MODEL TUNING WITH THE USE OF HEURISTIC-FREE GMDH (GROUP METHOD OF DATA HANDLING) NETWORKS

M.C. Schrijver (*), E.J.H. Kerckhoffs (**), P.J. Water (**), K.D. Saman (*)

(*) Rijkswaterstaat Directie Zeeland
Dpt. HMCZ
Koestraat 30
4331 KX Middelburg, the Netherlands
Email:m.schrijver(k.d.saman)@dzl.rws.minvenw.nl

(**) Delft University of Technology
Fac. of Information Technology and Systems, Dpt. Mediamatics
Mekelweg 4
2628 CD Delft, the Netherlands
Email: e.j.h.kerckhoffs(p.j.water)@its.tudelft.nl

KEYWORDS
Parameter tuning, North Sea model, data mining, dependency modeling, neural networks, Group Method of Data Handling (GMDH).

ABSTRACT

In this paper the use of so-called heuristic-free GMDH (Group Method of Data Handling) networks are considered to tune an unknown parameter (the wind stress coefficient) in a model used for prediction of the meteorological effect on the water level of the North Sea. GMDH tries to find unknown relationships between the parameter concerned and other variables, purely on the basis of measurement data (data mining, dependency modeling). The results achieved, i.e. the prediction capabilities of the GMDH-based tuned model, were found to be reasonably comparable with those from the model that had been previously fine-tuned through a years-lasting tuning procedure; moreover, the method which is generally applicable provided more insight into the physics behind the model concerned.

1. INTRODUCTION

Fine-tuning physical models may be a long lasting procedure, especially when parameters or variables cannot be measured directly and sufficient a priori knowledge about relations with other variables in the model is missing. In this paper we consider the use of GMDH (Group Method of Data Handling) networks, more in particular heuristic-free GHMH networks, to estimate a variable in relation to other variables. It is assumed that large amounts of measured data are available and that the unknown relations we are looking for, are hidden in these data. The resulting data mining problem (more in particular: dependency modeling problem) is solved with heuristic-free GMDH The method is applied to reveal the relevant variables (inputs) on which the considered unknown variable (output) depends upon, as well as a mathematical expression between the unknown output and its found relevant inputs.

The procedure is illustrated and validated on the basis of a realistic example: fine-tuning wind stress coefficients in an operational North Sea model which is used to predict meteorological effects on the water level. Heuristic-free GMDH is applied to reveal the unknown relationships between the wind stress coefficient and other physical variables such as wave energy, wind and pressure. The found relations were included in the model. New forecasts with the modified model were made and compared with forecasts made by the original model. In the original model wind stress coefficients have been determined via a previous long lasting fine-tuning process. Although not improving previous predictions, the GMDH-based fine-tuning method proved to give comparable results. However these GMDH-based results can be achieved in much shorter time than the time needed for the previous fine-tuning process.
The paper is built up as follows: in Sect. 2 a short description is given of both basic and heuristic-free GMDH; the latter was found to give the better results and is therefore used throughout the research. The considered example model tuning problem is dealt with in Sect. 3: an operational "North Sea model" with the wind stress coefficient as the parameter to be tuned. Network training and tuning performances are presented in Sect. 4. A Sect. 5 "Conclusions" completes the paper.

2. THE GROUP METHOD OF DATA HANDLING (GMDH)

2.1 Basic GMDH

The Group Method of Data Handling (GMDH) was introduced by A.G. Ivakhnenko in 1968 and reformulated in 1971 [Ivakhnenko 1971]. GMDH neural networks are typically used to approximate a continuous function $f : A \subset R^n \rightarrow R$; in case of vector functions one network per output element is needed. The typical GMDH network shown in Figure 1 has four inputs (components of the vector x) and one output, the estimate y' of the correct function value $y = f(x)$. The first layer of the network consists of a fanout-input layer. The nodes of this layer distribute their inputs $x_1, x_2, \ldots, x_n$ to the appropriate nodes of the first hidden layer. Every node in each layer following the fanout-input layer receives two inputs, which are outputs of the nodes of the previous layer. The scheme of the GMDH is a simple feed-forward sequence. The components of the input vector x are first supplied to the

input units. These then distribute them to the appropriate first hidden layer processing elements. The outputs of these hidden layer processing elements are then supplied to the next layer, and so on. The final output of the network is a single real number *y'*.

