

UNIVERSITY OF LJUBLJANA

Faculty of Mechanical Engineering

**Identification of the measurement characteristic of
the critical flow Venturi nozzle mass flow meter using
radial basis function neural network models**

Seminar at the course Neural Networks

Student: Klemen Rupnik

Mentor: assist. prof. Primož Potočnik

Ljubljana, January 2011

Contents

1	Introduction	2
1.1	Critical flow Venturi nozzle	2
1.2	Measurement of gas mass flow rate using critical flow Venturi nozzle.....	3
1.3	Measurement system	5
1.4	Problem description.....	7
1.5	Goals of the seminar	7
1.6	Solution approach.....	7
1.7	Expected results.....	7
2	Radial Basis Function Network.....	9
2.1	Theoretical background	9
2.2	Radial basis function networks in Matlab	10
3	Identification of the measurement characteristic of the CFVN mass flow meter	12
3.1	Designing the training data.....	12
3.2	Partitioning of the training set	12
3.3	Preparation of data.....	13
3.4	Modelling of the measurement characteristic.....	14
3.4.1	Linear regression model	14
3.4.2	Radial basis function network model NN-q	15
3.4.3	Radial basis function network model NN-C	16
3.5	Multifold cross-validation based on random partitioning of the training set	16
3.5.1	Mean absolute percentage error.....	17
3.5.2	Estimation and validation errors.....	17
3.6	Optimization of parameters of RBFN model	18
3.7	Algorithm for determination of models and selection of optimal RBFN models ...	18
4	Results.....	20
4.1	Models' predictions of the mass flow rate	20
4.2	Performances of models	22
4.3	Response of RBFN models at large values of spread.....	25
5	Conclusions	28
6	References	29

1 Introduction

1.1 Critical flow Venturi nozzle

A critical flow Venturi nozzle (CFVN) is a convergent-divergent restriction inserted into a system and is used to determine the mass flow rate of a gas flowing through a system. In the critical flow Venturi nozzle the gas flow accelerates to the critical velocity at the throat (local sonic velocity) due to geometrical configuration and conditions of use. Throat is the section of minimal diameter of a Venturi nozzle. At the critical velocity, the mass flow rate of the gas is the maximum possible for the existing upstream conditions. A drawing of a toroidal-throat Venturi nozzle is shown in Figure 1.

For a critical flow Venturi nozzle the only measurements required are the gas pressure and the gas temperature (or density) upstream of the Venturi nozzle, since the throat conditions can be calculated from the thermodynamic considerations. An important consideration is that the flow through the Venturi nozzle is independent of the downstream pressure, p_2 , if the ratio of this pressure to inlet stagnation pressure, p_0 , is smaller than critical:^{[1],[2]}

$$\frac{p_2}{p_0} \leq \left(\frac{p_2}{p_0} \right)_{cr} \quad (1)$$

If this is true, the Venturi nozzle can be used for critical flow measurement. For detailed description of determination of the critical pressure ratio $(p_2 / p_0)_{cr}$ have a look at references [1] and [2].

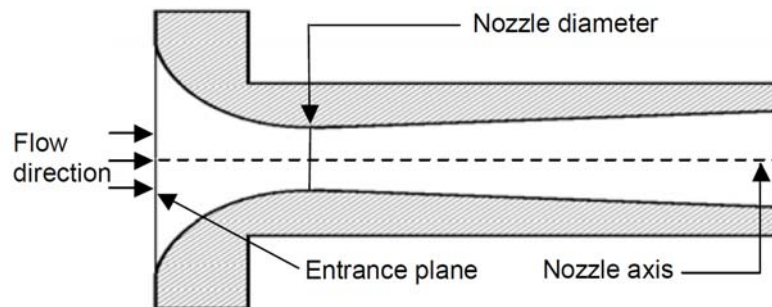


Figure 1: A toroidal-throat Venturi nozzle.^[3]

The most common applications of CFVNs are for tests, calibration and flow control.

