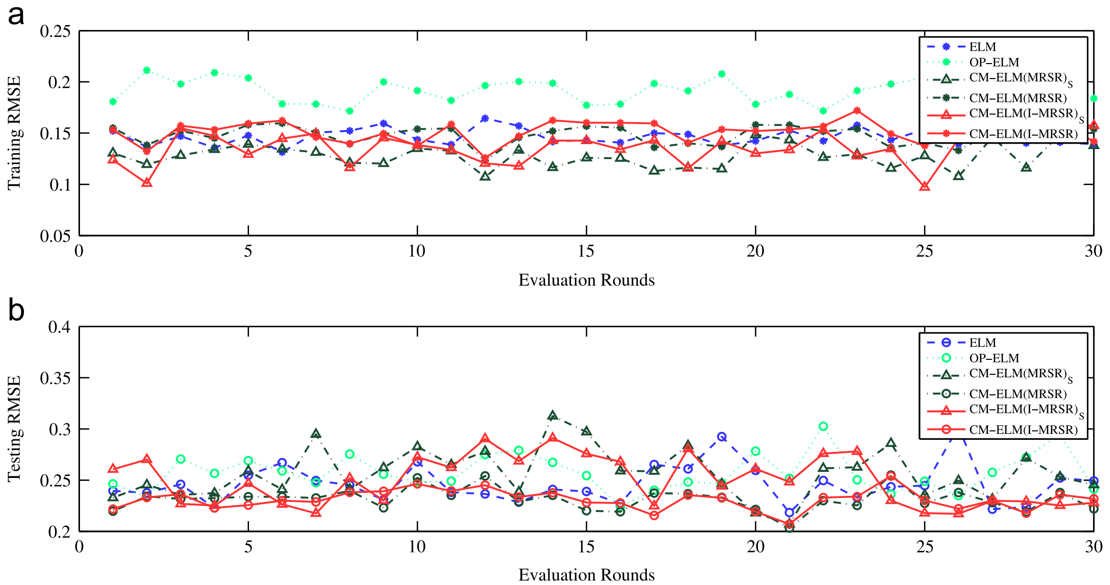


Fig. 2. Training and testing RMSE of CM-ELM, ELM and OP-ELM for concrete slump dataset. (a) Training RMSE vs. evaluation trials and (b) Testing RMSE vs. evaluation trials.

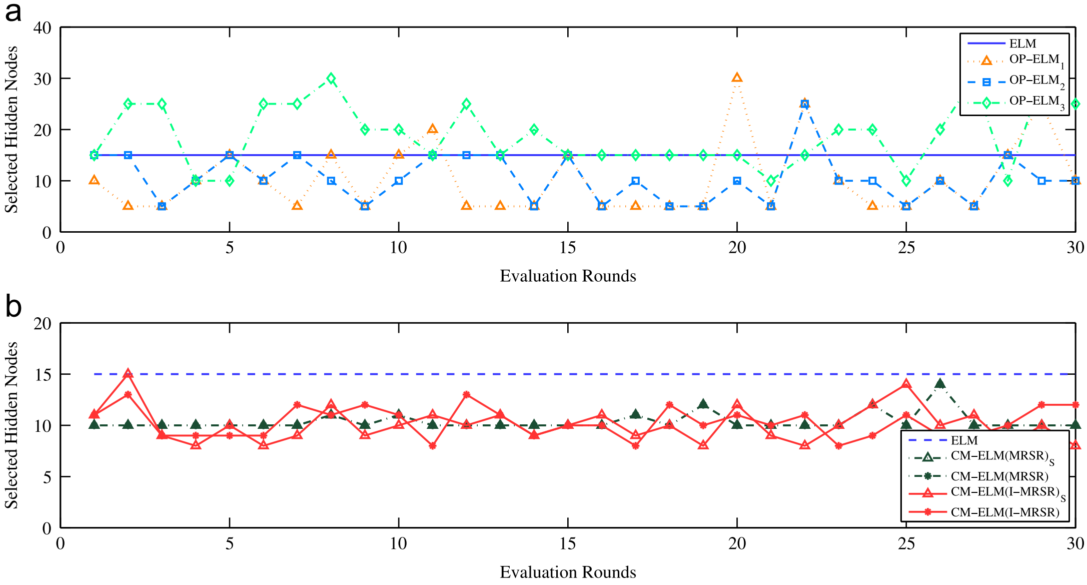


Fig. 3. Selected hidden neuron numbers of CM-ELM, ELM and OP-ELM for concrete slump dataset. (a) Selected hidden node numbers vs. evaluation trials and (b) Selected hidden node numbers vs. evaluation trials.

Table 4
Specification of tanker motion dynamics data.