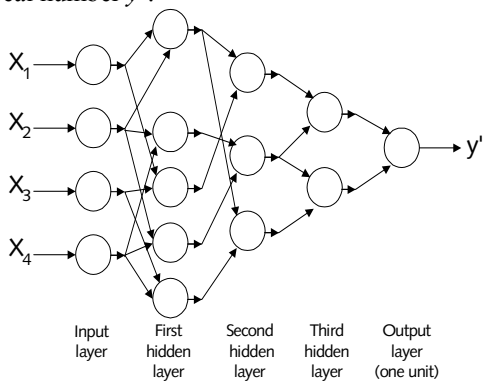

Figure 1: Example of a Typical GMDH Neural Network

Except for the input layer, all processing elements in the *k-th* layer (k = 2,3,…) have the configuration shown in Figure 2. The output signal of a layer-k processing element l is given by the quadratic transfer function:

$$z_l^k = a_l^k \left(z_i^{k-1}\right)^2 + b_l^k \left(z_i^{k-1}\right)\left(z_j^{k-1}\right) + c_l^k \left(z_j^{k-1}\right)^2 + d_l^k z_i^{k-1} + e_l^k z_j^{k-1} + f_l^k \qquad (2.1)$$

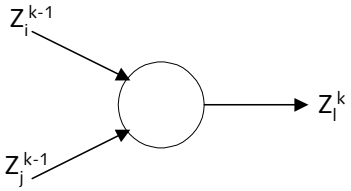

Figure 2: GMDH Processing Element l of Layer k

Thus, the GMDH network as a whole builds up a polynomial function of the input components. The output *y'* of the network can be expressed as a polynomial of degree 2^K (the so-called Ivakhnenko polynomial), where K is the number of layers in the network following the input layer.

The GMDH network is developed by starting at the input layer and growing the network progressively towards the output layer, one layer at the time. Starting with k = 1 and proceeding to k = 2 until the entire network is configured, the process used is the same, regardless the value of k. Layer k is configured with one processing element for each different pair of outputs from the previous layer. Assuming that the previous layer contains M_{k-1} processing elements, the number of processing elements in layer k will be the number of combinations of the M_{k-1} outputs, taken two at a time: $C(M_{k-1},2)$, where

$$C(M_{k-1},2) = \binom{M_{k-1}}{2} = \frac{M_{k-1}!}{2!\cdot(M_{k-1}-2)!} \qquad (2.2)$$

The basic idea of GMDH is that each of these processing elements with transfer function (2.1) wants to have its output *y'* match $y=f(x)$ as closely as possible for each network vector input x. This approximation is accomplished through linear regression. The six coefficients of each processing element are found using a large set of input/output examples (x_1, y_1), (x_2, y_2), ... (x_p, y_p). These determine numerical values to be entered into the transfer function of the l^{th}-processing element of layer k, which finally results in a system of linear equations in the coefficients $a_l^k, b_l^k, c_l^k, d_l^k, e_l^k, f_l^k$:

$$Z_l \left[a_l^k b_l^k c_l^k d_l^k e_l^k f_l^k\right]^T = Y \qquad (2.3)$$

where Z_l = the matrix of numbers in which each row corresponds to the values in equation (2.1) and Y = the output of the processing element (the estimate *y'* of the correct function value $y = f(x)$).
The set of linear equations expressed by (2.3) will almost never have an exact solution, only an approximate one:

$$\left[a_l^k b_l^k c_l^k d_l^k e_l^k f_l^k\right]^T = Z_l^+ Y \qquad (2.4)$$

where Z_l^+ = the pseudo inverse of the matrix Z_l. The solution $a_l^k, b_l^k, c_l^k, d_l^k, e_l^k, f_l^k$ represents the best possible selection of coefficients in terms of minimizing the mean squared error between the output of each processing element and the desired output.

