Journal of Naval Architecture and Marine Engineering
December, 2011
DOI: 10.3329/jname.v8i2.6945 http://www.banglajol.info

A FLEXIBLE SYSTEM FOR INITIAL SHIP DESIGN PARAMETERS ESTIMATION USING A SYSTEM OF NEURAL NETWORKS

Hamada Senousy[1] and Mahmoud Abou-El-Makarem[2]
Marine Engineering Department, Faculty of Engineering, Alexandria University, Egypt
[1]Email: snosy_ship_2010@yahoo.com [2]Email: apoelmakarem@yahoo.org

Abstract:

To initialize ship design process, it is very important to be able to develop an initial estimate of ship parameters to satisfy designer required specifications. For new emerging designs, this estimate has to be made based on a limited available set of examples. Moreover, a practical estimate prediction strategy should be flexible enough having no distinction between input (specified constraints) and outputs (parameters required to be estimated), since these vary from one design case to another. Conventional regression-based techniques, which are usually employed to provide the required estimates, suffer from low accuracy in case of a small number of available examples. In addition to that, they fail to capture the interrelation between different design parameters. To overcome these limitations and others, the present paper proposes a new approach based on a system of artificial neural-networks (ANNs). The new approach not only overcomes regression limitations but is also capable of providing a reliable estimate of initial design offset table based on different ANN outputs. The paper uses a case study for demonstrating the merits of the proposed approach.

Keywords: Ship design, regression, ship series, Artificial Neural Networks (ANNs), Multilayer Perceptrons (MLPs), Normalized Gaussian Modified Lagrangian (NGML)

1. Introduction

Prediction based design remains to be an important initial step in the design process of complex systems. This is especially exemplified in the complex process of ship design. Prediction-based design uses a set of user specifications for a certain class of systems (ships) to predict the rest of the design parameters. This prediction depends on available data of existing designs of the same class. This prediction problem is usually performed using conventional regression techniques. Despite the advances in Ship design software, prediction based design remains to be indispensable. All design softwares proceed by evaluating a design input by the user (calculates resistance, assess fatigue, estimate needed power, weight, stability). Thus, it remains the role of prediction based design to provide a good near-optimal initial design point, which can be verified and further optimized using available software tools. Ideally the procedure used to predict a suitable initial design point should exhibit the following desirable features:

1. The procedure should be able to respond to queries with varying inputs and of varying length. This is because the inputs to the prediction procedure vary from one case study to another dependent on the nature of the area of application. To clarify the importance of this point in particular, consider the following arguments. Any designer when designing a new ship must have input data from the owner, which are considered as design constraints. He/She must satisfy these constraints, while keeping the design as optimal as possible (large dead weight with small dimension and low power). For example, an owner may specify a ship with a certain dead weight and certain draft (design constraints). These parameters are very important parameters for a ship that will pass through Suez Canal. The Suez Canal has a certain draft that ships must not exceed to avoid additional resistance and grounding. In another example, however, an owner may specify a ship with certain dead weight and certain beam. These are the most important parameters in case the designed ship will pass through Panama canal. Panama canal has restricted breadth. Thus, ships must have a certain beam that does not exceed the Panama breadth. It is clear from these examples that having a design strategy with fixed predetermined inputs and outputs is highly undesirable and unpractical, since it will not be possible to employ it in different design situations.
2. The prediction procedure should take into account the inter-relation between the different design parameters. Series-based design clearly lacks this advantage, since the ship lines are extracted based on a hierarchical design procedure that take into account the ratios between the different design parameters

 Received on: January 2011

rather than their actual values (for example, the length to breadth ratio, the breadth to draft ratio). This is natural to prediction procedures that rely on single-variable regression.

3. The procedure should be able to make use of examples with partial available information and should be able to provide accurate estimates based on information from a limited number of design examples. This is common with new emerging designs. With such designs there are usually very limited design examples with detailed information made public.
4. The prediction procedure decisions should be transparent to the user i.e. the reasoning performed on available data (example) to produce the required estimates should be clear to the user. Moreover, the estimates should be accompanied by a degree of confidence that gives the user an idea of how confident the procedure is in a particular estimate.
5. Whenever, more examples become available, there should be an easy way of incorporating information from them within the prediction procedure with out having to re-build the prediction system from scratch.

Unfortunately, existing methods require thousands of entries to build the database. In addition to that traditional regression variants used to produce estimates for design variables lack most of the advantages stated above. Although advanced methods such as artificial neural networks (ANNs) and Bayesian networks successfully capture some of these advantages, they fail to satisfy them all. Thus, the authors present an efficient accurate alternative strategy for providing initial ship design estimates. The strategy uses a multi-ANN-based approach. The approach can be considered an adapted extension of the computational strategies proposed in literature (Nelwamondo et al. 2007), (Polikar et al. 2001) so as to handle the requirements of ship design discipline. The proposed strategy makes it possible to construct reliable initial ship design parameters estimates from only a few data points (ship data). This is due to the generalization property of ANNs. Fuzzy logic interpretation of neural networks decisions may be used in future research to guarantee the transparency of the system decisions to designers (Hamid et al. 2008). Using a multi-ANN approach helps us to cope with three problems. The first is handling incomplete entries which may meet designers during series building or when employing them in` design. The second is that they allow for incremental learning which improves the quality of prediction by adding new data entries (ship examples) if they become available without having to re-train the series ANNs as a whole. Furthermore, multiple ANNs can be used to eliminate the input output distinction and lets decision making rely on available evidence.