1.2 Measurement of gas mass flow rate using critical flow Venturi nozzle

The mass flow rate under real conditions shall be computed from the equation:^[1]

$$q_m = \frac{A_{nt} C_d C_* p_0}{\sqrt{\left(\frac{R}{M}\right) T_0}}, \quad (2)$$

where p_0 and T_0 are absolute stagnation pressure and temperature, respectively, of the gas at nozzle inlet, A_{nt} is cross-sectional area of Venturi nozzle throat, C_d is discharge coefficient, C_* is critical flow function, $R = 8,314472$ J/mol/K is universal gas constant and M is molar mass of the gas.

The **inlet stagnation pressure**, p_0 , and the **inlet stagnation temperature**, T_0 , may be determined from the relationships:^[1]

$$p_0 = p_1 \left[1 + \frac{1}{2} (\kappa - 1) Ma_1^2 \right]^{\frac{\kappa}{\kappa - 1}}, \quad (3)$$

$$T_0 = T_1 \left[1 + \frac{1}{2} (\kappa - 1) Ma_1^2 \right], \quad (4)$$

where p_1 and T_1 are measured inlet pressure and temperature, respectively, Ma_1 is Mach number at the nozzle upstream static conditions – ratio of the mean axial fluid velocity to the velocity of sound at the location of the upstream pressure tapping, and κ is isentropic exponent. For an ideal gas, κ is equal to the ratio of specific heat capacities $\gamma = c_p / c_v$.

Stagnation pressure and temperature would exist in a gas in a flowing gas stream if the stream were brought to rest by an isentropic process. For very small inlet Mach numbers it is possible to approximate stagnation pressure, p_0 , and temperature, T_0 , with inlet pressure, p_1 , and temperature, T_1 . Approximation error is negligible at large ratio of inlet pipe diameter, D , to the diameter of Venturi nozzle throat, d , since $Ma_1 \propto d^2 / D^2$.^[2]

Cross-section area of Venturi nozzle throat A_{nt} may be calculated from the formula:

$$A_{nt} = \frac{\pi d^2}{4}. \quad (5)$$

Discharge coefficient C_d is a dimensionless ratio of the actual flow rate to the ideal flow rate of non-viscous gas that would be obtained with one-dimensional isentropic flow for the same upstream stagnation conditions. This coefficient corrects for viscous and flow field curvature effects. Discharge coefficient C_d depends largely on the design of the CFVN and installation conditions.

It is a function of the throat Reynolds number and may be obtained from the following equation:

$$C_d = a - bRe_{nt}^{-n} . \quad (6)$$

Values of coefficients a , b and n depend on geometry (type) of CFVN.

Reynolds number is defined as:

$$Re = \frac{4q_m}{\pi d \mu_0} , \quad (7)$$

where μ_0 is gas dynamic viscosity.

To avoid iterative computational methods, the ideal mass flow rate $q_{m,id}$ may be used in Equation (7):^[2]

$$q_{m,id} = \frac{A_{nt} C_{*i} P_0}{\sqrt{\left(\frac{R}{M}\right) T_0}} \quad (8)$$

where C_{*i} is the critical flow function for one-dimensional isentropic flow of a perfect gas:^[2]

$$C_{*i} = \sqrt{\gamma \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{\gamma-1}}} . \quad (9)$$

Critical flow function of a real gas, C_* , is a dimensionless function which characterizes the thermodynamic flow properties of an isentropic and one-dimensional flow between the inlet and the throat of a Venturi nozzle. It is a function of stagnation conditions and nature of the gas. An empirical equation has been developed to accurately represent the C_* values for various gases. The empirical equation takes the form:^[1]

$$C_* = \sum_i a_i \left(\frac{p_0}{p_c}\right)^{b_i} \left(\frac{T_0}{T_c}\right)^{c_i} , \quad (10)$$

where p_c and T_c are critical pressure and critical temperature, respectively, and a_i , b_i and c_i are gas-dependent coefficients. Equation (10) is applicable over the restricted temperature ranges – for more details, have a look at reference [1].

In the case when gas is **humid air**, the mass flow rate can be calculated from the relationship:^[2]

$$\frac{q_m}{q_{m,dry}} = 1 - 0,210x_v, \quad (11)$$

where $q_{m,dry}$ is calculated by Equation (2), wherein M is molar mass of dry air, and x_v is molar mass fraction of water vapour, which is defined as:^[4]

$$x_v = \varphi f_e (p, T) \frac{p_{sv}(T)}{p}, \quad (12)$$

where φ is relative humidity, f_e is enhancement factor and p_{sv} is vapour pressure at saturation.

