

Self-Organizing Polynomial Networks for Time-Constrained Applications

Ivan Marić, *Member, IEEE*, Ivan Ivek, *Student Member, IEEE*

Abstract – A meta-modeling of complex calculation procedures (static systems) is investigated with the aim to create the low complexity surrogate models applicable in low computing power real-time measurement systems. Unlike the single parameter error criteria and MDL measure, the proposed compound squared relative error measure forces the GMDH algorithm to prefer the models having the smallest compound deviation of the accuracy and the execution time from the given thresholds and thus generally leads to more favorable models with respect to both conditions. The approximation errors, the execution speed and the applicability of derived GMDH models in real-time flow-rate measurements of natural gas are discussed and compared with the corresponding models derived by ANN and SVR.

Index Terms - approximation methods, GMDH, real-time systems, modeling, embedded systems

I. INTRODUCTION

In measurements that involve complex numerical procedures for the reconstruction of physical quantities from measured data in Real-Time (RT) the required computing power of the embedded system might be a limiting factor for the implementation. This can lead to a situation where the implementation of the procedures based on “first principles” is unrealistic in economic terms. In these cases it is reasonable to construct and to implement the meta-models [1] accurate enough not to deteriorate significantly the overall measurement accuracy and simple enough to be applicable in Low-Computing-Power (LCP) systems. Meta-modeling [1] is a procedure for deriving an acceptable low-complexity surrogate of a complex model. The overview of the approximation methods and meta-modeling techniques used in system engineering and simulations is given in [2]. Here we focus on modeling the nonlinear static systems by an algebraic equation.

Manuscript received July 22, 2008. Accepted for publication May 19, 2010.

This work was supported by the Croatian Ministry of Science, Education, and Sports through the projects: “Computational Intelligence Methods in Measurement Systems”, No. 098-0982560-2565 and “Real-Life Data Measurement and Characterization”, No. 098-0982560-2566.

Copyright © 2009 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org

I. Marić and I. Ivek are with the Ruđer Bošković Institute, Division of Electronics, Zagreb 10002, Croatia (phone: +385-1-4561191, fax: 385-1-4680114, e-mail: ivan.maric@irb.hr).

The use of artificial neural networks (ANNs) for the approximation of multidimensional nonlinear functions is analyzed in [3] where the authors conclude that ANN training is straightforward and in certain cases can be superior to the contemporary optimization techniques with respect to the execution speeds and the generalization properties. Most recent advances in training Arbitrarily Connected Neurons (ACNs) [4] show that fully connected networks need even smaller number of neurons than standard multilayer perceptrons to fulfill the same task. An ANN is usually programmed using fast digital signal processors thus enabling RT applications such as precise measurement [5], intelligent control [6] or accurate modeling [7] of nonlinear structures.

ACRONYMS

ACN	Arbitrarily Connected Neurons
ANN	Artificial Neural Network
CE	Compound Error
ET	Execution Time
FC	Flow Computer
FP	Floating Point
GMDH	Group Method of Data Handling
JT	Joule-Thomson
LCP	Low-Computing-Power
MDL	Minimum Description Length
MLP	Multi-Layer Perceptron
MPU	Microprocessor Unit
RAE	Relative Absolute Error
RBF	Radial Basis Function
RMSE	Root Mean Squared Error
RRSE	Root Relative Squared Error
RS	Reduced Set
RT	Real Time
SV	Support Vector
SVR	Support Vector Regression

Also, the analysis of the Support Vector Regression (SVR) [8], using various error measures, shows its efficiency in approximation of complex systems and in meta-modeling of complex engineering analyses. The SVR produces more accurate and more robust models when compared to other meta-modeling techniques [8]. But, in spite of high approximation accuracy, the embedding of nonlinear activation or kernel functions in software and the associated computational burden for the microprocessor unit (MPU) might become unacceptable when considering an

implementation of ANN or SVR models for RT calculations on LCP measurement systems.

Approximation of complex systems by heuristic self-organizing polynomials, known as Group Method of Data Handling (GMDH), was introduced by A.G. Ivakhnenko [9]. The GMDH is essentially a self-organizing data mining technique based on the automatic generation of multilayer polynomial structures from low-order two-dimensional polynomials, which are combined automatically to produce the optimum network structure using the principles of evolution (inheritance and selection). The obtained models are easy to embed in digital computers by implementing only the Floating Point (FP) addition and FP multiplication. The approximation accuracy can be increased when combining the method with genetic programming and backpropagation [10], or with sigmoid functions and feedback loop [11]. When applied to RT compensation of nonlinear behavior [12], the self-organizing nature of GMDH may eliminate the complicated structural modeling and parameterization, common to conventional modeling strategies.

The GMDH model performances are typically evaluated by the single parameter measures [13], mostly by the sum of squared errors, which minimizes the approximation error rather than the complexity of the model. The Minimum Description Length (MDL) performance measure [14] provides a tradeoff between the accuracy and the complexity but is not particularly useful when customizing the models for RT calculations. Here we propose a simple and intuitive Compound Error (CE) measure, which proves to be more effective than the MDL and the single parameter measures when training the GMDH models for time-constrained applications.

In the following section a brief overview of the GMDH algorithm is given including the proposed CE measure for model selection. Section III describes the preparation of training and validation examples of thermodynamic properties of natural gas using complex calculation procedures. The analysis of GMDH surrogate models obtained with different performance measures and an example of application in RT flow-rate measurements is given in Section IV. Finally, in Section V, the GMDH is compared to ANN and SVR with respect to both the accuracy and the computational cost of the models.

II. GMDH ALGORITHM

The algorithm [9] combines the variables (attributes) from lower layers in order to derive the approximation polynomials at the current layer. A simple example of a polynomial network, generated by the GMDH algorithm, is illustrated in Fig. 1 where x_1, \dots, x_4 denote the input variables and p_1, \dots, p_5 are the network nodes, each representing the corresponding low-order two-dimensional polynomial. Each layer $\lambda=1, \dots, L$ may contain up to P_λ nodes. The derived polynomial functions belong to the class of Kolmogorov-Gabor polynomials

$$P = a_0 + \sum_{i=1}^M a_i x_i + \sum_{i=1}^M \sum_{j=1}^M a_i a_j x_i x_j + \dots, \quad (1)$$

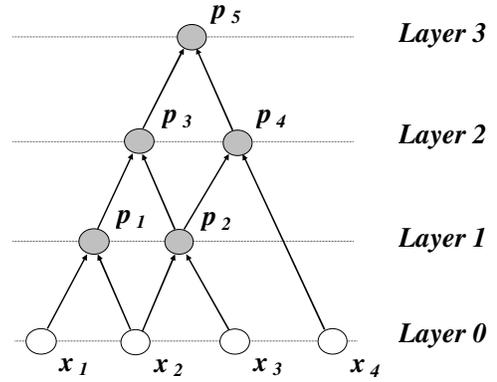


Fig. 1. Illustration of a polynomial model generated by the GMDH algorithm

where P represents the model of a multidimensional system, x represents the input variable and a denotes the coefficient. To simplify the representation, we write down the models in a recursive form, e.g. the polynomial model p_4 from Fig. 1 can be written as: $p_4 = p_4(p_2(x_2, x_3), x_4)$. The polynomials are derived by a multivariate regression using a training data set and evaluated on a validation data set.