After the six coefficients for each of the processing elements of layer k have been derived, the overall performance of each processing element in terms of its goal is evaluated through cross-validation. For this purpose a test set is used which is completely different from the data set used for training. For each processing element the so-called regularity criterion:

$$F = \frac{1}{r}\left\| Z_l' \left[a_l^k b_l^k c_l^k d_l^k e_l^k f_l^k\right]^T - Y^i \right\|^2 \qquad (2.5)$$

is determined. The next step in the GMDH process is to eliminate those processing elements on layer k that have a "large" regularity criterion value, whereby "large" is user defined. The final set of remaining processing elements then supplies the Z^k output vector that feeds layer k+1. When the layer is adapted it is frozen, and the process continues with the next layer. The process of adding new layers stops when the stop criterion is met, which normally is the case when the regularity criterion value of the best performing processing element in a layer reaches its minimum.

2.2 Heuristic-free GMDH

The GMDH network is not free of heuristics. Heuristics are needed with respect to the predetermination of the structure of the partial polynomials, the subdivision of the observation data into training data and validation data, and the predetermination of the number of nodes selected in

each layer. To find the optimal combination of these heuristics the GMDH algorithm must be executed many times, each time for a different combination. This gives a huge computational overhead, whereby unfortunately the real optimal combination is rarely found. A solution is a revised GMDH algorithm that was developed by Tamura and Kondo [Tamura and Kondo 1984]. This algorithm is free of heuristics. All the data are used for both training and testing which eliminates the problem of subdividing the data into training data and test data. Instead of using the regularity criterion, the so-called Prediction Sum of Squares (PreSS) is calculated. This criterion is used for generating optimal partial polynomials, selecting intermediate variables and stopping the GMDH algorithm.

The heuristic-free GMDH algorithm consists of four steps:

Step 1:
Generating optimal partial polynomials in each layer.

Optimal partial polynomials are generated through the polynomial generator G_1, G_2, G_3 or G_4, applying a stepwise regression procedure to the polynomial:

$$G_1 : \begin{cases} y_k = a_0 + a_1 x_i + a_2 x_j + a_3 x_i^2 + \\ \qquad a_4 x_i x_j + a_5 x_j^2 \\ y_k = a_0 + a_1 x_i + a_2 x_j + a_3 x_i^2 + \\ \qquad a_4 x_i x_j \\ \quad \vdots \end{cases} \tag{2.6}$$

$$G_2 : y_k = a_0 + a_1 x_i + a_2 x_j \tag{2.7}$$

$$G_3 : y_k = a_0 + a_1 x_i + a_2 x_i^2 \tag{2.8}$$

$$G_4 : y_k = a_0 + a_1 x_i \tag{2.9}$$

In this stepwise regression procedure, the above-mentioned PreSS is used as the criterion for selecting the best partial polynomial:

$$\mathrm{PreSS} = \sum_{k=1}^{m} \left[\frac{y_k - \hat{y}_k}{1 - x_k^T \left(X^T X \right)^{-1} x_k} \right]^2 \tag{2.10}$$

where:
$x_k = \left[1, x_{ik}, x_{jk}, x_{ik}^2, x_{jk}^2, x_{ik} \times x_{jk}\right]^T$ for k=1,…m and $X = [x_1, x_2, \ldots, x_m]^T$, m = the number of training observations, y_k = the kth observed value for the output variable, x_{ik} = the kth observed value for the input variable x_i and $\hat{y}$ = the kth estimated value obtained by a regression of one of the partial polynomials against all available training data.

Step 2:
Selecting the nodes.

P nodes with the smallest PreSS are selected from all the nodes in the layer. The number p is preferred to be as large as possible within the computational capacity of the computer. Therefore, p is determined by computational criteria and not by heuristics.