1.3 Measurement system

The measurement system for identification of the measurement characteristic of the CFVN mass flow meter consists of:

- a pressure valve for setting gauge pressure of gas in the upstream pipe of the nozzle,
- pressure meter Druck DPI 605 that measures inlet pressure (at the entrance) of a Venturi nozzle,
- a set of 5 critical flow Venturi nozzles mounted into a cluster; only the nozzle with diameter of 0,436 mm was used,
- reference volume flow meter Sierra Instruments Cal=Trak SL-800 that measures also temperature, T , and ambient pressure, p_a ,
- personal computer with supervision program (LabView).

A scheme and a picture of the measurement system are represented in Figures 2 and 3.

Reference mass flow rate is calculated by the equation:

$$q_m = \rho(p_a, T) \cdot q_{v,ref}, \quad (13)$$

where ρ is gas density, determined by ambient pressure, p_a , and temperature, T .

Gas temperature is set by room temperature in the laboratory, which is regulated by the air-conditioning system.

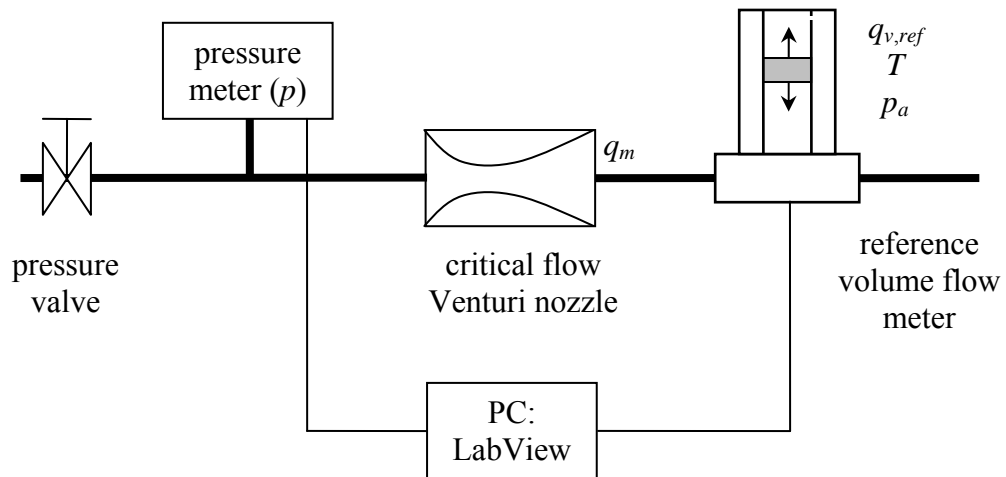


Figure 2: Scheme of the measurement system.