It is important to note that the use of neural networks in preliminary ship design itself is not new. Several authors have pointed out the importance of using ANNs in place of traditional regression techniques (Gougoulidis, 2008), (Bertram, 2004). ANNs have been used in various aspects of ship design and stability. Some developments are summarized below. For preliminary ship design, Clausen et al. (2001) have developed multilayer perceptrons (MLP) and Bayesian networks for the determination of the main particulars of ships at the initial design stage. A single hidden layer MLP network has been developed with three neurons in the hidden layer. The loading capacity of the vessel is the input to the network which estimates six parameters, namely, length, breadth, speed, draft, depth and displacement. Alkan et al. (2004) propose two ANNs for determining initial stability particulars of fishing vessels. The architecture is MLP with two hidden layers. Seven neurons in the first hidden layer and six in the second hidden layer have been used. Inputs to the first layer are the block coefficient, beam, depth and length to displacement ratio. The output is the vertical centre of gravity. In their second network, the inputs are the length overall, moulded beam, design draught, moulded depth, block coefficient, prismatic coefficient, water line area coefficient and displacement at the design waterline. Since it is important to estimate the metallic hull weight in primary design of the ship in order to control the weight and cost of the ships built,. Wu et al. (1999) have developed an MLP network for this purpose. They have used 10 neurons in the hidden layer. Inputs to the network are: length between perpendiculars (L), depth (D), draught (d), breadth (B), block coefficient (Cb), L/D, B/D, and d/D. Output is the metallic hull weight. Islam et al. (2001) have used ANNs for automatic hull form generation. They have used a three layer MLP network. The network has four inputs: length, breadth, draft and type of ship. Three hidden parameters are the water plane area, sectional area and midship area and the four outputs are the displacement, breadth, draft and speed. For the purpose of hull optimization, several authors (Schmitz,2004),(Abramowski,2010) used ANNs as response surfaces i.e. an ANN is trained to take the design particulars as input and predict its performance (value of the objective function of the optimization procedure) as output. This speeds up the optimization procedure.

However, all these efforts merely concentrated on using ANNs to predict a certain parameter based on some design constraints. They were not flexible enough to simultaneously handle the variation of design constraints from application to another. Moreover, they rely mainly on using ANNs trained with backpropagation algorithm. The backpropagation algorithm has several disadvantages. Training is slow and there is a large probability of getting trapped in a local minimum. Moreover, although the work of Hansen (2000) offers some flexibility in parameters values specification due to the use of Bayesian networks in conjunction with ANNs, it still requires a huge training database. In this work, however, an alternative ANN structure based on the work of (Abdelsalam 2009) called normalized gaussian modified lagrangian (NGML) and use a system of ANNs and not just one to add flexibility to our system and make it suitable for different application scenarios and also allow users to easily make use of all available data in training the system (even partial information can be used to train some of the ANNs). Also using NGML ANNs makes it easy to include additional examples whenever they are available. Unlike multilayer perceptrons (MLPs) trained with the conventional iterative backpropagation algorithm, NGML ANNs (as explained in section II) are one-shot trained ANNs. Thus, the addition of new examples does not involve lengthy retraining procedure. In addition to that NGML ANNs do not require many examples to give satisfactory results.

2. Mathematical Formulation

2.1 A brief overview of Normalized Gaussian Modified Lagrangian (NGML) Artificial Neural Networks (ANNs)

Conventional ANNs often suffer from local minima trapping and require long training and trial and error parameter-tuning (Haykin,2008). To avoid all this, Abdelsalam et.al. (Abdelsalam,2009) proposed a new one-shot trained ANN called Gaussian Modified Lagrangian (GML) ANN. The architecture of the proposed ANN is shown in Fig.1.

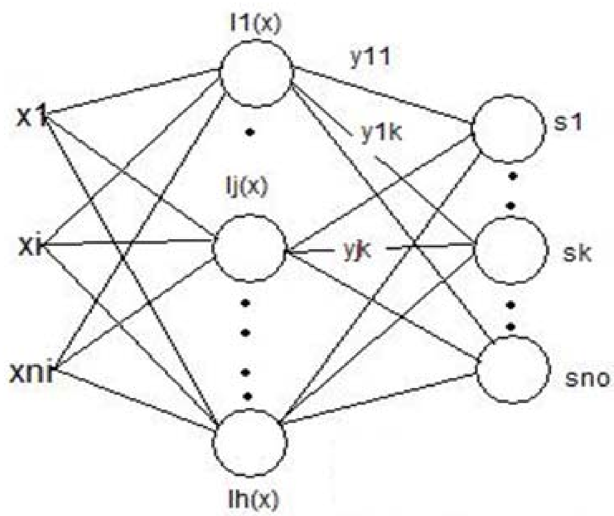

Fig.1: Architecture of the feedforward NGML ANN used in this research.

The shown architecture is for a general multi-input multi-output curve fitting problem. There are *h* hidden neurons. With GMLs the number of hidden neurons is equal to the number of training examples. The hidden neurons activation function is given by:

$$l_j(x) = \frac{\prod_{\substack{jj \\ jj \neq j}} \left(1 - g_{jj}(x)\right)}{\prod_{\substack{jj \\ jj \neq j}} \left(1 - g_{jj}(x_j)\right)} \tag{1}$$

Where $jj = 1,2,\ldots\ldots\ldots h,\ j = 1,2,\ldots.,h$

$$g_{jj}(x) = \exp\left(-\left\|x - x_{jj}^{p}\right\|^2\right) \tag{2}$$

While the activation function of the output neurons is given by:

$$s_k = \frac{\sum_{j=1}^{h} l_j(x)\, y_{jk}(x)}{\sum_{j=1}^{h} l_j(x)} \tag{3}$$

Where k=1,2,………N_o

N_o is the number of output neurons

The normalizing denominator in s_k is optional. The performance of NGML ANNs is further improved by replacing conventional Euclidean norm by a weighted Euclidean norm:

$$\sum_i w_i \left(x_i - x_{ij}\right)^2 \tag{4}$$

In case of using the normalizing denominator, we shall denote the ANN as Normalized Gaussian Modified Lagrangian ANN (NGML)