A. Polynomial regression

The regression problem can be formulated in the following way. Given a training data set

$$\mathcal{D}_t = \{(\mathbf{x}_{ii}, y_{ii})\}_{i=1}^M \quad (2)$$

of data vectors $\mathbf{x}_{ii} = (x_{ii_1}, x_{ii_2}, \dots, x_{ii_K})$, $x_{ii} \in \mathcal{R}^K$, and the corresponding dependent variable $y_{ii} \in \mathcal{R}$, find the best function model $y = f(x)$ which on average converges to the true unknown mapping $f(x)$ [10]. Consider, for example, a two-dimensional second-order polynomial p_k to be a basic GMDH building block:

$$p_k = a_{k0} + a_{k1}z_{k1} + a_{k2}z_{k2} + a_{k3}z_{k1}^2 + a_{k4}z_{k2}^2 + a_{k5}z_{k1}z_{k2}, \quad (3)$$

where k identifies the polynomial within all aggregated polynomials, with the corresponding input variables z_{k1} and z_{k2} and the coefficients a_{k0}, \dots, a_{k5} obtained by a linear regression with the least-squares estimator. Note that, similar to ACNs [4], the variables z_{k1} and z_{k2} in (3) can be the output from any network node from lower layers including the initial variables x from ‘Layer 0’ (see Fig. 1). In order to calculate the coefficients, a_{k0}, \dots, a_{k5} , of the polynomial (3), a set of 6 simultaneous linear equations

$$\left\{ \frac{\partial}{\partial a_{kj}} \sum_{i=1}^M (y_{ii} - p_k)^2 = 0 \right\}_{j=0}^5 \quad (4)$$

must be solved, where M is a total number of training samples

and y_{vi} is the i -th value of dependent variable from the training set. From (3) and (4) one can obtain the matrix form $\mathbf{y} = \mathbf{Z}\mathbf{a}$ for a least squares fit, which, after pre-multiplying by the matrix transpose \mathbf{Z}^T to get the matrix equation $\mathbf{Z}^T\mathbf{y} = \mathbf{Z}^T\mathbf{Z}\mathbf{a}$, can be solved by obtaining the coefficient vector $\mathbf{a} = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{y}$, where

$$\mathbf{Z} = \begin{bmatrix} 1 & (z_{k1})_1 & (z_{k2})_1 & (z_{k1}^2)_1 & (z_{k2}^2)_1 & (z_{k1}z_{k2})_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & (z_{k1})_M & (z_{k2})_M & (z_{k1}^2)_M & (z_{k2}^2)_M & (z_{k1}z_{k2})_M \end{bmatrix},$$

$$\mathbf{y}^T = [y_{v1} \ y_{v2} \ \dots \ y_{vM}] \text{ and } \mathbf{a}^T = [a_{k0} \ a_{k1} \ \dots \ a_{k5}].$$

In our implementation of the GMDH algorithm the above set of linear equations (4) is solved by using Gauss elimination method with forward elimination, back substitution, and pivoting [15].

B. Model selection

The total number of possible GMDH models is increasing exponentially by increasing the number of layers. To make the search feasible, a beam search is typically used in the GMDH algorithm. Keeping the algorithm tractable by such a constraint causes it to perform sub-optimally, but often a satisfactory suboptimal solution suffices. The model selection can be formulated in the following way. Given a validation data set

$$\mathcal{D}_v = \{(\mathbf{x}_{vi}, y_{vi})\}_{i=1}^N \quad (5)$$

of data vectors $\mathbf{x}_{vi} = (x_{vi_1}, x_{vi_2}, \dots, x_{vi_K})$, $x_{vi} \in \mathcal{R}^K$, and the corresponding output values $y_{vi} \in \mathcal{R}$, evaluate the derived model $p(\mathbf{x})$ by using the performance measure $E = E(p(\mathbf{x}))$ and align it accordingly. A maximum of P_λ best qualified models are selected and retained at each layer. In this way the models on the next layer combine only the best models from the previous layers.

Single Parameter Performance Measures: Different single parameter performance measures [13] can be used when selecting the models with best prediction. A GMDH algorithm typically selects the model based on the least squares estimate e.g. Root Mean Squared Error (RMSE)

$$E_{rms} = \sqrt{\frac{1}{N} \sum_{i=1}^N (p_{vi} - y_{vi})^2}, \quad (6)$$

where y_{vi} denotes the i^{th} instance of dependent variable from the validation data set and p_{vi} is its approximation calculated by the corresponding polynomial model using the validation data set (5). While the mean squared error magnifies the effect of outliers, the Relative Absolute Error (RAE) measure

$$E_{ra} = \sum_{i=1}^N |p_{vi} - y_{vi}| / \sum_{i=1}^N |y_{vi} - \bar{y}_v|, \quad (7)$$

treats all sizes of error evenly according to their magnitude, where $\bar{y}_v = \frac{1}{N} \sum_{i=1}^N y_{vi}$ denotes the mean value of the dependent variable from the validation data set.

The Root Relative Squared Error (RRSE) takes the total squared error and divides it by the squared error of the average of the actual values from the validation data set i.e.:

$$E_{rrs} = \sqrt{\sum_{i=1}^N (p_{vi} - y_{vi})^2 / \sum_{i=1}^N (y_{vi} - \bar{y}_v)^2}. \quad (8)$$

The RRSE measure is equivalent to RMSE since it can be obtained by multiplying the RMSE by a constant i.e.

$E_{rrs} = E_{rms} \cdot \sqrt{N / \sum_{i=1}^N (y_{vi} - \bar{y}_v)^2}$. The RRSE shows how close the model approximates the real system regardless of the dynamic range of the output function. The single parameter measures (6)-(8) optimize only the corresponding error performances taking no care of the complexity, i.e. the Execution Time (ET) of the model. This can lead to complex models with low approximation error having unacceptably long ET.

MDL Measure: Minimum Description Length (MDL) [14] is a well known principle which provides a tradeoff between the accuracy and the complexity of the model. According to [16], the MDL for linear polynomial regression consists of two terms:

$$MDL = 0.5 \cdot N \cdot \log(E_{rms}^2) + 0.5 \cdot k \cdot \log(N), \quad (9)$$

where N denotes the number of observations (validation data vectors) and k is the number of parameters (coefficients) in the model. The first term in (9) can be interpreted as the number of bits necessary to encode the observations while the second term can be understood as the number of bits necessary to encode the parameters of the model. The model achieving the lowest MDL is encoding the observations the most efficiently but the most efficient encoding does not always lead to the best solution to a particular modeling problem. Next we introduce a simple Compound-squared-relative-Error (CE) measure of model performances, which forces the algorithm to build more appropriate models for RT execution than the MDL and the single parameter measures.

CE Measure: In RT measurements the calculation procedures must end within the specified time intervals. In cases when the computing power of the embedded system does not guarantee the execution of the procedures in RT it may be necessary to construct simplified surrogates with as low degradation of the approximation accuracy as possible. Here we propose a simple and intuitive two-parameter Compound squared relative Error (CE) measure for model selection i.e.

$$E_{CE} = c_w \cdot (E_{rrs} / E_{rrs0})^2 + (1 - c_w) \cdot (T_{exe} / T_{exe0})^2, \quad (10)$$

where T_{exe} denotes the ET of the model, E_{rrs0} and T_{exe0}

III. TRAINING AND VALIDATION SAMPLES OF THERMODYNAMIC PROPERTIES OF NATURAL GAS

represent the corresponding thresholds for the RRSE and ET, while c_w ($0 \leq c_w \leq 1$) denotes the weighting coefficient. The CE measure (10) consists of two normalized error terms weighing up the RRSE and the ET (complexity) of the model. We employed the RRSE instead of RMSE in (10) since the relative error is more frequently used for the characterization of the measurement system accuracy than the absolute error. The use of RRSE instead of RMSE in (10) has the same effect on model selection since it can be concluded from (6) and (8) that: $E_{rrs}/E_{rrs0} = E_{rms}/E_{rms0}$. Normalization by the corresponding thresholds (E_{rrs0} and T_{exe0}) makes the approximation error and the ET (complexity) of the model commensurable since they become dimensionless. The weighting coefficient specifies the contribution of each term. For $c_w=1$ the CE measure (10) reduces to RRSE term only and for $c_w=0$ to ET term only.