The outputs of the selected nodes are regarded to be the inputs of the next layer. Of each layer, the smallest PreSS is kept for later use as the stopping criterion for the iterative computation.

Step 3:
Stopping the multi-layered iterative computation.

The iterative computation of the model is terminated when the PreSS cannot be further improved, or all selected partial polynomials are generated by G_4.

Step 4:
Computation of the predicted values.

The final model describing the relationship between the input and the output variables can be obtained in two ways:

1. Select in the final layer the node with the smallest PreSS. The output of this node is the output of the final model.

2. Select the L nodes with the smallest PSS as was done in the preceding layers, and calculate the weighted average of these nodes.

In the training executed during this study, the first method was used.

2.3 GMDH-based Dependency Modeling

The dependency modeling problem we were confronted with in this research, is to discover on the basis of (a large amount of) measurement data which entities of an a priori selected set of entities x_j (where $j \in J = \{1, 2, 3, \ldots, N\}$) have relevant dependency to a certain variable y and to find a descriptive mathematical expression between y and those found relevant entities x_i (where $i \in I \subset J$). In this description (for different values of i) x_i can be different variables and/or the same variables at different time instances; i.e. there might be, for instance, a dependency between y(t) and z_1(t-3), z_2(t) and z_3(t-1): $y = f(x_1, x_2, x_3)$ with $x_1 = z_1(t-3)$, $x_2 = z_2(t)$, $x_3 = z_3(t-1)$. The data that are assumed to hide the unknown relationships are measurements of the variables (entities) y and x_i at many time instances.

From the description of Sect. 2.1 on how to build up a GMDH-network it is clear that during training the subsequent elimination of processing elements (neurons), that do not perform well on the basis of their regularity criterion values, has as final result that irrelevant relationships are cut off and therefore irrelevant entities x_i are removed: the value y of the single output neuron is a polynomial of only the relevant entities x_i.
It is easy to understand that the dependency modeling capacities of heuristic-free GMDH (usually considerably) outpaces those of basic GMDH; this was also found in practice. However, for this a price has to be paid: heuristic-free GMDH is much more computing intensive than basic GMDH which in itself is already computing intensive! In

this research we have used a home-made parallel implementation of heuristic-free GMDH (a slightly modified version of the one considered in Sect. 2.2), running on a (hypercube) parallel computer and/or a network of UNIX machines [Water and Kerckhoffs 1999].

3. MODEL TUNING PROBLEM

3.1 An Operational "North Sea Model"

We illustrate and evaluate the considered GMDH-based data mining method to fine-tune physical models on the basis of an example. The example concerns a numerical model of the North Sea and parts of the adjoining waters (see also [Schrijver et al. 2001]). This so-called North Sea model is based on linearized shallow water equations:

$$\frac{\partial u}{\partial t} + g\frac{\partial h}{\partial x} - fv + \lambda\frac{u}{D} - \gamma\frac{V^2\cos\psi}{D} + \frac{1}{\rho_w}\frac{\partial p_a}{\partial x} = 0 \qquad (3.1)$$

$$\frac{\partial v}{\partial t} + g\frac{\partial h}{\partial y} + fu + \lambda\frac{v}{D} - \gamma\frac{V^2\sin\psi}{D} + \frac{1}{\rho_w}\frac{\partial p_a}{\partial y} = 0 \qquad (3.2)$$

$$\frac{\partial h}{\partial t} + \frac{\partial (Du)}{\partial x} + \frac{\partial (Dv)}{\partial y} = 0\,, \qquad (3.3)$$

where t = time, h = meteorological effect on the water level, (u,v) = water velocities in the x- and y-direction, D = depth of the water, f = Coriolis parameter, λ = linear bottom stress coefficient, γ = wind stress coefficient, V = wind velocity, ψ = direction of the wind with respect to the positive x-axis, ρ_w = density of water, p_a = atmospheric pressure, and g = acceleration of gravity. The model is used to predict the meteorological effect on the water level, which is extremely important for decision support purposes. The North Sea model was extended with a Kalman filter which has improved the prediction accuracies. The prediction capabilities of the model can be validated by comparing predicted and actually measured water levels at certain fixed measurement locations.