The GML/NGML ANNs can be trained using a one shot training procedure as follows. First a representative set of (input, output) centroid patterns are chosen to be memorized by the ANN. The chosen patterns may be determined based on expert's knowledge or using an unsupervised clustering algorithm. The chosen patterns are encoded in the ANN parameters as follows. Each input pattern is stored as a centroid for one of the hidden neurons. The corresponding target is stored as the weight connecting that node to the output nodes. For example, to store $[x_j, y_j]$, set one of the centroids of the hidden neurons to x_j and set the weight connecting it to each output node k to y_{jk}. (Note that by centroid we mean the pattern at which the neuron outputs 1). Just as with conventional Lagrangians, this is done by removing the function $g_j(x)$ centered at x_j from the product in the numerator and denominator of l_j, with the denominator being the result of substituting with $x = x_j$ in the numerator. Moreover, note that conventional have been replaced by the Euclidean norm so that the proposed ANN can handle multi-input/multi-output case.) It is noteworthy that the basis functions in (1) are not the only possible choice. Researchers (Adel et. Al, 2011) have proposed alternative choices and reported good results using those.

2.2 The proposed procedure for initial ship design generation

The proposed system for initial ship design parameters estimation makes use of complete information of examples of ships belonging to a particular category of interest to the designer. NGML ANNs can be trained to find the relation between different design parameters. Thus, after training, they can be used in design by simply presenting them with design constraints. The ANNs generalization capability guarantees that they will produce as output reasonable estimates of the unspecified design parameters. In this section, the different stages for preparing and testing the Multi-ANNs based Design Parameters Prediction System (MADPPS) will be discussed.

a. Construction phase

The different steps of constructing and testing the proposed MADPPS are illustrated in the flow chart in Fig.2. First, enough examples that represent a particular ships category are gathered. These examples will be used to form the database that will used to train, test and validate the ANNs. The information that is stored in the database for each example is the hull displacement (*D*), length overall (*LOA*), draft (*T*), the maximum beam width (*B*) as well as the offset table. Throughout the remaining of the paper, {*D,L,T,B,P*} will be referred to as the "design parameters string". In addition the water plan areas (*WPA*) and sectional areas (*SA*) are stored in the database. As has been illustrated in the introduction, different design cases, dictate different design constraints. Thus, it is desirable that regardless of which parameters values are specified (these are considered constraints), the MADPPS is able to produce reliable estimates of the rest of the unspecified parameters, such that they are consistent with the main theme of the category of ships of interest. To achieve this goal, a system of ANNs is trained using the information in the database. Each ANN is trained to learn the relation between different combinations of inputs and outputs. For example, as shown in Fig.3, one ANN is trained to learn the relation between *D*, *L*,*T* (inputs) and *B*, *P* (outputs), while another ANN will be trained to learn the relation between *D*,*T*,*B* (inputs) and *L*,*P* (outputs). Thus, the number of ANNs trained should be such that they cover all reasonable combinations of inputs and outputs.

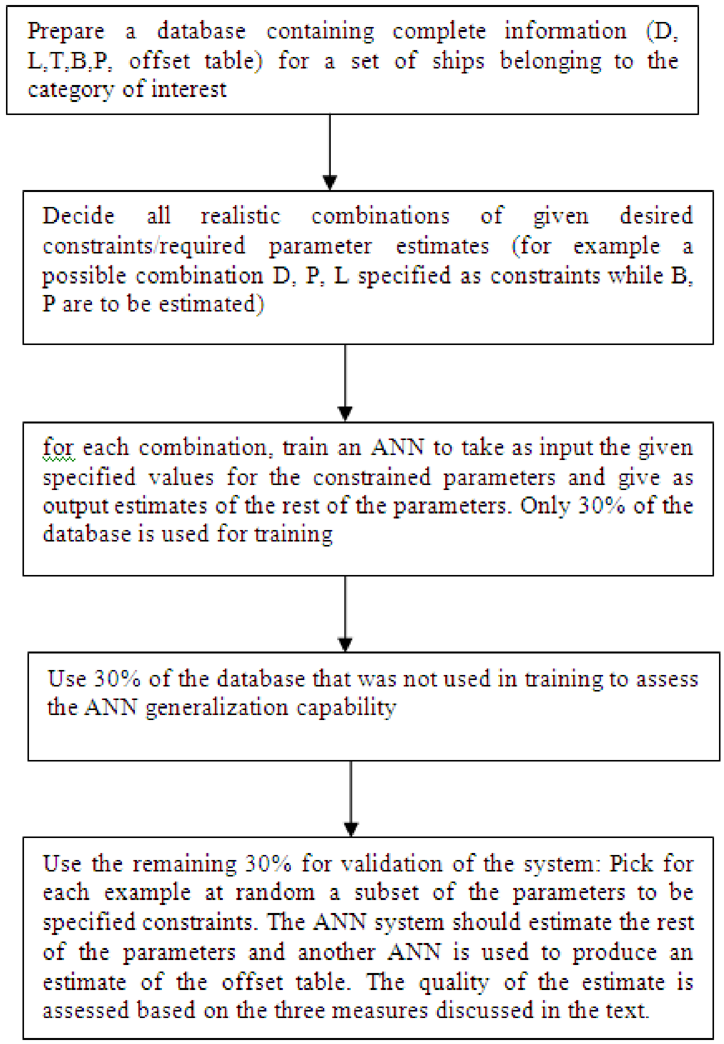

Fig.2: A Flow chart illustrating the steps of constructing and validating the proposed initial design parameters estimation system.