Data type	Maneuvers			
	Sin/35°	*Sin*/20°	*Zig*/20°	*Zig*/35°
# Observations	6000	6000	6000	6000
# Training data	2000	0	0	0
# Testing data	4000	6000	6000	6000

The reference model (9) would be used to generate input–output data pairs described as $\{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1}^{N} \in \mathbf{R}^4 \times \mathbf{R}^3$, where $\mathbf{x}_i = [u_i, v_i, r_i, \delta_i]^T$, $\mathbf{t}_i = [\Delta u, \Delta v, \Delta r]^T$, and u, v, r, δ are surge, sway, yaw speed and rudder angle input, respectively. With loss of generality, the external input signal (rudder angle $\delta(k)$) is governed by the sine function under rudder rate limitation, i.e., $\delta(k) = A \sin(\omega k \Delta T)$, where A, ω, ΔT are amplitude, frequency and sampling parameter of rudder angle generation function, respectively. Here, we select $\omega = 0.1\ ^\circ/\mathrm{s}$ and $\Delta T = 1$ s. In addition, typical zigzag maneuvers with equal heading and rudder angle Z are also considered to generate testing data pairs for prediction evaluation of CM-ELM based tanker motion dynamics. For the simplicity, we denote the above-mentioned sine rudder angle input and zigzag maneuvers as *Sin*/*A* and *Zig*/*Z*, where *A* and *Z* are corresponding angle amplitudes (i.e., 20°, 35°, *etc.*). To be specific, specifications of all the generated data pairs for training and validation are listed in Table 4.

The first 2000 observations of *Sin*/35° maneuver are used to train the CM-ELM based tanker motion dynamics model, the rest of *Sin*/35° maneuver data pairs and all the observations of *Sin*/20°, *Zig*/20° and *Zig*/35° are fed to test the prediction and generalization performance of the resultant CM-ELM based model. The parameter $L_{\max}$ is chosen as 100. For intensive comparisons, the CM-ELM using MRSR and I-MRSR methods are all conducted to enhance the validation of our proposed algorithm. Simulation results for training and testing performance comparisons of *Sin*/35° maneuver with that of ELM are listed in Table 5, from which we find that the proposed CM-ELM based tanker motion models with compact structure perform remarkable superiority of approximation and generalization to the ELM based model using 30 hidden nodes. In light of training time, the CM-ELM(MRSR) and CM-ELM(I-MRSR) take much larger time to realize optimal model selections while the ELM performs dramatically fast learning speed at the cost of redundant network structure and inferior generalization.

As the visual illustration, the training and testing RMSE versus evaluation rounds are shown in Fig. 4. Comparing with the ELM, the CM-ELM(MRSR) and CM-ELM(I-MRSR) achieve preferable performance of approximation and generalization. It should be highlighted that the CM-ELM based models possess significant superiority to the ELM in terms of better generalization and parsimonious networks consisting of 32.7333 and 31.3333 hidden nodes in average. In addition, the hidden node selections for the CM-ELM using MRSR and I-MRSR during 30 training trials are shown in Fig. 5 which demonstrates that the subset of selected hidden nodes is comparably stable. It should be noted that the CM-ELM(I-MRSR) exhibits slightly larger deviation than the CM-ELM(MRSR) since chunks of hidden nodes are ranked and selected to contribute to the elite subset of model selections in the CM-ELM (I-MRSR). Moreover, the random selection of the final CM-ELM model from the resulting elite subset would also somewhat arise sensitivities to candidate hidden nodes. In addition, it should be emphasized that the CM-ELM(I-MRSR) can achieve much faster learning speed than the CM-ELM(MRSR).

In order to furthermore validate the prediction performance of ELM and CM-ELM based tanker dynamics models for other typical maneuvers, 6000 testing observations generated by *Sin*/20°, *Zig*/20° and *Zig*/35° maneuvers are used to evaluate the resulting CM-ELM based models including 20 finally selected hidden nodes compared with the ELM based model with 30 hidden nodes. The comparison results are listed in Table 6, from which we can see that for the same maneuver type, i.e., *Sin*/20°, both ELM and CM-ELM methods possess high prediction capabilities, where the CM-ELMs actually perform the superior performance the ELM. However, for different maneuvers, i.e., *Zig*/20° and *Zig*/35°, the CM-ELM method is significant superior to the ELM since the generalization of ELM is unfortunately lost to some extent. Nevertheless, the network structure complexity for the CM-ELM is merely one-fifth of initial candidate hidden nodes. For visually illustrations, Fig. 6 shows the prediction performance of *Sin*/20°, *Zig*/20° and *Zig*/35° maneuvers. It is clear that the CM-ELM based tanker models occupy uniform superiority to the ELM method.