Discretization of the model in space yields to a spatial grid; the used size of each cell in this grid is $\Delta x = \Delta y = 33{,}8$ km. At each grid point the discrete differential equations are solved for discrete times t_i with $\Delta t = t_i - t_{i-1} = 10$ min. In the considered model the North Sea is subdivided into 6 areas (1; South part, 2: the Channel, 3: Middle West part, 4: North West part, 5: North East part, 6: Middle East part). Each area has its own pressure, wind speed and wind direction, which are available every three hours and interpolated (in space and time). On the basis of an atmospherical numerical model a forecast of pressure p_a, wind speed V and wind direction ψ as functions of space and time is calculated. Since the grid of the latter model is coarser than the grid used by the North Sea model and pressure and wind data are in the forecast only available every 6 hours, both spatial and time interpolations are used to obtain pressure, wind speed and wind direction on the grid points of the North Sea model. Since the grid is only defined for the North sea, external surges are modeled through the correct initial values at the grid boundaries. However, these correct initial values are unknown. The North Sea model was extended with a steady state Kalman filter to solve this problem.

As said, the North Sea model is used for tide prediction (meteorological effect on the water level). In preparation of any forecast, first a hindcast is made: the model is executed with known input data from a certain moment on in the past until current time. The hindcast results in the model's state vector representing the actual situation at current time. When running the hindcast, known input data is used to perform the measurement update of the Kalman filter. After the hindcast the forecast is made with as starting point the model's state vector built from the hindcast. The Kalman corrections on the wind and boundary conditions are also input parameters for the forecast; these corrections certainly have influence especially during the first hours of the forecast, but are then smoothed by the model. The forecast is done by running the model with as input the calculated expected future pressure and wind (in the different wind areas), starting at the time the hindcast has finished up to a number of hours ahead. After having ran a forecast, at time t+n a new hindcast/forcast cycle can be performed, thereby making a new prediction over the next n hours, and so on.

3.2 The Wind Stress Coefficient γ

Most parameters in the shallow water equations (3.1), (3.2) and (3.3) are physical constants or measurable variables. Two variables, however, the bottom stress coefficient λ and wind stress coefficient γ, are physical quantities that cannot be measured. During previous tuning of the North Sea model it appeared that the bottom stress coefficient λ has a much lower influence on the meteorological effect than the wind stress coefficient, therefore λ is modeled as a constant and is in this research beyond further investigation. Since the wind stress coefficient γ is obviously dependent on the wind speed and any of the afore-mentioned wind areas has its own wind speed, in the model different γ's are assumed. Two areas (North East and Middle East parts) were found to have a minor impact on the meteorological effect. Therefore we will try to establish a relation between the four relevant wind stress coefficients and other variables. Commonly, a relation between the wind speed and the wind stress coefficient is used, for instance the Charnock relation. However, it could well be that there are other variables that may influence the wind stress coefficient. Since measurements of several physical variables were available for many years, the problem can be viewed as a data mining problem: find those physical variables the wind stress coefficient essentially depend on, as well as find a mathematical relation between them, as all this information is supposed to be hidden in the stored physical measurements. Having found an expression for the relation between γ and other physical variables, it can be included in the model and the model's predictions can be validated.