Only some of the examples are used for training (for example, 30%). A different set of examples (another 30%) is used to test the ANN. The error based on the ANNs response to the test examples (that were unseen during training) is a measure of the ANN generalization capability and reliability of the estimates that will be produced when these ANNs are employed. The rest of the database examples are used to validate the MADPPS as a whole. In addition to the ANNs trained to learn the relation between the different parameters, another ANN is trained to learn the relation between *D*,*L*,*T*,*B*,*P* and the corresponding offset table. Thus, this ANN serves as offset table predictor; it takes as input the design parameters string and gives as output the corresponding offset table. A similar successful use of NGMLs to recover offset tables based on some of the hulls parameters have been demonstrated in (El-bastawesy et. al., 2011).

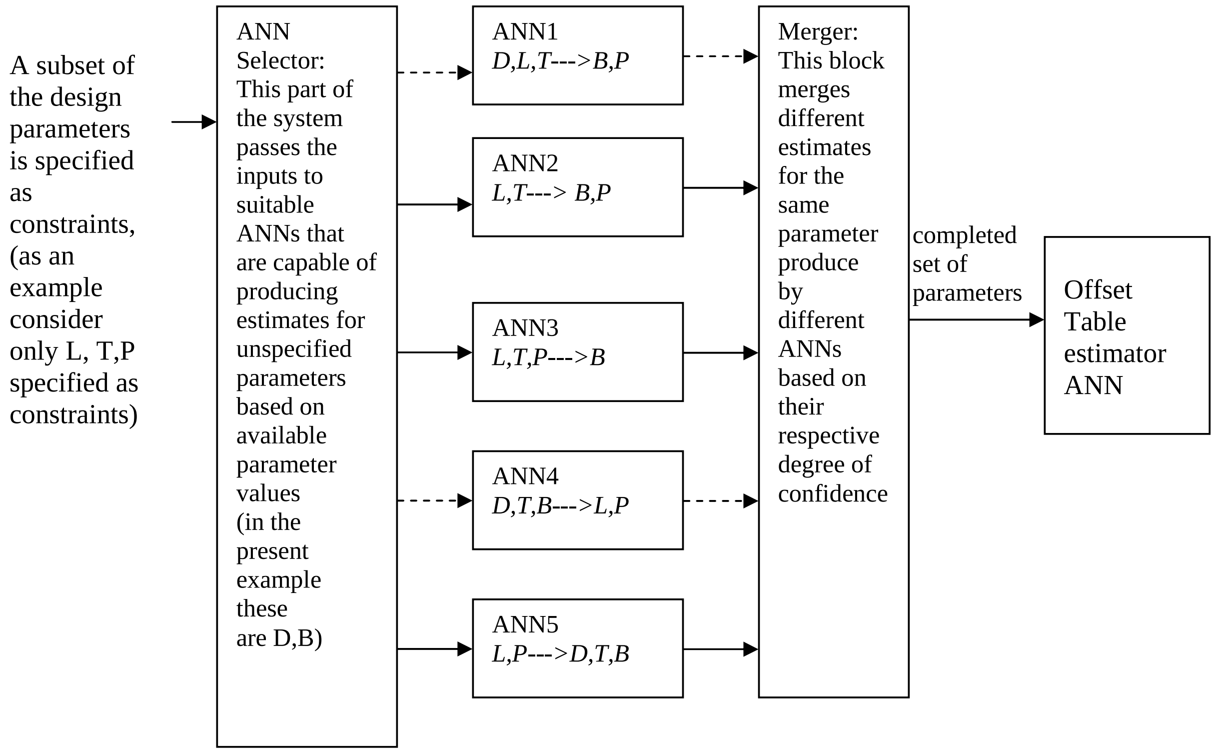

Fig.3: An illustration of how the proposed parameters estimation system is used to generate an initial offset table for a certain case study-A dashed arrow indicates an ANN that is inactive for the current input parameters. This may be because it is only capable of producing estimates for parameters that are already known or because it requires as input a parameter that is unspecified for the current case study.

b. Validation and deployment phase

A typical scenario that illustrates the flow of the validation stage is shown in Fig.3. In the validation stage, the designer makes up different scenarios that depict how the MADPPS will be used in design. This is done by setting at random some of the design parameters of the examples used for validation to zero. The system understands that this means that the system is required to produce estimates for these parameters. An "ANN-selector" passes the design parameters "string" to the ANNs that are capable of producing estimates of the remaining parameters. The capability of the ANN to give an estimate of a certain parameter depends on which combination of inputs/outputs has been used to train it. For example, in Fig.3 in one of the validation examples, the values of *L,T,P* are specified. Thus, the system is required to give estimates for *D,B* that are consistent with those specified parameter. The "ANN-Selector" decides to pass the specified parameters values to ANN2 (which is trained to predict *B,P* based on specified values for L,T), ANN3 (which is trained to predict *B* based on specified values for *L,T,P*) and ANN5 (which is trained to predict *D,T,B* based on specified values for *L, P*). ANN1 is not used because it requires *D* to be known, while ANN4 is not considered because it predicts *L, P*, which are already specified in this example. It can be noted though that some of the ANNs used to produce the desired estimates (ANN2, ANN5) give estimates for parameters that are already specified (P in case of ANN2, T in case of ANN5). These estimates are simply ignored by the MADPPS (However, the difference between these estimates and their specified values can be outputed to the user as a measure of the consistency and reliability of the MADPSS for the example under consideration: a low difference indicates a high degree of consistency and reliability). Moreover, it is clear that ANNs system will produce three different estimates for *B* (from ANN2, ANN3, ANN5). This is resolved using the "Merger" block. This block produces a weighted average of multiple estimates of the same parameter. The weights are taken to be inversely proportional to the ANNs errors on the test data. For example, if the normalized error on test data of ANN2 is 0.1, while that of ANN3 is 0.4 and that of ANN5 is 0.2 then the weighted average estimate of B is computed as follows:

$$B = \frac{(1-0.1)B_2 + (1-0.4)B_3 + (1-0.2)B_5}{(1-0.1)+(1-0.4)+(1-0.2)} \tag{5}$$

where B_2, B_3, B_5 are the estimates for B produced by ANN2, ANN3,ANN5, respectively.