In order to illustrate the convincing prediction performance of CM-ELM based model for complicated dynamics during tanker maneuvering, the desired and predicted multiple outputs of *Zig*/20° and *Zig*/35° maneuvers in 1000 sampling periods are separately depicted in Figs. 7–9, whereby all the desired outputs

Table 5
Training and testing performance comparisons on *Sin*/35° maneuver.

Algorithms	Training RMSE		Testing RMSE		# Hidden nodes			Training time (s)
	Mean	Dev.	Mean	Dev.	$L_{\max}$	Mean	Dev.	
ELM	0.0093	0.0013	0.0439	0.0054	30	30	–	0.0244
CM-ELM(MRSR)	0.0021	0.0014	0.0079	0.0038	100	32.7333	8.3622	13.9969
CM-ELM(I-MRSR)	0.0025	0.0014	0.0093	0.0051	100	31.3333	7.8798	7.1178

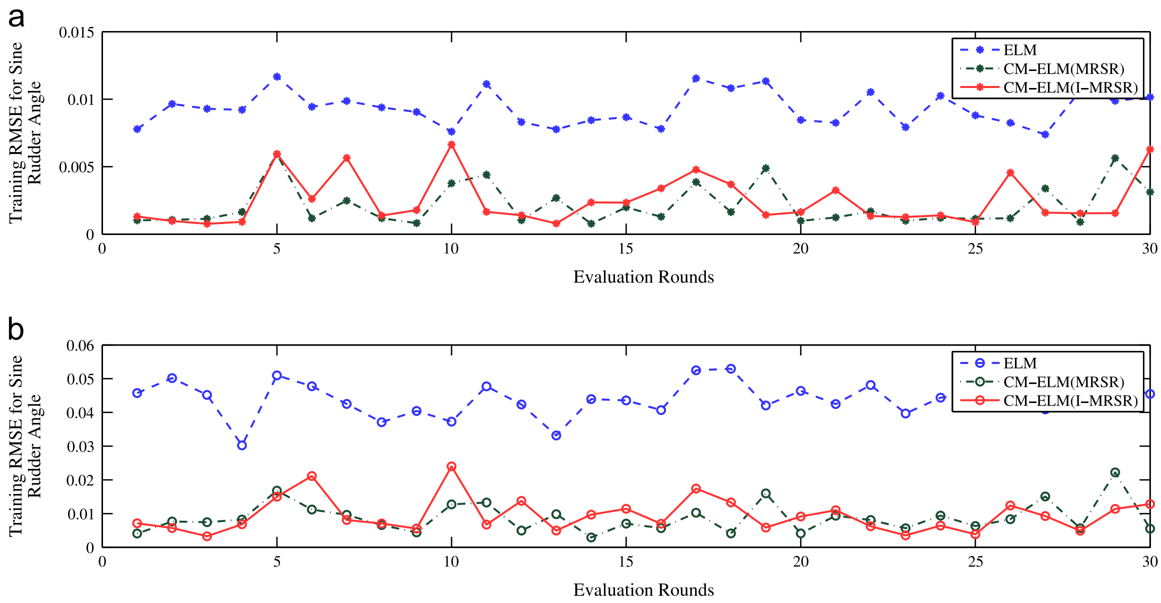


Fig. 4. Training and testing RMSE for *Sin*/35° maneuver. (a) Training RMSE for sine rudder angle (Sin/35°) and (b) Testing RMSE for sine rudder angle (Sin/35°)

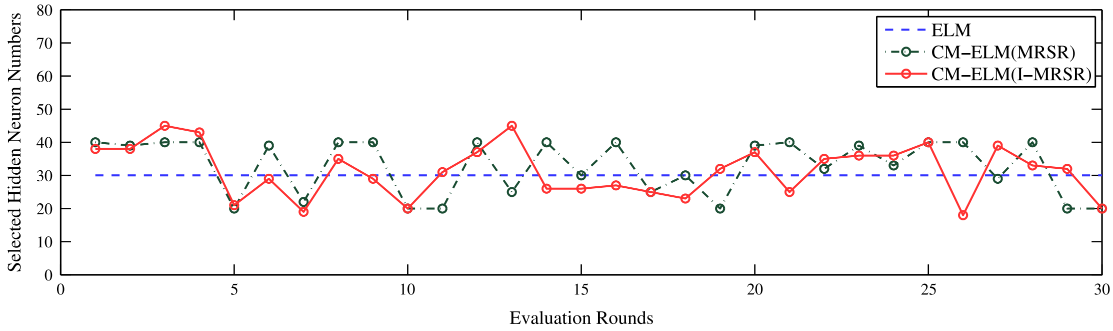


Fig. 5. Selected hidden node numbers for *Sin*/35° maneuver.