4. GMDH-BASED MODEL TUNING

4.1 Data Sets To Train The GMDH Networks

For each γ_i (γ_i is the wind stress coefficient for the i-th wind area; i = 1, ..4), a separate GMDH network is used. In order to train a network, data was gathered in the time period 01/01/1998 – 30/09/1999 (i.e. 638 days, with a resolution of 6 hours resulting in 4 samples a day, so all together 2552 data records). Each record in the training set (for a single net) contains input/output data at a certain time t_j (see figure 3). The 18 measured *basic* inputs are: pressure $p_1 - p_4$ in each of the 4 areas, the x- and y-components of the wind in these areas, the significant wave height at two special locations L1 and L2, where measured wave data was available, the high frequency wave energy (> 0.2 Hz) at these locations L1 and L2, and the medium range frequency wave energy (0.1-0.2 Hz) at L1 and L2. Moreover, *additional* inputs were included: the differences of all the basic inputs at time t and t-6, as well as each basic input shifted backward in time over a period of 6, 12, 18 and 24 hours; so all together we have 18 + 18 + 18*4 = 108 inputs. The output data in each record is obviously γ_i. The values of γ_i were found by running (i.e. performing hindcast and successive forecast) the model on a 4-times a day basis during the entire indicated period and for a predetermined number of 7 different values of γ_i on its definition interval $[0;6x10^{-6}]$, and then determining that value of γ_i that minimizes the difference between predicted and actually measured meteorological effect. In summary, the training set for each separate net consists of 2552 records, each having 108 input items and 1 output γ_i. (see Figure 3). It is the task of the trained GMDH network to reveal which of these 108 inputs are actually relevant for γ_i and to provide a mathematical expression $\gamma_i = f(x_1, x_2, .., x_N)$ in these found relevant inputs.

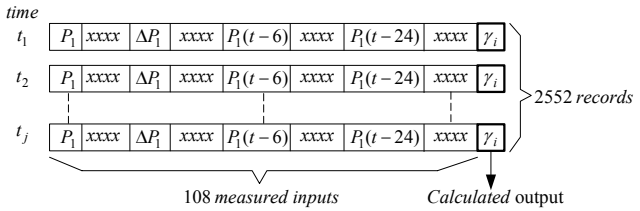

Figure 3: Structure of the Used Data Set

In addition to the above training, the networks are also trained with measurements in the period 24/02/1998 – 14/05/1998 (with a resolution of 1 hour, finally resulting in 1921 data records). Here each record contains, in addition to the 18 above-mentioned basic inputs also the pressure, wind speed and wind direction in the areas North East and Middle East, the differences at time t and t-1 as well as each input shifted backwards in time over a period of 1, 2, …, 24 hours, so all together 24 + 24 + 24*24 = 624 inputs. Now the trained GMDH is assumed to reveal the relevant inputs out of these 624.

4.2 GMDH Performance

GMDH networks of various size (1- 5 layers) have been trained for each γ_i (i = 1, 2, 3, 4), using both the data gathered for the period 01/01/1998 – 30/09/1999 (6 hours resolution) and the period 24/02/1998 – 14/05/1998 (1 hour resolution). In the training of the networks an important role was played by the so-called "condition qualifier", which defines the sensitivity of the matrix in the algorithm. By enlarging this value, which is normally set to 10^6, more dependency between the inputs is assumed. A disadvantage of enlarging this value, however, is that the output becomes more sensitive to noise on the inputs. In our research the condition qualifier was set to 10^{12}.

In order to validate the result of the training, the mathematical relations between γ_i and the relevant inputs found were included in the North Sea model. With this modified model a series of forecasts was made. Every 10 minutes (in simulation time) the predicted meteorological effect on the water level was stored and afterwards compared to the actually measured meteorological effect through the well-known RMSE (Root Mean Square Error) criterion. Similar forecasts with RMSE estimations were also made using the currently operating North Sea model with γ–values that have been found after a many years lasting, difficult tuning process and that have proved to give satisfactory prediction results. We are obviously interested in the difference ΔRMSE between both these RMSE performances. In all cases the results in terms of RMSE obtained from predictions with GMDH-based γ's were comparable with (actually however a little less than) those for the fine-tuned model; however, the interesting point is that the differences in percentage proved to be relatively small so that GMDH can be considered worthwhile for quickly estimating reasonable γ 's purely on the basis of measurements. Table 1 presents results of ΔRMSE for one GMDH-based γ_i (i = 1 – 4) while the other three γ 's were not changed in the North Sea model. Table 2 shows ΔRMSE-figures for the case that all four γ 's were GMDH-based. Note that in the experiments the best results with either 1-layer or 2-layer networks are between 7 and 8 %. In both tables, the upper placed figures reflect the result for the data set that covers the period 01/01/1998 - 30/09/1999 (6 hours resolution) and the lower placed figures the result for the other data set covering the period 24/02/1998 - 14/05/1998 (1 hour resolution); in case of only one figure, the result concerns the second mentioned data set.