Once the system produces an estimate for *D,B* the whole design parameters string is completed. The complete design parameters string is passed to the ANN offset table predictor.

Note that this same sequence of operations used in the validation will be followed when the MADPPS is used for design. However, during design, the designer will have to trust the system, since the unspecified parameters are truly unknown. However, in the validation stage, the true values of the unspecified parameters are available in the database (they are simply hidden from the system to test the quality of its performance). This gives the designer the chance to assess the quality of estimates produced by the system. Thus, the validation stage is very important to judge the reliability of the system and assess whether or not it can be used in design. In case the validation errors are unsatisfactory, more training examples should be sought and added to the database, the ANNs should be retrained and the system re-validate. In what follows, three different measures for assessing the performance of the MADPPS as a whole based on its response to the validation examples are discussed.

c. Assessment of obtained estimates from the validation stage.

To be able to follow the different assessment measures that will be discussed, three different of parameter values are defined. First, there are the true parameters values available in the database (TV). Second, there are the values of the MADPPS estimates of the parameters (MV). Third, there are the actual values of the parameters (AV). The actual values of the parameters are found by taking the values of *D,L,T,* and the offset table to the computational fluid dynamics (CFD) software. Michlet software is used in this work (Tuck,2008). The software gives the actual values of *B* and *P* consistent with these inputs. Power (*P*) is computed as the product of the total resistance at a certain speed and this speed value. Michlet predicts the total resistance to steady motion of a ship as the sum of a skin friction estimated by the standard ITTC
1957 line and a wave resistance computed by Michell's integral yields quite good results compared to model experiments (Tuck,2008).

i. Direct Errors

Direct errors are computed by comparing the TVs and MVs of the unspecified parameters and the offset table. This is summarized in the following formula:

$$Direct_{Error} = \frac{\sqrt{\sum_{i=1}^{Nup}\left(UP_i^{TV} - UP_i^{MV}\right)^2}}{Nup} \tag{6}$$

where $Nup, UP_i^{TV}, UP_i^{MV}$ are, respectively, the number of unspecified parameters, the TV of the ith unspecified parameter and the MV of the unspecified parameter. In this study the direct error of the parameters in the design parameters string and that of the offset table are calculated separately.

ii. Effective Error

The effective error is a measure of the similarity of certain aspects of the actual performance (computed using CFD software) of the predicted offset table with those of the true values stored in the database. The compared aspects are the values of TV and AV of B, WPA, SA and P. For most applications, a sufficiently low effective error is enough to certify that the MADPPS can be reliably used in design. The average effective error is computed using the following formulas:

$$SAE = \frac{\sum_{i=1}^{Nsa}\left(SA_i^{TV} - SA_i^{AV}\right)^2}{Nsa} \qquad WPAE = \frac{\sum_{i=1}^{Nwp}\left(WPA_i^{TV} - WPA_i^{AV}\right)^2}{Nwp} \tag{7}$$

($WPAE$ $Nwp, WPA_i^{TV}, WPA_i^{AV}$ are the effective error in the WPA, the number of waterlines, the TV and the AV of the WPA of the i[th] waterplane, SAE, $Nsa, SA_i^{TV}, SA_i^{AV}$ are the effective error in the SA , number of stations, the TV and the AV of the SA of the i[th] station, respectively)

The total average effective error is given by:

$$Effective_{Error} = \frac{\sqrt{\sum_{i=1}^{4}\left(B^{TV} - B^{AV}\right)^2 + \left(P^{TV} - P^{AV}\right)^2 + (WPAE) + (SAE)}}{4} \tag{8}$$

iii. Inconsistency Error

The proposed MADPPS produces estimates of the unspecified design parameters and the offset table based on the specified parameter values. Thus, the MADPPS ouput can be interpreted as the following statement: "The output offset table has $B = B^{MV}, P = P^{MV}$ ". For these estimates to be consistent, their values should be close enough to B^{AV}, P^{AV} that are computed using the CFD software when it simulates the output offset table produced by the MADPPS. Thus, the inconsistency error may be defined using the following formula:

$$Consistency_{Error} = \frac{\sqrt{\sum_{i=1}^{2}\left(B^{MV} - B^{AV}\right)^2 + \left(P^{MV} - P^{AV}\right)^2}}{2} \tag{9}$$

3. Results and Discussions

To verify the effectiveness of the proposed MADPPS, the following case study has been adopted. 100 ship examples (offset tables) have been generated using a mathematical series (refer to http://www.cyberiad.net/michlet.htm).

A random number generator has been used to produce the eight parameters for each of the 100 examples as well as the associated values for *D, L,T* . Each example was presented to Michlet CFD software to estimate the values of the corresponding *P* and *B* values. The range of *D* is [61 300] tons. The range of *L* is [25 91] meters. The range of *T* is [0.78 5.5] meters. 5 ANNs were trained using different input output combinations as shown in Fig.3. The inputs to the ANNs were normalized by dividing them by the maximum value of each parameter through the 100 examples. The normalizing factors for the design string {*D,L,T,B,P*} were {300, 91.,5.5, 8.533699, 4822.5612}(The normalizing factors for *B* and *P* are based on the results of simulating all of the 100 hulls using Michlet software). The power *P* for each example was calculated by averaging the power at three speeds (14.6, 15,15.4 m/sec). 33 examples were used for training and a different 33 example were used for test. The average errors for the 5 ANNs on the test data were: 0.0232234, 0.0227662, 0.0424491, 0.0185591, 0.0219834. The average error of the offset table predictor ANN is 0.0019351. The remaining 34 examples were used for validation. The overall average error in the offset tables generated in validation is 0.0018. (Please note that offset table matrices were also normalized by dividing them by the database half maximum beam width). Due to normalization multiplying these errors by 100 gives an indication of a semi-percentage error.