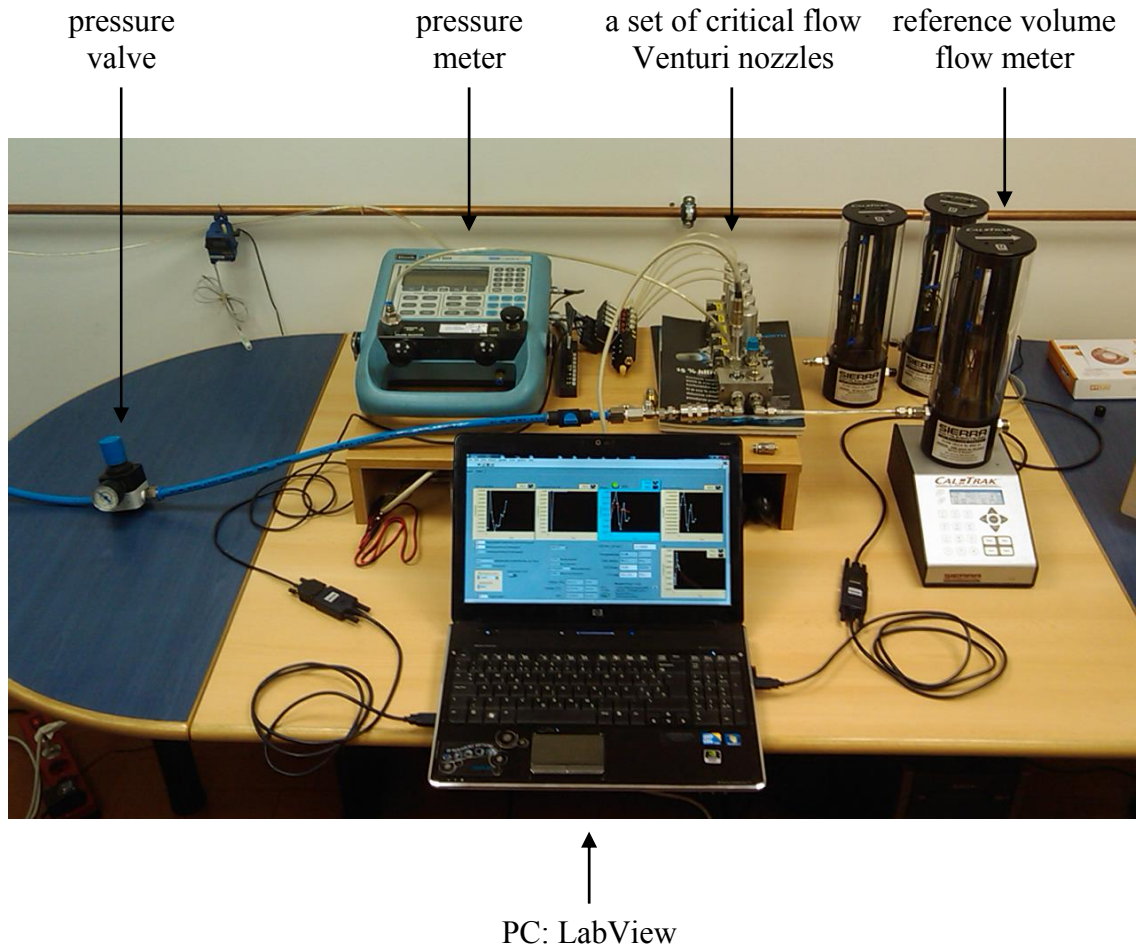


Figure 3: Picture of the measurement system.

1.4 Problem description

As introduced in prior sections, pressure and temperature influence the mass flow rate directly and indirectly, i.e. through influence on gas thermodynamic and transport properties (e.g. critical flow function or discharge coefficient). The method for identification of the relationship between pressure, temperature and mass flow rate is known and standardized – look at reference [1]. But on the other hand, modelling of this relationship by a neural network represents an alternative possibility and also a challenge.

1.5 Goals of the seminar

Goals of the seminar are:

- identification of the measurement characteristic of the CFVN mass flow meter using:
 - a linear regression model,
 - a radial basis function neural network model,
 - a radial basis function neural network model with incorporated prior knowledge,
- selection of optimal parameters of both radial basis function neural network models,
- comparison of performances, i.e. approximation and generalization abilities, of all three models.

1.6 Solution approach

We measured mass flow rates in typical ranges of pressure and temperature, i.e. approximately from 200 to 650 kPa and from 20 to 25 °C. Thus measured data is representative for the problem considered. At every set pressure and temperature 10 measurements of mass flow rate were repeated and data were saved. 45 differently set values of pressure and temperature were selected, thus we got 450 measurement points (Figure 4). On these 45 clusters, each consisting of 10 data points, approximation models were estimated and validated. As described in Section 1.5, our goal is to identify the measurement characteristic of the CFVN mass flow meter using three various models.

1.7 Expected results

In the model, described by Equation (2), it is obvious that the function $q_m = f(p, T)$ is nonlinear, even though its nonlinearity is not visible from plotted values of this function at the first sight (Figure 4). But for the CFVN measurement system with target expanded measurement uncertainty of about 0,2 % of the measured mass flow rate, it is necessary for the model to involve this nonlinearity.