Table 6
Prediction performance comparisons on *Sin*/20°, *Zig*/20° and *Zig*/35° maneuvers.

Algorithms	Prediction RMSE						# Hidden nodes
	Sin/20°		*Zig*/20°		*Zig*/35°		
	Mean	Dev.	Mean	Dev.	Mean	Dev.	
ELM	0.0265	0.0027	0.0676	0.0052	0.2215	0.0179	30
CM-ELM(MRSR)	0.0070	0.0046	0.0080	0.0055	0.1866	0.0381	20
CM-ELM(I-MRSR)	0.0082	0.0059	0.0080	0.0036	0.1816	0.0264	20

have been normalized into the interval [0,1]. From Figs. 7 and 8, for the *Sin*/20° and *Zig*/20° maneuvers, both ELM and CM-ELM methods could well predict multiple outputs Δv, Δr except for the output Δu prediction of which the error of the ELM is seriously large. Unfortunately, for the large amplitude zigzag maneuver, i.e., *Zig*/35° shown in Fig. 9, the ELM method could hardly predict corresponding multiple outputs simultaneously. On the contrary, compared with the ELM, the CM-ELM based tanker models could still be able to predict comparably approving multiple outputs to some extent. It follows that the CM-ELM method is likely to

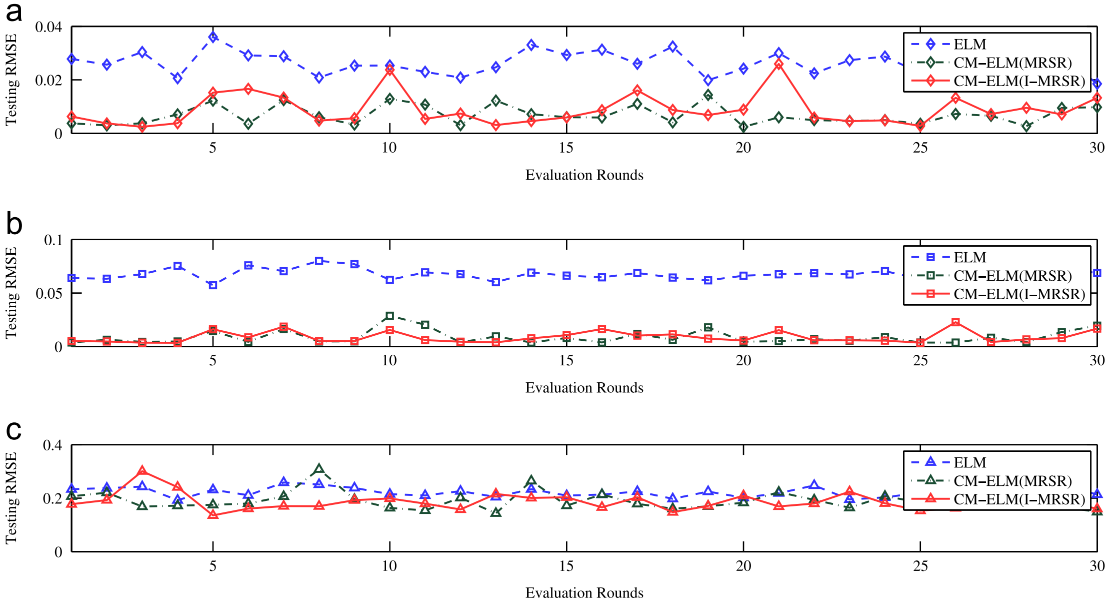


Fig. 6. Prediction performance of (a) *Sin*/20°, (b) *Zig*/20° and (c) *Zig*/35° maneuvers.