Table.1 lists the details of the results of the validation stage (OTDE, DPDE, EffE, CoE are the offset table direct error, the design parameters direct error, the effective error and the consistency error, respectively). As has been explained in earlier sections, in the validation stage some of the design parameters are set to zero at random. This are called the unspecified parameters. These unspecified parameters are indicated by placing "0" in their places in the input specifications string. The proposed MADPPS completes the design parameters string and passes the complete design parameters string to the offset table predictor ANN. The true information in the database along with the output of the MADPPS and the simulator (Michlet) output in response to the offset table and the values of *L, T, D* of the completed string are used to compute the direct, effective and consistency errors, respectively. The highlighted cells in *Table.1* indicate an unspecified parameter whose value has been estimated by the proposed MADPPS. The errors are considered to be satisfactory since the primary interest of the present work resides in generating a suitable initial design point, whose performance can be fine tuned using available ship design software packages. The reliability of the offset table estimated by the ANN is demonstrated through the effective error computed in *Table.1*. It is clear that the produced offset tables by our multi-ANN systems results in hulls whose performance is close to that dictated by the designer's specifications (low effective error). Lower errors can be attained if the ranges of the design parameters strings are narrowed or if more examples are included. It is also clear from the table that the more the unspecified parameters, the more the deviation from the true values. This is to be expected due to the effect of accumulation of errors in the different predicted parameters on the MADPPS output offset table.

Table.1 Details of the results of the validation stage

Completed String after												
Estimation of Unspecified Parameters					Direct Errors			Effective Errors				
D	L	T	B	P	OTDE	OTDEc	DPDE	DPDEc	EffE	EffEc	CoE	
0.474	0.440	0.576	0.495	0.390	0.007	0.011	0.040	0.066	0.057	0.143	0.087	
0.501	0.438	0.318	0.741	0.478	0.016	0.015	0.058	0.071	0.094	0.154	0.063	
0.685	0.408	0.585	0.385	0.363	0.012	0.014	0.087	0.276	0.133	0.185	0.148	
0.783	0.648	0.756	0.498	0.380	0.009	0.012	0.074	0.192	0.059	0.160	0.125	
0.630	0.606	0.762	0.498	0.379	0.009	0.013	0.092	0.185	0.038	0.169	0.134	
0.687	0.709	0.574	0.415	0.414	0.008	0.012	0.042	0.265	0.034	0.148	0.063	
0.567	0.698	0.667	0.496	0.378	0.012	0.015	0.114	0.193	0.056	0.189	0.144	
0.678	0.471	0.573	0.402	0.439	0.009	0.012	0.080	0.260	0.065	0.163	0.049	
0.285	0.396	0.380	0.512	0.379	0.006	0.012	0.080	0.146	0.024	0.175	0.119	
0.772	0.443	0.195	0.736	0.483	0.014	0.011	0.273	0.220	0.351	0.115	0.621	
0.578	0.299	0.507	0.505	0.380	0.009	0.008	0.086	0.080	0.107	0.139	0.228	
0.733	0.454	0.475	0.555	0.483	0.010	0.010	0.146	0.120	0.075	0.160	0.097	
0.685	0.350	0.581	0.393	0.609	0.007	0.010	0.098	0.193	0.069	0.143	0.128	
0.752	0.586	0.253	0.470	0.341	0.012	0.007	0.289	0.085	0.103	0.100	0.268	
0.683	0.423	0.580	0.394	0.537	0.010	0.008	0.121	0.059	0.097	0.133	0.057	
0.771	0.446	0.645	0.585	0.479	0.007	0.009	0.061	0.122	0.053	0.121	0.104	
0.697	0.791	0.684	0.494	0.357	0.012	0.016	0.303	0.173	0.042	0.193	0.140	
0.690	0.707	0.572	0.402	0.232	0.008	0.010	0.110	0.263	0.056	0.167	0.097	
0.766	0.582	0.278	0.469	0.378	0.011	0.005	0.264	0.078	0.085	0.087	0.229	
0.685	0.604	0.580	0.397	0.627	0.018	0.017	0.216	0.228	0.204	0.175	0.105	
0.442	0.473	0.235	0.502	0.265	0.011	0.011	0.005	0.084	0.076	0.150	0.122	
0.623	0.407	0.560	0.497	0.403	0.012	0.013	0.120	0.082	0.116	0.171	0.083	
0.508	0.780	0.322	0.486	0.338	0.007	0.011	0.113	0.107	0.067	0.141	0.079	
0.689	0.727	0.573	0.402	0.226	0.007	0.009	0.157	0.243	0.091	0.140	0.101	
0.686	0.522	0.602	0.361	0.353	0.010	0.009	0.119	0.096	0.092	0.139	0.065	
0.696	0.631	0.541	0.408	0.312	0.007	0.010	0.056	0.217	0.041	0.161	0.060	
0.773	0.473	0.447	0.507	0.387	0.015	0.011	0.177	0.096	0.111	0.133	0.177	
0.631	0.443	0.545	0.630	0.476	0.005	0.009	0.042	0.069	0.047	0.123	0.092	
0.663	0.560	0.523	0.435	0.446	0.008	0.011	0.065	0.219	0.036	0.136	0.021	
0.435	0.384	0.640	0.496	0.386	0.008	0.011	0.002	0.085	0.050	0.146	0.096	
0.722	0.403	0.676	0.501	0.380	0.008	0.007	0.100	0.113	0.076	0.128	0.140	
0.402	0.461	0.700	0.491	0.305	0.013	0.017	0.249	0.143	0.067	0.206	0.139	
0.708	0.656	0.669	0.497	0.380	0.009	0.012	0.074	0.170	0.038	0.150	0.113	