The GMDH algorithm is greedy and there is no guarantee that the satisfactory model is going to be found with respect to both constraints regardless of the applied performance measure. Unlike MDL and single parameter measures the CE measure controls the way the selected models approach to RRSE and ET thresholds and thus increases the probability to discover the satisfactory model. The weighting coefficients larger than 0.5 are enhancing the constraint on RRSE while relaxing the control of the ET. The coefficients lower than 0.5 have opposite effect. The optimal value of the weighting coefficient for a problem is not known beforehand and it has to be found experimentally. For a suboptimal coefficient, the algorithm may fail to find a model that conforms to given constraints.

Execution Time of the Model: In order to be applicable to RT measurement systems the surrogate model of the complex calculation procedure must satisfy the requirements regarding the accuracy and the ET, the most critical parameters in RT measurements:

$$(E_{rms} \leq E_{rms0} \text{ Or } E_{rrs} \leq E_{rrs0}) \text{ And } T_{exe} \leq T_{exe0}. \quad (11)$$

The ET of the polynomial model can be estimated in the following way:

$$T_{exe} \approx N_x \cdot (N_{add} \cdot T_{add} + N_{mul} \cdot T_{mul}), \quad (12)$$

where N_x denotes the total number of basic second-order two-dimensional polynomials (3) in the model to be calculated, N_{add} and N_{mul} is the corresponding total number of FP additions and FP multiplications in the basic polynomial, while T_{add} and T_{mul} denote the average execution time of software routines implementing the FP addition and the FP multiplication, respectively. If we rewrite the basic polynomial (3), its calculation is reduced to $N_{add}=5$ FP additions and $N_{mul}=5$ FP multiplications i.e.

$$p = a_0 + x_1(a_1 + x_1a_3 + a_5x_2) + x_2(a_2 + a_4x_2). \quad (13)$$

A necessary precondition for accurate flow measurements of natural gas is the compensation of the adiabatic expansion effects based on the precise calculation of natural gas properties. The procedures for the calculation of thermodynamic properties of a real gas [17]-[19] need substantial computing power in order to be repeatedly executed in RT. It may be necessary to embed into the Flow-Computer (FC) hardware a high speed MPU or math coprocessor or both in order to ensure completion of the calculations in RT. That would increase both the power consumption and the price of a FC.

The flow measurement standards [20][21] allow the use of properties of an ideal instead of a real gas only in low pressure applications. At higher pressures it may cause a significant error in flow-rate measurements, especially at low temperatures and at high differential pressures [17]. In order to preserve as high accuracy as possible and to ensure the completion of the calculations in RT, we aim to build a low complexity GMDH surrogate of complex original procedures optimized with respect to both the approximation error and the execution time. The approach will be verified on the models of natural gas thermodynamic properties. Particularly we will emphasize the effect of the Joule-Thomson (JT) expansion on the accuracy of measurements of the flow-rate and demonstrate its compensation by the corresponding GMDH polynomial model tailored for RT execution.

The measurement of the fluid flow-rate by means of differential pressure devices is specified by the corresponding measurement standards [20][21]. The calculation of pressure-, temperature- and composition-dependent properties of natural gas for industrial measurements is detailed in [22]-[24]. The procedure for the calculation of compression factor and density of a natural gas mixture using AGA-8 detail characterization method [22][23] is based on 23 input parameters specified in Appendix I. The compression factor, density, etc. can be also calculated from physical properties of a natural gas using SGERG-88 equation [24] where the corresponding iterative calculation is based on 6 input parameters detailed in Appendix II. Compared to [23], the procedure in [24] is considerably faster but, unfortunately, there is no procedure available in literature for the calculation of thermodynamic properties of a natural gas, based on parameters given in Appendix II.

Hence, we aim to map the complex 23-dimensional models [19] into the corresponding 6-dimensional space [24]. To accomplish this, the corresponding 23-dimensional training and validation data vectors are randomly generated, by satisfying the range limits given in Appendix I, and the corresponding thermodynamic properties (JT coefficient, molar heat capacity, and isentropic exponent) [19] are calculated. The 23-dimensional data sets are then mapped into the 6-dimensional space by satisfying the operating ranges given in Appendix II, where the values of only two variables (HS and d) need to be calculated in accordance with [25],

while the remaining four variables are identical in both spaces. Finally, the following 6-dimensional random data sets are obtained:

$$\mathcal{D}_t = \{p_{ti}, T_{ti}, HS_{ti}, d_{ti}, x_{CO2ti}, x_{H2ti}, y_{ti}\}_{i=0}^{M-1}, \quad (14)$$

and

$$\mathcal{D}_v = \{p_{vj}, T_{vj}, HS_{vj}, d_{vj}, x_{CO2vj}, x_{H2vj}, y_{vj}\}_{j=0}^{N-1}, \quad (15)$$

where the subscript ‘*t*’ stands for the training and the subscript ‘*v*’ for the validation samples while $M=20000$ and $N=20000$ denote the total number of training and validation samples, respectively. Each sample in (14) and (15) consists of 6 input variables (Appendix II) and the corresponding relation (thermodynamic property) y . The polynomial models of the relation y are generated from data sets (14) and (15) using the GMDH algorithm and the corresponding error criterion (6)-(10).

IV. EXPERIMENTAL RESULTS

A. Modeling the Joule-Thomson Coefficient of Natural Gas

The GMDH polynomial models are tailored for our FC based on LCP MPU (Z84C15, 8-bit/16-MHz) with embedded FP subroutines for single precision addition and multiplication having the average execution time approximately equal to 50 μ s and 150 μ s, respectively. The models are learned offline by using training (14) and validation (15) samples. Different results have been obtained by varying the error thresholds (11), the error criteria (6)-(10) and the maximum total number of the best models that can be retained per each layer.

RRSE, RMSE and RAE Performance Measures: Fig. 2 illustrates how the RRSE criterion (8) forces the algorithm to decrease the RRSE error of the ‘best’ JT coefficient polynomial models when increasing the number of layers, but allows the ET of the models to grow uncontrolled. Fig. 2 shows the normalized ET (dotted lines) and RRSE (solid lines) in a logarithmic scale for the best 10 models per each layer. From Fig. 2 it can be seen that all ET lines exceeded the threshold (normalized error = 1) before any of the RRSE lines goes below it and the algorithm fails to find the satisfactory model. The identical results are obtained when using RMSE (6) instead of RRSE (8) and very similar algorithm behavior for RAE (7) measure.

MDL Performance Measure: Fig. 3 illustrates how the MDL criterion (9) forces the algorithm to build the models with minimum description length but fails to find any model satisfying both conditions (RRSE and ET thresholds).

CE Performance Measure: Fig. 4 demonstrates how the CE performance measure (10) forces the algorithm to search for the satisfactory model of the JT coefficient by decreasing the RRSE while controlling the increase of the ET. From Fig. 4 it can be seen that the RRSE of at least one model (solid lines) drops below the threshold before any of the ET (dotted lines) exceeds it and the algorithm finds the first satisfactory model

at layer 13. By increasing the number of layers the RRSE of the models is further decreased but the ET slightly increases and at layer 22 exceeds the threshold. The results in Fig. 4 have been obtained for the weighting coefficient $c_w=0.6$. Similar patterns have been also obtained for different training and validation data sets.

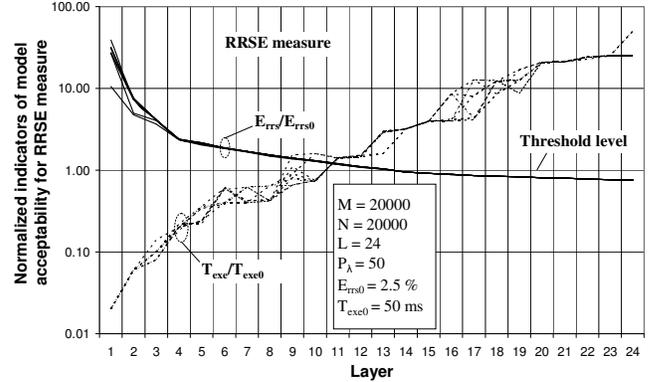


Fig. 2. Illustration of the GMDH algorithm with RRSE criterion (8) failing to find a satisfactory model for the JT coefficient. The RRSE (solid lines) and the ET (dotted lines) are normalized to their respective thresholds.