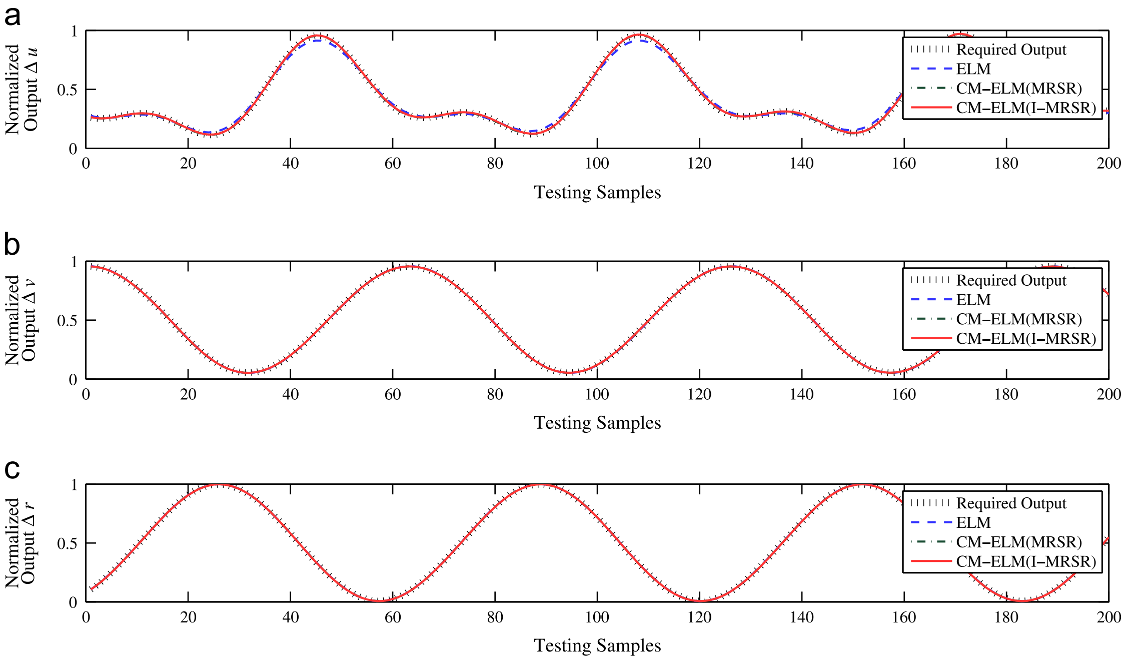


Fig. 7. Prediction performance for the *Sin*/20° maneuver.

possess high performance of approximation and generalization with compact network structure although the multiple-output mode ELM algorithm occupies comparable capabilities in some occasions. It should be noted that the CM-ELM using I-MRSR performs superior performance to the one using MRSR in terms of reliable generalization and accelerated learning speed.

5. Conclusions

In this paper, we propose a novel constructive multiple-output extreme learning machine (CM-ELM) method for multiple-output regression problems, and apply the CM-ELM algorithm to large tanker motion dynamics identification. To be specific, the CM-ELM method can be divided into four stages as follows. Initially, a set of L_{max} hidden nodes is randomly generated according to the ELM strategy as the candidate pool of our proposed algorithm. An improved multi-response sparse regression (I-MRSR) algorithm incorporating with λ weighting is presented to rank the hidden nodes chunk by chunk in candidate pool based on the cumulative correlation between hidden node output and current residual error. It is followed by the third stage that constructive selection of candidate hidden nodes is proposed based on the generalization measure (*GM*) determined by the C_p criterion. The elite subset of model selections including *C* models is determined by the *GM* criterion. As the final stage, output weight update using pseudo-inverse technique is conducted on the final CM-ELM model randomly selected from the elite subset of model selections. Furthermore, the promising application of the CM-ELM to large tanker motion dynamics identification is investigated by considering a group of well established nonlinear differential equations for tanker motion dynamics. Finally, the effectiveness and superiority of the CM-ELM is demonstrated by conducting simulation studies and