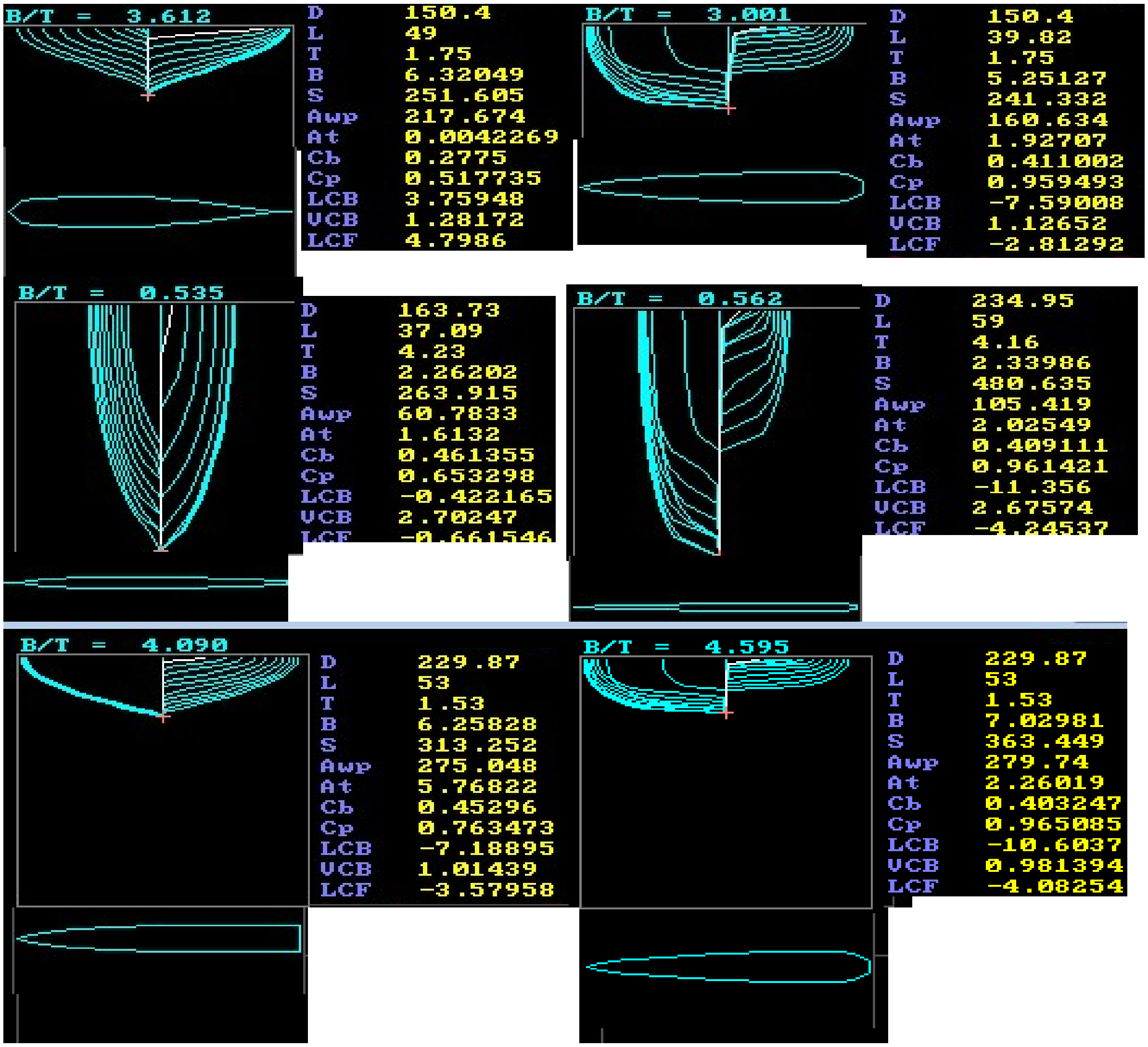

Fig.4: Top to bottom are the results of examples 2, 3, 19 in the validation stage. To the left are the true parameters of the hulls generated using the mathematical series, whereas to the right are the parameters of the hulls produced by the MADPPS

This point is further clarified through Fig.4. Figure 4 shows the results of the validation stage 2nd ,3rd and 19th examples (Top to bottom). To the left are the true parameters of the hulls generated using the mathematical series, whereas to the right are the parameters of the hulls produced by the MADPPS. By comparing these results and taking into account the number of unspecified parameters in each example (refer to the highlighted cells in *Table.1*), the impact of this number on the accuracy of the produced lines and hull parameters is easily noticed. Moreover, during training and testing it was clear that power prediction (P) was the major contributor to the error. This can be improved in future work by training the ANN to predict the power over a wider speed range and taking the average.

To quantitatively assess the merits of our proposed approach over existing approaches, we compared the estimates produced by our proposed strategy with those that would be produced by a conventional design strategy. Conventionally, given a database of existing ship designs and a string of specifications for a new design, a designer finds the closest design in the database (based on Euclidean distance between the specifications of the new design and those in the database) and takes it as the initial design. Ship design software is then used to fine tune it. In *Table.1*, the columns EffEc, DPDEc,OTDEc contain the effective error, direct error in design parameters estimates, direct error in offset table using this conventional design strategy, respectively. It is clear by comparing EffEc (effective error using the conventional strategy) with EffE (effective error using our proposed strategy) and DPDEc,OTDEc (direct errors using the conventional strategy) with DPDE,OTDE (direct errors using our proposed strategy) that our proposed method is capable of producing initial designs whose specifications are closer to those required by a designer. (Note with the conventional design strategy, there is no room for inconsistency since the computed estimates are the actual parameters of the new design's nearest neighbor in the database).