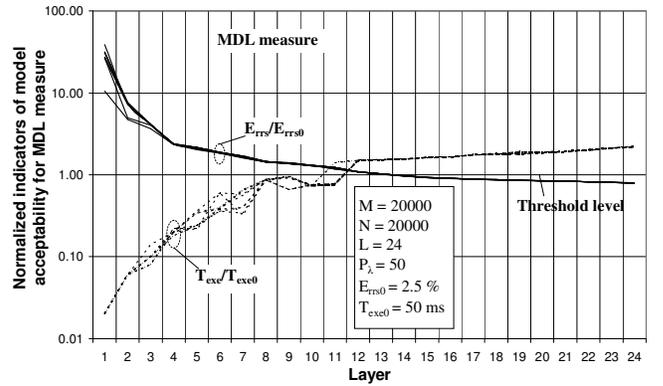


Fig. 3. Illustration of the GMDH algorithm with MDL criterion (9) failing to find a satisfactory model for the JT coefficient. The RRSE (solid lines) and the ET (dotted lines) are normalized to their respective thresholds.

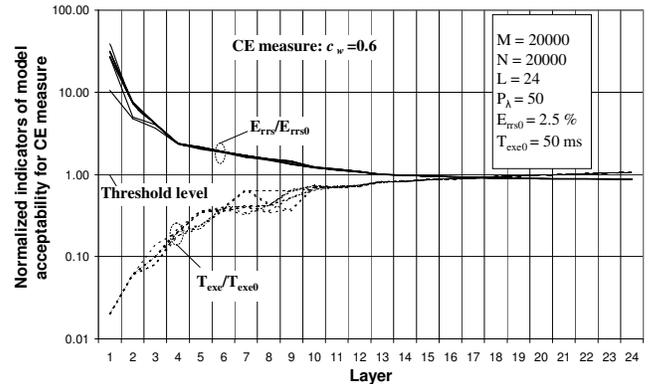


Fig. 4. Illustration of the GMDH algorithm approaching the error thresholds (11) for JT coefficient by using the CE criterion (10). The RRSE (solid lines) and the ET (dotted lines) are normalized to their respective thresholds.

Fig. 5 illustrates the effect of the weighting coefficient c_w in

CE measure on the RRSE and ET, where the average normalized RRSE and ET are shown for the best 10 models obtained at layer 15. From Fig. 5 it can be seen that, in this particular case, the RRSE is decreasing and the ET is increasing with the increase of the weighting coefficient and in the interval $0.5 \leq c_w \leq 0.7$ both parameters are below the given thresholds where the satisfactory models have been found. The best models with almost identical performances have been found for $c_w=0.5$ and $c_w=0.6$.

GMDH Model: The computer generated graph of the best model of a JT coefficient, obtained at layer 13 by applying

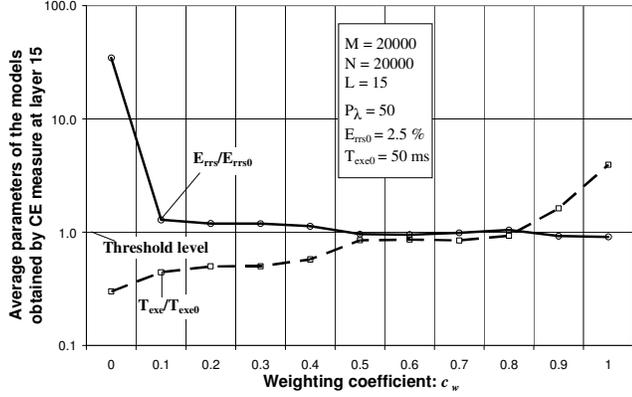


Fig. 5. Illustration of the effect of the weighting coefficient in the CE performance measure on the RRSE and ET of the models. The average results for the 10 best models obtained at layer 15 are shown.

minimum CE criterion (10) with $c_w=0.5$, which satisfies the RRSE and ET requirements, is shown in Fig. 6 and the corresponding polynomial in recursive form is given by

$$y = P_{23}(P_{19}(P_{18}(P_{17}(P_{16}(P_{15}(P_{14}(P_{11}(P_9(P_8(P_4(P_2(P_0(x_1, x_3), P_1(x_0, x_5)), P_3(x_2, x_5))), P_7(P_6(P_5(x_4, x_5), x_3), P_3(x_2, x_5))), x_2), P_{10}(x_1, x_4)), P_{13}(P_{12}(x_3, x_5), P_5(x_4, x_5))), P_3(x_2, x_5))), P_{11}(P_9(P_8(P_4(P_2(P_0(x_1, x_3), P_1(x_0, x_5)), P_3(x_2, x_5))), P_7(P_6(P_5(x_4, x_5), x_3), P_3(x_2, x_5))), x_2), P_{10}(x_1, x_4))), P_1(x_0, x_5)), x_1), x_2), P_{22}(P_{20}(x_0, x_3), P_{21}(x_1, x_2)))) .(16)$$

Due to space limitations the regression coefficients of the polynomials (P_0, \dots, P_{23}) from (16) are not shown. The threshold levels (5) were set to $E_{rrs0}=2.5\%$ and $T_{exe0}=50$ ms and the maximum number of qualified polynomial models per each layer, $\lambda=1, \dots, 13$, to $P_\lambda=50$. The input parameters (variables) in Fig. 6 are denoted by 'X0' to 'X5' and are described in Appendix II. The layers in Fig. 6 are denoted by 'L00' to 'L13' and the polynomials by 'P_m(n)' where 'm' indicates the order in which the corresponding polynomial is calculated and 'n' denotes the total number of the basic polynomial (13) computations when the polynomial function is calculated recursively, e.g. (16).

The calculated RRSE (2.498 %) and the ET (41 ms) are both below the threshold levels. Fig. 6 illustrates how the algorithm, at upper layers, almost regularly combines the high

order model with the corresponding low order one trying to improve the accuracy of the resulting model as much as possible with the lowest possible increase of its complexity. The estimated ET of the polynomial model (41 ms) corresponds to the computation of the recursive relation 'y' given in (16), where 41 basic polynomial calculations need to be executed. Note that the ET can be also estimated by the straightforward calculation of the polynomial tree (24 polynomials; P_0 to P_{23}) instead of the recursive relation (41 polynomials) but for the simplicity of the implementation we estimate here the execution time of the recursive equation. The

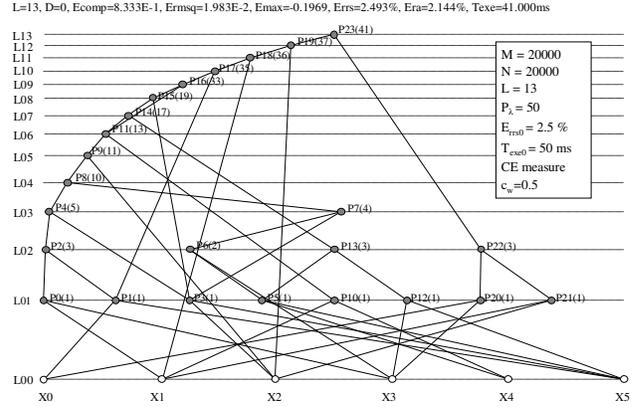


Fig. 6. The GMDH polynomial model of a JT coefficient, obtained at layer 13 using the CE criterion (10) with $c_w=0.5$.

calculation of the polynomial model needs only the software routines for FP addition and FP multiplication that can be easily incorporated into the FC software. For the calculation of the original JT coefficient model [19] a complete FP package is necessary.

When simulating the execution in the FC with the integrated FP software package instead of the math coprocessor, the average calculation time of the original JT calculation procedure [19] takes roughly 1500 ms on average. The average execution time of the recursive equation of the best polynomial model (16) is more than 36 times shorter than the original procedure what makes the polynomial model suitable for implementation in RT measurements particularly in LCP systems. It is obvious that the same GMDH approach could be used for high computing power systems, only the execution times and the corresponding thresholds must be properly adjusted.