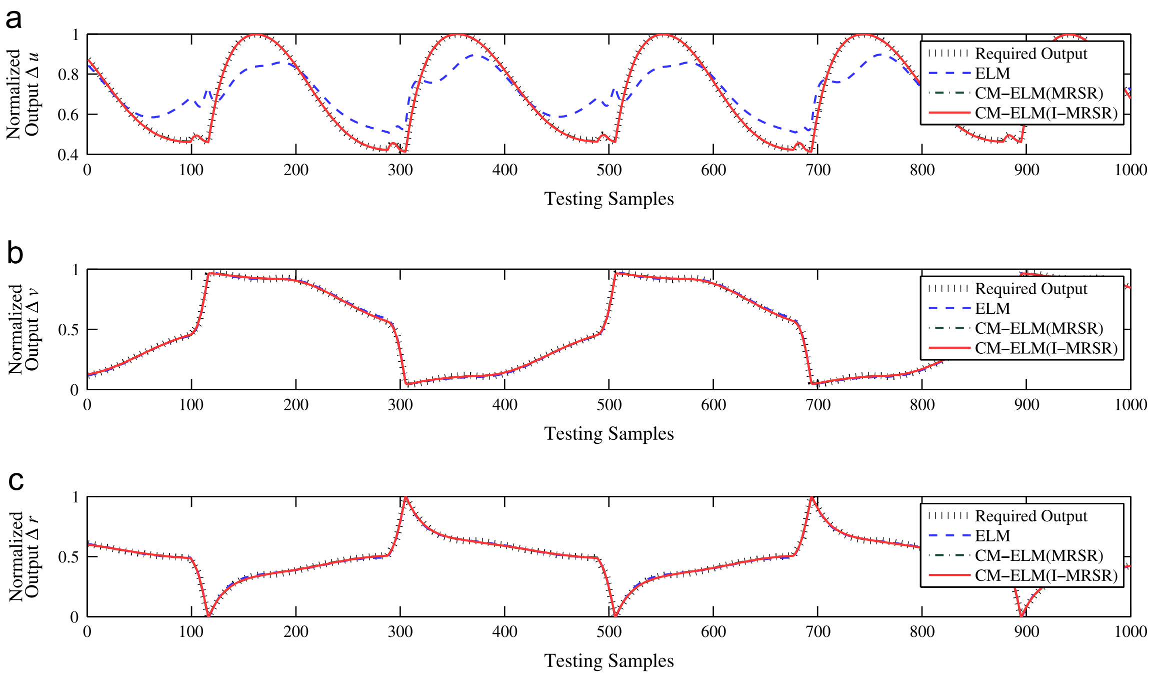


Fig. 8. Prediction performance for the *Zig*/20° maneuver.

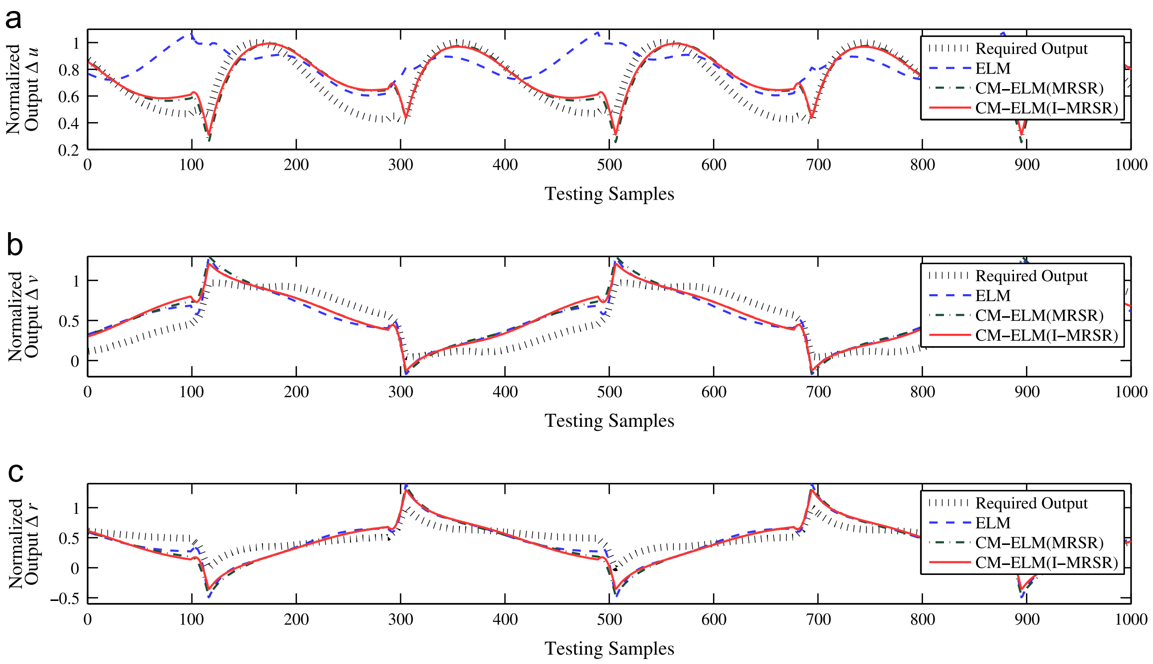


Fig. 9. Prediction performance for the *Zig*/35° maneuver.

comprehensive comparisons with ELM and OP-ELM methods on benchmark multiple-output regression datasets. The results indicate that the CM-ELM performs remarkable generalization with I-MRSR critically reducing the training time. In order to illustrate that the proposed tanker dynamics identification scheme based on the CM-ELM is effective and superior to the ELM based model, typical maneuvering scenarios, i.e., sine rudder angle input and zigzag maneuvers with moderate and extreme steering, are considered to evaluate the performance of approximation and prediction. Simulation results indicate that the CM-ELM based tanker motion model with compact structure preserves promising identification and generalization performance in terms of both moderate and extreme maneuvering.

Acknowledgments

The authors would like to thank anonymous referees for their invaluable comments and suggestions. This work is supported by the National Natural Science Foundation of PR China (under Grants

51009017, 51379002 and 61074096), Applied Basic Research Funds from Ministry of Transport of PR 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 PR China (under Grants 2009QN025, 2011JC002 and 3132013025).