4. Conclusion

The primary aim of the present paper was to propose a suitable strategy for producing an initial ship design based on available specified parameters. The main contributions are summarized as follows. First, we adapted the multi-classifier ANN approach that is typically used in pattern recognition to suite the needs of naval architects. Second, the choice of the ANN architecture allowed for the extremely fast construction of a prediction strategy of satisfactory performance based on a very limited number of examples compared to those required by traditional methods employed in literature. Third, the proposed estimates assessment measures (direct, effective and consistency errors) are expected to prove to be useful for researchers of similar and even different design interests. Future research directions include repeating the design strategy for different series, trying more sophisticated parameters normalization schemes, investigating the effect of the choice and number of considered input/output combinations as well as offering a fuzzy-logic based interpretation of the system decision to the designer so that he/she can assess the experience learned by the system and thus its expected reliability.

ACKNOWLEDGEMENT

Thanks to Allah almighty whose love and support guided and motivated this work. Thanks to all his prophets and messengers who taught us to prove this love by putting much effort in all we do. The authors also would like to thank engineers Ahmed El-bastawesy and Mohey El-deen Ahmed for their supports.

REFERENCES

Abdel-Salam A., B. Salah, A. El-bastaweesy, Islam Adel, A. El-sayed, M. El-laffy, B. Magdy, M. Tarek, M. Mahmoud, I. Abdel-majeed (2009): Designing High Speed Monohull Small Crafts (HSMSC) Using Neural Networks Guided CFD Based Optimization, IEEE OCEANS Conference Bremen, Germany.

Abramowski T. (2010): Application of artificial intelligence methods for improving ship transport efficiency, Scientific Journals Zeszyty Naukowe, Maritime University of Szczecin, Akademia Morska w Szczecinie, 21(93) pp. 5–11.

Adel I., A. El-bastawesy, Omar Fathi, M. Fouad, B. Salem, A. El-sayed, E. Heeba, M. Refaai, WI-TI(2011): An Optimized Green Energy Farm Capable of Solving Egypt Energy Crisis Based on Wind and Marine Current Energies, ICEST 2011 International Conference on Energy Systems and Technologies, Cairo, Egypt. [Available online: http://www.physicsegypt.org/icest2011web/icest11017.pdf]

Alkan A.D.; Gulez K.; Yilmaz H.(2004): Design of a robust neural network structure for determining initial stability particulars of fishing vessels, Ocean Engineering, Vol. 31, pp. 761-777. doi:10.1016/j.oceaneng.2003.08.002.

Bertram, V.; Mesbahi, E. (2004): Estimating Resistance and Power for fast Monohulls Employing Artificial Neural Nets, 4th Int. Conf. High-Performance Marine Vehicles (HIPER), Rome.

Clausen H.B.; Lutzen M.; Friis-Hansen A.(2001): Bjorneboe N.; Baysian and neural networks for preliminary ship design, Marine Technology, Vol. 38, No. 4, pp. 268-277.

El-Bastawesy A., Ahmed El-sayed, Mohamed Abdel-Salam, Basam Salah, Islam Adel, Mohamed Alaa El-laffy, Mohamed Tariq, Beshoy Magdi, Omar Fathi (2011): A New Intelligent Strategy for Optimal Design of High Dimensional Systems, 2011 IEEE GCC Conference & Exhibition, Dubai, United Arab Emirates.

Gougoulidis G. (2008): The Utilization of Artificial Neural Networks in Marine Applications: An Overview. Naval Engineers Journal, 120: 19–26. doi: 10.1111/j.1559-3584.2008.00150.x.

Hamid H., H. Senousy, Mahmoud Abou-Elmakarem (2008): An Improved Fuzzy Logic Controller For Ship Steering Based on Ior Operator and Neural Rule Extraction, ICCES'08, Cairo Egypt.

Hansen A. (2000): Bayesian Networks as a Decision Support Tool in Marine Applications, PhD Dissertation, Department of Naval Architecture and Offshore Engineering, Technical University of Denmark,.

Haykin S. (2008): Neural Networks and Learning Machines, Prentice Hall.

Jankowski S., A. Iozowski, J. Zurada (November 1996): Complex-Valued Multistate Neural Associative Memory, IEEE Trans. On Neural Networks Vol. 7 No.6. doi: 10.1109/72.548176.

Nelwamondo F. and T. Marwala (May 2007): Fuzzy Artmap And Neural Network Approach To Online Processing Of Inputs With Missing Values, Artificial Intelligence.

Polikar R., Lalita Udpa, Satish S. Udpa, Vasant Honavar (2001): Learn++: An Incremental Learning Algorithm For Supervised Neural Networks, IEEE Transactions On Systems, Man, And Cybernetics—Part C: Applications And Reviews, Vol. 31, No. 4. doi: 10.1109/5326.983933.

Schmitz, A., Besnard, E., and Hefazi, H. (2004): Automated Hydrodynamic Shape Optimization using Neural Networks, Paper No. C6 (D19), SNAME Maritime Technology Conference & Expo, Washington D.C.

Tuck E., L. Lazauskas (2008): Drag on a Ship and Michell's Integral , XXII ICTAM, Adelaide, Australia.

Wu J.; Yu M.; Xu C. (1999): A neural network approach for estimating the metallic hull weight of transport ships, International Shipbuilding Progress, Vol. 46, No. 446, pp. 141-150.