B. JT Coefficient Calculation Example

The accuracy and the precision of the JT coefficient model from Fig. 6 was tested on 10 randomly generated validation data sets each consisting of 20000 samples. The standard deviation of RMSE equals approximately 1% of the average value for any given validation set and we may conclude that the initial model is well represented by the derived surrogate. Fig. 7 shows an example of relative error $(JT_{gmdh}-JT_{iso})/JT_{iso}$ of the JT coefficient calculated by the best GMDH polynomial model (JT_{gmdh}) obtained at layer 13 (16) and the complex accurate procedure (JT_{iso}) detailed in [19]. The error is given for a JT coefficient of a natural gas mixture specified in [19]

(“Gas3” from Table G.1) which exhibits the worst GMDH polynomial approximation of all six mixtures given for validation purposes. From Fig. 7 it can be seen that the relative absolute error never exceeds 4.00 % for temperatures above 273 K. For temperatures below 273 K the relative absolute error increases especially at higher pressures and goes up to 7.0% at 12 MPa and 263 K. Useful surrogates are also derived for the molar heat capacity and the isentropic exponent of a natural gas by applying the CE measure but, due to space limitations, only the results obtained for the JT coefficient model are included in the paper.

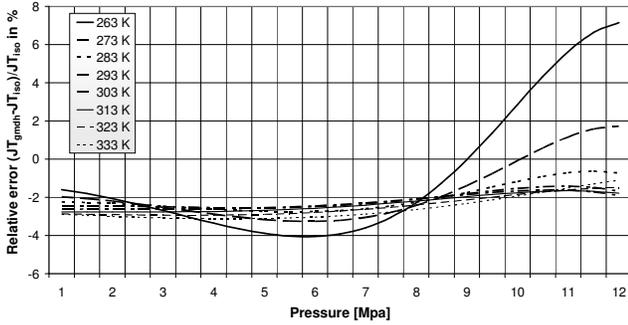


Fig. 7. Illustration of a relative error of the JT coefficient calculated by the corresponding polynomial model (16) instead of the complex accurate procedure detailed in [19].

C. Compensation for the JT Effect in the Flow-Rate Measurements by the GMDH Polynomial Model

The temperature of a gas flowing through a constriction (orifice) is changing and thus affects the flow-rate measurement accuracy. The effect is known as JT expansion and can be compensated by the JT coefficient. In order to illustrate the JT effect and its compensation we simulated the measurement of the flow-rate by means of square-edged orifice plate with corner taps [20], with the orifice diameter of 120 mm, the pipe diameter of 200 mm, the differential pressure of 200 kPa, the dynamic viscosity of 0.000012 Pa·s and by assuming the temperature sensor locating downstream of the orifice. Again, the natural gas mixture ‘Gas 3’ is selected from Table G.1 in [19]. Fig. 8 illustrates the results

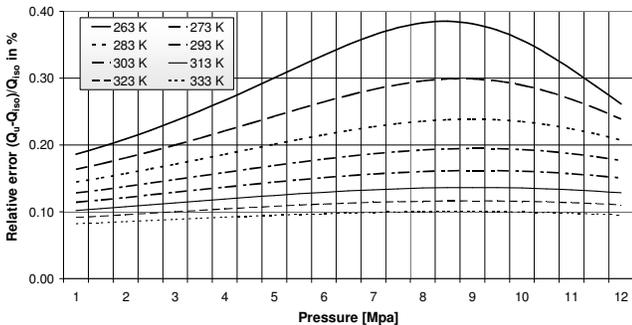


Fig. 8. Illustration of a relative error in a measurement of a natural gas flow-rate when ignoring the temperature drop due to the JT expansion effect.

of the simulation of the error $(Q_u - Q_{iso})/Q_{iso}$ in the measurement of the natural gas flow-rate when ignoring the temperature drop due to the JT expansion effect (Q_u), instead of its precise

correction (Q_{iso}) in accordance with [19]. The relative flow-rate error increases when decreasing the temperature and almost reaches 0.40 % at 263 K and 8.5 MPa.

Fig. 9 illustrates the results of the simulation of relative error $(Q_{gmdh} - Q_{iso})/Q_{iso}$ in the flow-rate (Q_{gmdh}) obtained when using the GMDH polynomial model (16) and the flow-rate (Q_{iso}) obtained when using accurate but complex procedure detailed in [19] for the calculation of the JT coefficient and the corresponding compensation of the temperature drop [17]. From Fig. 8 and 9 it can be seen that the compensation for the effect of temperature change, by using the JT coefficient

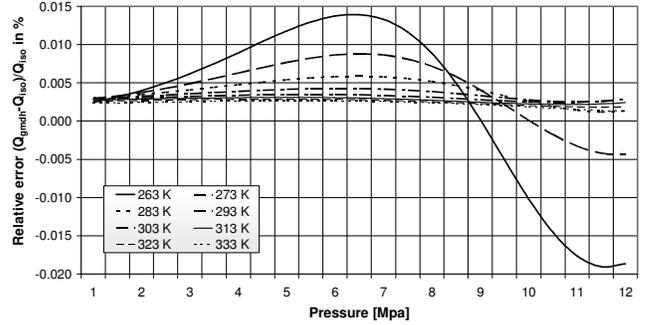


Fig. 9. Illustration of a relative error in a measurement of a natural gas flow-rate after using the corresponding polynomial model of the JT coefficient (16) for the correction of a temperature drop instead of the complex accurate procedure detailed in [17].

model (16), decreases the relative error of the uncorrected flow-rate at least by the order of magnitude whilst the maximum relative error is decreased more than 20 times.

The CE measure proved to be very useful in tailoring a suitable multilayer polynomial model of a complex calculation procedure for the application in RT embedded systems since it optimizes concurrently the accuracy and the complexity of the model. We have generated different models of the JT coefficient and other natural gas properties by using different training and validation data sets and by varying the thresholds, the maximum total number of models per layer, etc. and we found out that in the ET-critical applications the CE measure (10) generally leads to the most favorable solution.

V. COMPARISON TO OTHER METHODS

In this section experiments are described that include SVR formulations [26] and multilayered perceptrons in an attempt to compare them to the GMDH that uses different selection measures in terms of approximation error with an explicit ET constraint. What is needed first is to estimate ET for each family of models we consider, to make them comparable; this will be done in subsection A. ETs of the GMDH models are estimated according to (12). In the subsection B, each family of regression algorithms will be addressed separately, presenting experimental setup followed by results, comparisons and observations.

A. Estimation of the Computational Cost of the Models

In this work we restricted our attention to the

implementations of the models in LCP microcomputers having no math coprocessor. As we mentioned in Section IV the execution time for single precision FP addition in our FC is $T_{add}=50 \mu\text{s}$ and for FP multiplication $T_{mul}=150 \mu\text{s}$. The ET of the GMDH model can be roughly estimated by (12).

The embedding of the multilayer perceptron (MLP) with sigmoid activation functions $(1+e^{-x})^{-1}$ into our FC implies the implementation of the exponential function (e^x) in FP software. According to [27] the exponential function (e^x) can be approximated by the 7th order polynomial $e^x \approx \sum_{k=0}^7 a_k x^k$ ($|x| \leq 1$) with the approximation error $\varepsilon = 2 \cdot 10^{-7}$. Since the above equation approximates the e^x in the range $|x| \leq 1$ each result must be properly scaled to fit the range of x for each real exponent falling outside this interval what increases the computational time. To complete the calculation of the sigmoid function one FP division is needed additionally. Hence, the ET of the calculation procedure implemented in our FC software for the simple fully connected MLP with N inputs, 1 output neuron with linear activation function and 1 hidden layer consisting of M neurons with sigmoid activation functions can be roughly estimated by:

$$T_{mlp} \approx M(N+1)(T_{add} + T_{mul}) + M(T_{exp} + T_{add} + T_{div}), \quad (17)$$

where T_{exp} , and T_{div} denote the execution time for the FP exponential function and FP division, respectively. The average ETs for the above FP operations in our FC are: $T_{exp}=3470 \mu\text{s}$ and $T_{div}=430 \mu\text{s}$.