References

[1] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (2006) 489–501.

[2] G.-B. Huang, L. Chen, C.-K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Trans. Neural Networks 17 (4) (2006) 879–892.

[3] G.-B. Huang, D.-H. Wang, Y. Lan, Extreme learning machines: a survey, Int. J. Mach. Learn. Cybern. 2 (2) (2011) 107–122.

[4] N. Wang, M.J. Er, X.-Y. Meng, A fast and accurate online self-organizing scheme for parsimonious fuzzy neural networks, Neurocomputing 72 (16) (2009) 3818–3829.

[5] N. Wang, M.J. Er, X.-Y. Meng, X. Li, An online self-organizing scheme for parsimonious and accurate fuzzy neural networks, Int. J. Neural Syst. 20 (5) (2010) 389–405.

[6] N. Wang, M.J. Er, X.-Y. Meng, X. Li, A generalized ellipsoidal basis function based online self-constructing fuzzy neural network, Neural Process. Lett. 34 (1) (2011) 13–37.

[7] H.-J. Rong, Y.-S. Ong, A.-H. Tan, Z. Zhu, A fast pruned-extreme learning machine for classification problem, Neurocomputing 72 (1) (2008) 359–366.

[8] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, A. Lendasse, Op-elm: optimally pruned extreme learning machine, IEEE Trans. Neural Networks 21 (1) (2010) 158–162.

[9] T. Similä, J. Tikka, Multiresponse sparse regression with application to multidimensional scaling, in: Proceedings of the 15th International Conference on Artificial Neural Networks (ICANN 2005), vol. 3697, Warsaw, Poland, September 11–15, 2005.

[10] G.-B. Huang, L. Chen, Enhanced random search based incremental extreme learning machine, Neurocomputing 71 (2008) 3460–3468.

[11] G.-B. Huang, L. Chen, Convex incremental extreme learning machine, Neurocomputing 70 (2007) 3056–3062.

[12] G. Feng, G.-B. Huang, Q. Lin, R. Gay, Error minimized extreme learning machine with growth of hidden nodes and incremental learning, IEEE Trans. Neural Networks 20 (8) (2009) 1352–1357.

[13] Y. Lan, Y.C. Soh, Guang-Bin Huang, Constructive hidden nodes selection of extreme learning machine for regression, Neurocomputing 73 (2010) 3191–3199.

[14] T.I. Fossen, Marine control systems: guidance, navigation and control of ships, rigs and underwater vehicles, in: Marine Cybernetics AS, Trondheim, Norway, 2002.

[15] S. Sutulo, L. Moreira, C.G. Soares, Mathematical models of ship path prediction in manoeuvring simulation systems, Ocean Eng. 29 (1) (2002) 1–19.

[16] H.-K. Yoon, K.-P. Rhee, Identification of hydrodynamic coefficients in ship maneuvering equations of motion by estimation-before-modeling technique, Ocean Eng. 30 (18) (2002) 2379–2404.

[17] X.-G. Zhang, Z.-J. Zou, Identification of Abkowitz model for ship maneuvering motion using ε-support vector regression, J. Hydrodynamics 23 (3) (2011) 353–360.

[18] G. Rajesh, S. Bhattacharyya, System identification for nonlinear maneuvering of large tankers using artificial neural network, Appl. Ocean Res. 30 (4) (2008) 256–263.

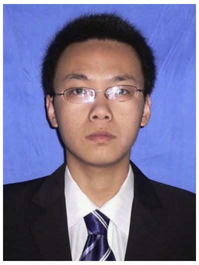
[19] A.B. Mahfouz, M.R. Haddara, Effects of the damping and excitation on the identification of the hydrodynamic parameters for an underwater robotic vehicle, Ocean Eng. 30 (8) (2003) 1005–1025.

[20] L. Moreira, C.G. Soares, Dynamic model of manoeuvrability using recursive neural networks, Ocean Eng. 30 (13) (2003) 1669–1697.

[21] C.L. Mallows, Some comments on C_p, Technometrics 42 (1) (2000) 87–94.

Ning Wang received his B.Eng. degree in Marine Engineering and the Ph.D. degree in Control Theory and Engineering from the Dalian Maritime University (DMU), Dalian, China, in 2004 and 2009, respectively. From September 2008 to September 2009, he was financially supported by China Scholarship Council (CSC) to work as a joint-training Ph.D. student at the Nanyang Technological University (NTU), Singapore. He is currently an Associate Professor with the Marine Engineering College, Dalian Maritime University (DMU), Dalian 116026, China. His research interests include artificial neural networks, fuzzy systems, machine learning, ship intelligent control, and dynamic ship navigational safety assessment. Dr. Wang also received the DMU Excellent Doctoral Dissertation and the DMU Outstanding Ph.D. Student Award in 2010.