Table 1: Results of ΔRMSE for one GMDH-based γ_i

	γ_1	γ_2	γ_3	γ_4
ΔRMSE (Prediction over 5 months, 1-layer GMDH)	0,1% 1,9%	4,5% 3,3%	0,3% 0,9%	0,1% 0,1%

Table 2: ΔRMSE for the case that all four γ 's were GMDH-based

	GMDH (1-layer)	GMDH (2-layer)	GMDH (3-layer)	GMDH (4-layer)	GMDH (5-layer)
ΔRMSE (Prediction over 5 months)	7,6% 7,7%	7,9% 31,6%	* *	* 54%	* *
ΔRMSE (Prediction over 2 months)	11,7%	7,5%	17,6%	27,2%	*

5. CONCLUSION

In this research heuristic-free GMDH networks are used to fine-tune a North Sea model for wind stress coefficients in four different wind areas. Although the GMDH-based dependency modeling method did not improve the prediction capabilities of the current North Sea model, the eventual differences in prediction performance were relatively small. Considering the fact that the current model has been extensively tuned during several years, it is an interesting result that roughly the same performance can be achieved with a completely different technique. The GMDH data mining method therefore proved to be usable to achieve reasonably good results, with the additional advantage that the period of time for collecting training data, training and validating is much shorter than the time (several years) that has been necessary in the past to fine-tune the current model. Summarizing, GMDH-based dependency modeling can be fruitfully used for a fast reasonable estimate of one or more parameters that cannot be directly measured and that lack (sufficient) a priori knowledge on relational dependencies.

Another interesting aspect of the GMDH-based dependency modeling is that unexpected variables might be found to have relevant impact on the wind stress coefficients. The procedure may thus increase physical insight.
Our research has shown that wind stress coefficients considerably depend on wave energy, which confirms certain assumptions with experts in the field and can be made explicable by physics. It means that the usually assumed dependence of the wind stress coefficients on only wind speed and wind direction is not optimal and can be improved.

Although no a priori knowledge about the relationship between the inputs and the output is needed, a thorough knowledge of the system being modeled is necessary. This knowledge is needed when creating a valid data set and especially when verifying training results. One cannot use heuristic-free GMDH for dependency modeling without knowledge of the system being modeled.

REFERENCES

Ivakhnenko, A.G. 1971. "Polynomial theory of complex systems", IEEE Trans. Systems Man Cybernet, SMC-1(4), pp. 364-378.

Mueller, J.A. 2001. "Automatic model generation based on GMDH. In: E.J.H. Kerckhoffs and M. Snorek (eds.): Modelling and Simulation 2001 (Proc. of the 15th European Simulation Multiconference, Prague, 2001), SCS Europe, pp. 661-668.

Schrijver, M.C., Saman K.D. and Kerckhoffs E.J.H. 2001. "Fine-tuning of a North Sea model with the aid of GMDH neural networks". In: N. Giambasi and C. Frydman (eds.): Simulation in Industry 2001 (Proc. of the 13th European Simulation Symposium, Marseille, France, 2001), pp. 696-704.

Tamura, H. and Kondo T. 1984. "On revised algorithms of GMDH with applications". In: S.J. Farlow (ed.): Self-Organizing Methods in Modeling: GMDH Type Algorithms, Vol. 54 of Statistics: Textbooks and Monographs, Marcel Dekker, Inc., chapter 12, pp. 225-241.

Water, P.R. and Kerckhoffs E.J.H. 1999. "GMDH-based financial forecasting on a hypercube parallel computer. In: Simulation in Industry (Proc. of the 11th European Simulation Symposium, Erlangen, Germany, 1999), SCS Europe, pp. 586-596.