Support vector machines model [26] can be written in the following form: $f(\mathbf{x}) = \sum_{i=1}^M \alpha_i \cdot k(\mathbf{x}_i, \mathbf{x}) + \alpha_0$, where $k(\mathbf{x}_i, \mathbf{x})$ is the kernel, α_i and α_0 are the coefficients and M is the total number of support vectors (SV). The polynomial kernel [28] has the following form $k(\mathbf{x}_1, \mathbf{x}_2) = (\gamma \cdot \langle \mathbf{x}_1, \mathbf{x}_2 \rangle + \gamma_0)^K$, where γ and γ_0 are the constants and K is a positive integer denoting the degree of the polynomial. The radial basis function (RBF) kernel [28] has the following form: $k(\mathbf{x}_1, \mathbf{x}_2) = e^{-\gamma \|\mathbf{x}_1 - \mathbf{x}_2\|^2}$, where γ is a coefficient. The ET of the calculation procedure implemented in our FC software for the SVR with polynomial kernel and with MN -dimensional SVs is estimated by:

$$T_{svm_p} \approx M((N+2)(T_{add} + T_{mul}) + T_{iexp}), \quad (18)$$

where T_{iexp} denotes the ET for positive integer exponentiation, which depends on the binary representation of the exponent. The integer exponentiation can be easily implemented in software by saving the intermediate results. Thus the ET of x^K for the exponent $K=2$ (x^2) is $T_{iexp} \approx T_{mul}$ for $K=3$ ($x^2 \cdot x$) and $K=4$ ($(x^2)^2$) $T_{iexp} \approx 2 \cdot T_{mul}$, for $K=5$ ($(x^2)^2 \cdot x$), $K=6$ ($(x^2)^2 \cdot x^2$) and $K=8$ ($((x^2)^2)^2$) $T_{iexp} \approx 3 \cdot T_{mul}$, for $K=7$ ($(x^2)^2 \cdot x^2 \cdot x$) $T_{iexp} \approx 4 \cdot T_{mul}$, etc. The corresponding ET for the RBF kernel is estimated by:

$$T_{svm_r} \approx M(N(T_{sub} + T_{mul} + T_{add}) + T_{exp} + 2T_{mul} + T_{add}) \quad (19)$$

where $T_{sub}=50 \mu\text{s}$ denotes the average ET for FP subtraction.

From (17)-(19) one can calculate the maximum allowed number of neurons in a hidden layer and SVs for the predefined ET of the model. As can be seen from Fig. 6, the GMDH model consists of 24 polynomial nodes and the execution time (12), when calculating all 24 basic polynomials successively, is $T_{gmdh} \approx 24(5T_{add} + 5T_{mul}) = 24 \text{ ms}$. Now we can use the equations (17-19) to estimate the corresponding number of neurons and SVs in MLP and SVR models having similar computational complexity (ET $\approx 24 \text{ ms}$). In case of MLP with sigmoid activation function the total number of neurons in a hidden layer, calculated by (17), is $M \approx 4.486$ and we fixed the number of neurons to $M=4$ with $T_{mlp} \approx 21.400 \text{ ms}$. In case of SVR the total number of SVs for RBF kernel, calculated by (19), is $M \approx 4.511$ and we fixed the number of SVs to $M=5$ with $T_{svr} \approx 26.600 \text{ ms}$, respectively. The maximum allowable number of SVs for the polynomial kernel depends on the degree of the polynomial. Table I shows the estimated and the maximum allowable number of SVs with the corresponding execution time for the polynomial degree varying in the range 1 to 10.

TABLE I

EXECUTION TIMES OF THE SVR MODEL WITH POLYNOMIAL KERNEL WITH RESPECT TO POLYNOMIAL DEGREE AND THE TOTAL NUMBER OF SELECTED SVs

Polynomial degree	Estimated number of SVs for ET=24 ms	The maximum allowed number of SVs	Execution time in ms
1	15	15	24.000
2	13.714	14	24.500
3	12.632	13	24.700
4	12.632	13	24.700
5	11.707	12	24.600
6	11.707	12	24.600
7	10.909	11	24.200
8	11.707	12	24.600
9	10.909	11	24.200
10	10.909	11	24.200

B. Experiments

GMDH with other measures: The conducted experiments have been described in Section IV. For the sake of completeness of this chapter, we will retrospect and summarize the observations once more. Changes in behavior of the GMDH using the proposed CE measure, the RRSE norm, and the MDL based measure as selection criteria have been examined and their performances evaluated and listed in the Table II below. Using our dataset, it has been shown how different criteria consider different kinds of models as building blocks for future models.

It can be noticed that the RRSE guided GMDH is characterized by the quickest convergence of error, but has absolutely no preference over models of smaller complexities. When selection is performed according to the MDL measure,

model complexities are penalized, but the algorithm still fails to find a satisfactory solution. The CE offers a variable ratio of weights for error and complexity penalties, making it suitable for controlling model complexities while increasing accuracies, as has been shown in Section IV.

Note that the MDL principle, seen as a mathematical formulation of the Occam's razor, outside the GMDH context, inherently tends to prevent overfitting by penalizing model complexity. But the GMDH has no need for an overfitting-preventing selection criterion because model performances are calculated on a validation set, different from the training set. So a different complexity-penalizing criterion may be more useful here; although similar to the MDL-measure in formulation, the CE measure has been derived heuristically with the goal to serve application-specific requirements.

Support Vector Regression Variants: The following experiments explore support vector regression formulations in a prediction-time-constrained context and effects of such constraints on model performance. Although SVMs are known to yield sparse solutions, their sparsity is often not of sufficiently low order for real-time applications, as will be shown later on our case. Learning schemes bound to boost model sparsity that we consider here are ϵ -SVR [26] with RBF kernel followed by Reduced Set (RS) postprocessing [29] and v -SVR [26] with RBF and polynomial kernels. Experiments were run in MATLAB, using LIBSVM [30] and Statistical Pattern Recognition Toolbox [28].

For all SVR variants used, experiments have been performed as follows. Each attribute has been scaled to $[0, 1]$, for all training instances. To optimize model parameters, an appropriate logarithmic/linear grid search has been performed, using 5-fold crossvalidation to evaluate performance of each parameter set. Typically, a finer search in the parameter space would follow, based on the same performance criterion. Finally, a model with the best found set of parameters has been built on the entire training set, with its performance evaluated on the independent testing set. Attributes in the testing set had beforehand been scaled by the same factors as with the training set.

As a starting point, performance of ϵ -SVR on our dataset is first examined in terms of the RRSE only, disregarding prediction time. For *off-the-shelf* ϵ -SVR usages, one can find recommendations to use some sensible prefixed ϵ depending on the estimated noise level, while optimizing only the C and kernel parameters. Indeed, in case of noisy datasets, models obtained in such manner may have generalization properties that are both good in performance and robust to changes in ϵ . However, what is specific to our dataset is that it contains no additive noise (other than noise due to quantization) and is assumed to have had captured all the nonlinearities that characterize the underlying, sampled function; in our case it is more a matter of interpolating the function value rather than generalizing from noisy observations. With that in mind, optimal values for ϵ are expected to fall close to zero (with a potentially large C), but should not be made too small in order to pave way for some harmless bias in our models.

Using RBF kernels, our search resulted in a model with

RRSE = 1.179%, with parameters $(\epsilon, C, \gamma) = (0.01, 30, 1)$, having as much as 4491 SVs - it would take 23892.12 milliseconds to calculate a single output on our platform. It is this large prediction time that makes it inapplicable, for the model simply isn't sparse enough.