Min Han received the B.S. and M.S. degrees from the Department of Electrical Engineering, Dalian University of Technology, Dalian, China, and the M.S. and Ph.D. degrees from Kyushu University, Fukuoka, Japan, in 1982, 1993, 1996, and 1999, respectively. She is a Professor with the Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology. Her current research interests include neural networks and chaos and their applications to control and identification.



Nuo Dong received the B.S. degree from the Department of Electrical Engineering and Automation in Marine Engineering College, Dalian Maritime University, Dalian 116026, China, in July 2012. He is currently pursuing his M.S. degree at the same university. His current research interests include unmanned crafts and their intelligent modeling and control.

Meng Joo Er is currently a Full Professor in Electrical and Electronic Engineering, Nanyang Technological University, Singapore, and a Chair Professor of the Dalian Marinetime University, PRC. He served as the Founding Director of Renaissance Engineering Programme, College of Engineering (CoE) and an elected member of the NTU Advisory Board from 2009 to 2012. He also served as a member of the NTU Senate Steering Committee from 2010 to 2012 and the Institution of Engineers, Singapore (IES) Council from 2008 to 2012.

He has authored five books entitled Dynamic Fuzzy Neural Networks: Architectures, Algorithms and Applications and Engineering Mathematics with Real-World Applications published by McGraw Hill in 2003 and 2005, and Theory and Novel Applications of Machine Learning published by In-Tech in 2009, New Trends in Technology: Control, Management, Computational Intelligence and Network Systems and New Trends in Technology: Devices, Computer, Communication and Industrial Systems, both published by SCIYO, 16 book chapters and more than 400 refereed journal and conference papers in his research areas of interest.

In recognition of the significant and impactful contributions to Singapore's development by the research project entitled Development of Intelligent Techniques for Modeling, Controlling and Optimising Complex Manufacturing Systems, Professor Er won the Institution of Engineers, Singapore (IES) Prestigious Engineering Achievement Award 2011. Under his leadership, the NTU Team emerged first runner-up in the Freescale Technology Forum Design Challenge 2008. The NTU is the only Asian Team among the top three positions at the first Freescales first green engineering design contest, reaffirming NTUs strength in design, creativity and innovation. He is also the only dual winner in Singapore IES Prestigious Publication Award in Application (1996) and IES Prestigious Publication Award in Theory (2001). He received the Teacher of the Year Award for the School of EEE in 1999, School of EEE Year 2 Teaching Excellence Award in 2008 and the Most Zealous Professor of the Year Award 2009. He also received the Best Session Presentation Award at the World Congress on Computational Intelligence in 2006. On top of this, he has more than 40 awards at international and local competitions.

Currently, Professor Er serves as the Editor-in-Chief of the International Journal of Electrical and Electronic Engineering and Telecommunications, an Area Editor of International Journal of Intelligent Systems Science and an Associate Editor of 11 refereed international journals, namely International Journal of Fuzzy Systems, Neurocomputing, International Journal of Humanoid Robots, Journal of Robotics, International Journal of Mathematical Control Science and Applications, International Journal of Applied Computational Intelligence and Soft Computing, International Journal of Fuzzy and Uncertain Systems, International Journal of Automation and Smart Technology, International Journal of Modeling, Simulation and Scientific Computing, International Journal of Intelligent Information Processing and the Open Electrical and Electronic Engineering Journal. Furthermore, he served as an Associate Editor of IEEE Transactions on Fuzzy Systems from 2006 to 2011 and a Guest Editor of International Journal of Neural Systems.

Professor Er has been invited to deliver more than 60 keynote speeches and invited talks overseas. He has also been active in professional bodies. Currently, he is the Vice-Chairman of IEEE Computational Intelligence Society (CIS) Standards Committee, Chairman of IEEE Computational Intelligence Society Singapore

Chapter and Chairman of IES Electrical and Electronic Engineering Technical Committee. Under his leadership, the IEEE CIS Singapore Chapter won the CIS Outstanding Chapter Award 2012. The Singapore Chapter is the first chapter in Asia to win the award since it was inaugurated in 2006. In recognition of his outstanding contribution to IES, he was awarded the IES Silver Medal in 2011.

Professor Er Meng Joo's areas of expertise are intelligent control theory and applications, fuzzy logic and neural networks and robotics and automation. His current works focus on control theory and applications, fuzzy logic and neural networks, computational intelligence, cognitive systems, robotics and automation, sensor networks and biomedical engineering.