To reduce the number of vectors that characterize the previously found model, effect of running RS algorithm on it has been measured, with number of vectors limited to 5, according to subsection A. The best result it offered is listed in Table II, having an unacceptable RRSE = 93.880%. Note that we performed an extreme reduction of vectors here, which may account for overly-sized approximation error. Indeed, a more intelligent strategy might be to process models that are less successful but have significantly less SVs, but for demonstration purposes this approach suffices.

The following experiments focus on reducing prediction times of SV models more directly, via learning parameters. To tweak the number of SVs in the cost-function optimization stage, rather than specifying the desired accuracy via ϵ in ϵ -SVR, using v -SVR is more appropriate. According to [26], v is an upper bound on the fraction of errors and a lower bound on the fraction of SVs, which makes a strategy of choosing v closer to zero likely to result in models with fewer support vectors.

Prediction-time-constraint translates to a maximum of five SVs in the case of RBF kernel (see subsection A.). In the case of polynomial kernel, roughly speaking, maximal number of SVs varies between eleven and fourteen for integer degree increasing from two to ten (Table I.). Performance of the best models we managed to obtain in this manner, with both RBF and polynomial kernels, are included in Table II, The best model obtained with polynomial kernel has 11 SVs.

TABLE II

COMPARISON OF THE EXECUTION TIMES AND THE ROOT RELATIVE SQUARED ERRORS OF THE MODELS OBTAINED BY USING DIFFERENT TECHNIQUES

Method	Parameters	ET in ms	%RRSE
GMDH-CE	(layerWidth, c_w) = (50, 0.5)	24.000	2.493
GMDH-RRSE	layerWidth = 50	24.000	3.495
GMDH-MDL	layerWidth = 50	24.000	3.495
RS ϵ -SVR, RBF kernel	$(\epsilon, C, \gamma, \text{reducedTo}) = (0.01, 30, 1, 5)$	26.600	93.880
v -SVR, RBF kernel	$(v, C, \gamma) = (0.0001, 100.0, 0.0625)$	26.600	34.653
v -SVR, polynomial kernel	$(v, C, \gamma, \gamma_0, K) = (0.00001, 8, 5, 1, 4)$	20.900	39.780
MLP, trained by gradient descent	learningRate = 0.001	21.400	12.139
MLP, trained by LM	learningRate = 0.001	21.400	1.536

Optimization is performed fast in case of models with few support vectors, but the downside is that performances in terms of error on the testing set are sensitive to small changes in learning and kernel parameters, still making it hard to perform a solid parameter search. Although both models obtained in this way are better than the one using RS postprocessing, probably due to the fact that selection of the

SV takes place in the optimization rather than postprocessing stage, their approximation performance is not admirable.

Multilayered Perceptrons: For the MLP experiments, MATLAB Neural Network Toolbox [31] has been used, with data preprocessing similar as with previous experiments. With the MLPs it is possible to limit model complexity *a priori*. We considered a fully connected MLP limited in size to fit the imposed time-constraint (Subsection A), consisting of four sigmoid input neurons and one linear output neuron trained by the gradient descent and with the LM algorithms, respectively. Results are shown in Table II.

Gradient descent is known to get stuck in local minima for many problems [32][33], however, MLP trained by LM managed to outperform proposed method in terms of accuracy in a complexity-restrained scenario. Recalling the way optimization is preformed with the GMDH may offer an explanation: once calculated, parameters of the GMDH model are fixed, and models get refined by building more complex ones based on precalculated ones with greedily determined parameters, while in the case of the MLP all model parameters are subject to optimization simultaneously.

VI. CONCLUSION

Modeling the complex calculation procedure by self-organizing polynomial networks, with respect to the accuracy and the execution time, offers the possibilities of their efficient customization for RT applications. The complexity of the model can be decreased extremely as well as the execution time but the accuracy of the surrogate is somewhat degraded when compared to referent physical model. This algorithmic trade-off between the accuracy and the complexity proves to be a favorable approach for LCP embedded systems since the referent physical models are sometimes too complex to be executed in RT.

The CE measure of model efficiency generally discovers more favorable GMDH surrogates with respect to the execution time and the accuracy then the single error criteria or MDL measure. It was also demonstrated how the implementation of the derived polynomial model of the JT coefficient into the FC enables the compensation for the adiabatic expansion effect in RT and considerably improves the accuracy of the flow-rate measurements.

Comparison with commonly used regression methods shows superiority of the MLP with respect to both the approximation accuracy and the computational cost. Still, a high approximation accuracy, low ET and fairly simple coding of the GMDH-CE model make it quite suitable for the implementation in LCP systems with limited resources. To increase descriptive power of the GMDH while retaining the same model complexity, backward parameter tuning methods similar to [10] in conjunction with the proposed CE complexity-penalizing measure seem to point in a promising direction.

APPENDIX I

RANGES OF INPUT PARAMETERS FOR THE CALCULATION OF THERMODYNAMIC PROPERTIES OF A NATURAL GAS IN ACCORDANCE WITH [19] AND [23]

Index	Parameter description	Range of application
0	nitrogen	$0 \leq x_{N_2} \leq 0.20$
1	carbon dioxide	$0 \leq x_{CO_2} \leq 0.20$
2	methane	$0.7 \leq x_{CH_4} \leq 1.00$
3	ethane	$0 \leq x_{C_2H_6} \leq 0.10$
4	propane	$0 \leq x_{C_3H_8} \leq 0.035$
5	n-butane	n-butane+iso-butane
6	iso-butane	$0 \leq x_{C_4H_{10}} \leq 0.015$
7	n-pentane	n-pentane+iso-pentane
8	iso-pentane	$0 \leq x_{C_5H_{12}} \leq 0.005$
9	n-hexane	$0 \leq x_{C_6H_{14}} \leq 0.001$
10	n-heptane	$0 \leq x_{C_7H_{16}} \leq 0.0005$
11	n-octane	n-octane+n-nonane+n-decane $0 \leq x_{C_8H_{18}} + x_{C_9H_{20}} + x_{C_{10}H_{22}} \leq 0.0005$
12	n-nonane	
13	n-decane	
14	hydrogen	$0 \leq x_{H_2} \leq 0.10$
15	carbon monoxide	$0 \leq x_{CO} \leq 0.03$
16	water	$0 \leq x_{H_2O} \leq 0.00015$
17	helium	$0 \leq x_{He} \leq 0.005$
18	oxygen	$0 \leq x_{O_2} \leq 0.0002$
19	hydrogen sulfide	$0 \leq x_{H_2S} \leq 0.0002$
20	argon	$0 \leq x_{Ar} \leq 0.0002$
21	Absolute pressure in MPa	$0 < p \leq 12$
22	Temperature in K	$263 \leq T \leq 368$

Total sum of mole fractions must be equal to 1.00

APPENDIX II

RANGES OF INPUT PARAMETERS FOR THE CALCULATION OF COMPRESSION FACTOR OF A NATURAL GAS IN ACCORDANCE WITH [24]

Index	Parameter description	Range of application
0	x_{CO_2} - mole fraction of carbon dioxide	$0 \leq x_{CO_2} \leq 0.20$
1	x_{H_2} - mole fraction of hydrogen	$0 \leq x_{H_2} \leq 0.10$
2	p - absolute pressure in MPa	$0 < p \leq 12$
3	T - temperature in K	$263 \leq T \leq 368$
4	d - relative density	$0.55 \leq d \leq 0.80$
5	HS - superior calorific value in MJ/m ³	$30 \leq HS \leq 45$

REFERENCES

- [1] R. Jin, W. Chen, and T. W. Simpson: "Comparative studies of metamodelling techniques under multiple modeling criteria," AIAA-2000-4801, AIAA/USAF/ NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 8th, Long Beach, CA, Sept. 6-8, 2000.
- [2] T. W. Simpson and A. J. Booker, Dipankar Ghosh, Anthony A. Giunta, Patrick N. Koch, and Ren-Jye Yang: "Approximation Methods in Multidisciplinary Analysis and Optimization: A Panel Discussion," *Approximation Methods Panel, 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis & Optimization, Atlanta, GA, September 1-4, 2002.*
- [3] S. Ferrari and R. F. Stengel, "Smooth Function Approximation Using Neural Networks," *IEEE Transactions on Neural Networks*, Vol. 16, No. 1, January 2005, 24-38.
- [4] B.M. Wilamowski, N.J. Cotton, O. Kaynak and G. Dündar, "Computing Gradient Vector and Jacobian Matrix in Arbitrarily Connected Neural Networks," *IEEE Transactions on Industrial Electronics*, Vol. 55, No. 10, October 2008, 3784-3790.

- [5] L. Gong, C.L. Liu and X.F. Zha, "Model-Based Real-Time Dynamic Power Factor Measurement in AC Resistance Spot Welding With an Embedded ANN," *IEEE Transactions on Industrial Electronics*, Vol. 54, No. 3, June 2007, 1442-1448.
- [6] S. Jung and S. su Kim, "Hardware Implementation of a Real-Time Neural Network Controller with a DSP and an FPGA for Nonlinear Systems," *IEEE Transactions on Industrial Electronics*, Vol. 54, No. 1, February 2007, 265-271.
- [7] X. Kong and A.M. Khambadkone, "Modeling of a PEM Fuel-Cell Stack for Dynamic and Steady-State Operation Using ANN-Based Submodels," *IEEE Transactions on Industrial Electronics*, Vol. 56, No. 12, December 2009, 4903-4914.
- [8] S.M Clarke, J.H. Griebisch and T.W. Simpson, "Analysis of Support Vector Regression for Approximation of Complex Engineering Analyses," *Journal of Mechanical Design*, Vol. 127, November 2005, 1077-1087.
- [9] A. G. Ivakhnenko, "Polynomial Theory of Complex Systems," *IEEE Transactions on Systems Man, and Cybernetics*, Vol. SMC-1, No.4, Oct 1971, 364-378.
- [10] N. Y. Nikolaev and H. Iba, "Learning Polynomial Feedforward Neural Networks by Genetic Programming and Backpropagation," *IEEE Transactions on Neural Networks*, Vol. 14, No. 2, March 2003, 337-350.
- [11] T. Kondo and J. Ueno, "Revised GMDH-Type Neural Network Algorithm with a Feedback Loop Identifying Sigmoid Function Neural Network", *International Journal of Innovative Computing, Information and Control*, Vol. 2, No. 5, Oct. 2006, 095-996.
- [12] M. Iwasaki, H. Takei and N. Matsui, "GMDH-Based Modeling and Feedforward Compensation for Nonlinear Friction in Table Drive Systems," *IEEE Transactions on Industrial Electronics*, Vol. 50, No. 6, December 2003, 1172-1178.
- [13] I. H. Witten and F. Eibe, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, ISBN: 9780120884070, Morgan Kaufmann Publishers, 2005.
- [14] J. Rissanen, "Modeling by shortest data description," *Automatica*, 14, 1980, 465-471.
- [15] S. C. Chapra and R. P. Canale, *Numerical Methods for Engineers*, 3rd ed., McGraw-Hill, New York, 1998.
- [16] M. F. Tenorio and W. T. Lee, "Self-organizing Network for Optimum Supervised Learning," *IEEE Transactions on Neural Networks*, Vol. 1, No. 1, March 1990, 100-110.
- [17] I. Marić, "A procedure for the calculation of the natural gas molar heat capacity, the isentropic exponent, and the Joule-Thomson coefficient," *Flow Measurement and Instrumentation*, Vol. 18/1, March 2007, 18-26.
- [18] E. W. Lemmon and K. Starling, "Speed of Sound and Related Thermodynamic properties Calculated from the AGA Report No. 8 Detail Characterization Method Using a Helmholtz Energy Formulation," *American Gas Association Conference and Biennial Exhibition*, Orlando FL, April 28, 2003.
- [19] ISO-20765-1, *Natural gas – Calculation of thermodynamic properties - Part1: Gas phase properties for transmission and distribution applications*, Ref. No. ISO-20765-1:2005, ISO; 2005.
- [20] ISO-5167-2, *Measurement of fluid flow by means of pressure differential devices inserted in circular-cross section conduits running full - Part 2: Orifice plates*, Ref. No. ISO-51671-2:2003(E), ISO; 2003.
- [21] AGA Report No. 3, *Orifice metering of natural gas and other related hydrocarbon fluids -Part1: General equations and uncertainty guidelines*, AGA and API; 2003.
- [22] K. E. Starling and J. L. Savidge, "Compressibility Factors for Natural Gas and Other Hydrocarbon Gases, American Gas Association (AGA) Transmission Measurement Committee Report No. 8, American Petroleum Institute (API) MPMS, chapter 14.2, second edition, November 1992.
- [23] ISO-12213-2, *Natural gas – Calculation of compression factor – Part 2: Calculation using molar-composition analysis*, Ref. No. ISO-12213-2:2006(E), ISO; 2006.
- [24] ISO-12213-3, *Natural gas – Calculation of compression factor – Part 3: Calculation using physical properties*, Ref. No. ISO-12213-3:2006(E), ISO; 2006.
- [25] ISO 6976, *Natural gas - Calculation of calorific values, density, relative density and Wobbe index from composition*, Ref. No. ISO-6976:1995(E), ISO; 1995.
- [26] Bernhard Schölkopf and Alexander J. Smola, *Support Vector Machines, Regularization, Optimization, and Beyond*, The MIT Press, December 2001
- [27] N. V. Kopchenova and I. A. Maron, *Computational Mathematics*. Moscow: Mir Publishers, 1975.
- [28] Franc, V. and Hlavac, V., *Statistical Pattern Recognition Toolbox for Matlab*, Prague, Czech: Center for Machine Perception, Czech Technical University, 2004
- [29] Schölkopf, P. Knirsch, C. Smola and A. Burges. Fast approximation of support vector kernel expansions, and an interpretation of clustering as approximation in feature spaces, *Mustererkennung 1998*, pp. 124–132, Springer-Verlag, Berlin, Germany, 1998.
- [30] Chih-Chung Chang and Chih-Jen Lin, *LIBSVM: a Library for Support Vector Machines*, 2001 (<http://www.csie.ntu.edu.tw/~cjlin/libsvm>)
- [31] H. Demuth, M. Beale, M. Hagan, *Neural Network Toolbox™ 6 User's Guide*, The MathWorks Inc. September 2009
- [32] S. S. Haykin: *Neural Networks and Learning Machines*, Prentice Hall; 3rd edition, November 28, 2008
- [33] B.M. Wilamowski, "Architectures and Learning Algorithms," *IEEE Industrial Electronics Magazine*, Vol. 3, No. 4, December 2009, 56-63.



Ivan Maric received the B.S., M.S. and Ph.D. in electrical engineering from the Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia, in 1974, 1981 and 1989, respectively.

From 1975 to 1979 he was with "Nikola Tesla" company, Zagreb, where he was engaged in the design and engineering of telecommunication systems. In 1979 he joined the Ruder Bošković Institute (RBI), Zagreb, where he has been involved in various research projects in the field of process instrumentation, measurement systems and automation. Currently, he holds the position of Senior Scientist for Instrumentation and Measurement Systems in the Division of Electronics of the RBI and leads the research project "Computational Intelligence Methods in Measurement Systems". His research interests include, computational intelligence, numerical algorithms, embedded and distributed real-time measurement and control, intelligent instrumentation and fluid flow measurement.



Ivan Ivek received his B.S. in electrical engineering from the Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia in 2007.

He joined the Ruder Boškovic Institute in 2008, where his current position is research assistant for the Computational Intelligence Methods in Measurement Systems project. Currently, he is pursuing his Ph.D. in electronics at the Faculty of Electrical Engineering and Computing, University of Zagreb.