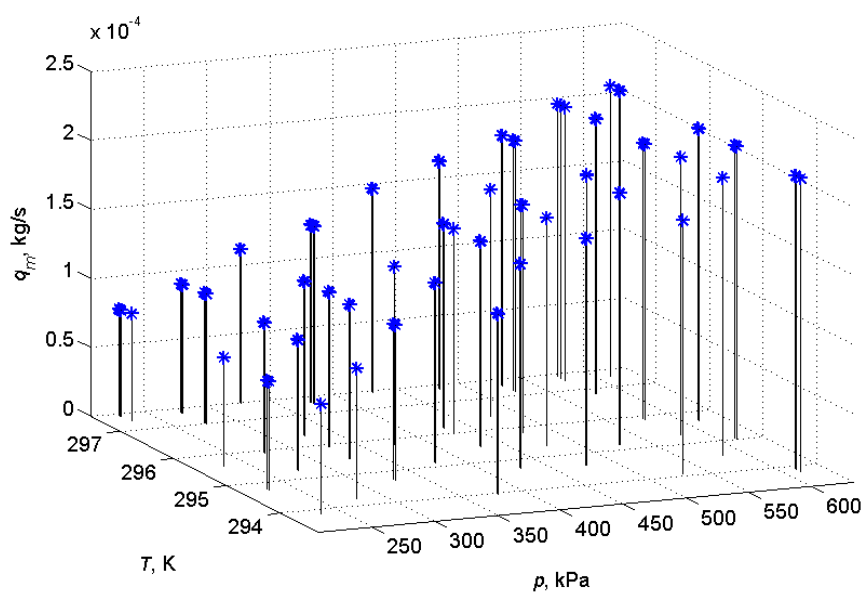


Figure 4: Measured data: 45 clusters, each containing 10 data points (blue stars).

2 Radial Basis Function Network

2.1 Theoretical background

A radial basis function network (RBFN) has two computational layers. Hidden layer consists of a set of radial basis functions and output layer implements linear summation functions.^[5] Scheme of a RBFN is represented in Figure 5.

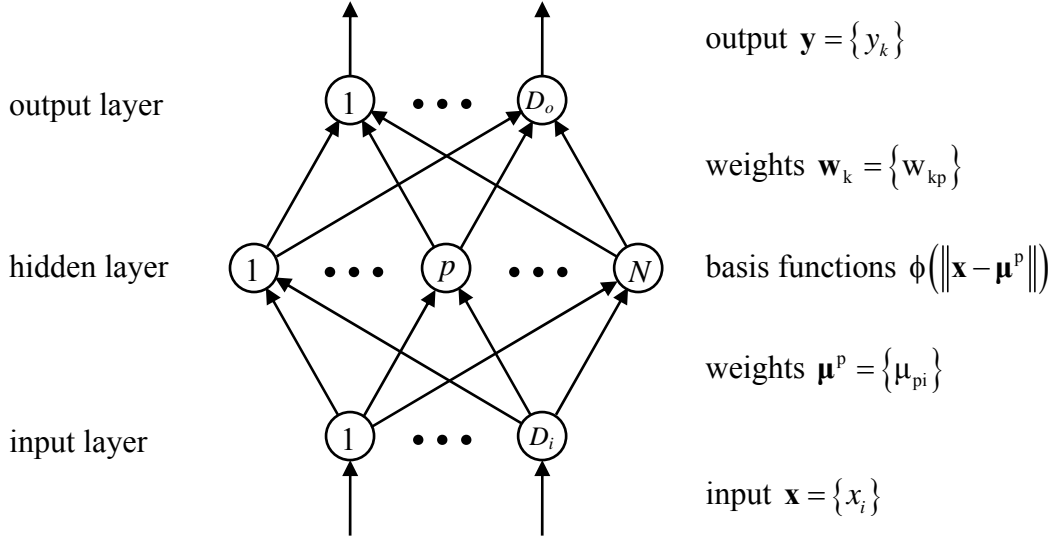


Figure 5: Scheme of a radial basis function network.

Activation of each (p -th) hidden unit is determined by distance between input vector, \mathbf{x} , and its prototype vector, $\boldsymbol{\mu}^p$. An example of a radial basis function is the Gaussian function with the following form:^[5]

$$\phi_g(\|\mathbf{x} - \boldsymbol{\mu}^p\|) = \exp\left(-\frac{\|\mathbf{x} - \boldsymbol{\mu}^p\|^2}{2\sigma^2}\right). \quad (14)$$

Gaussian and some other basis functions are localised, which means:^[5]

$$\phi(\|\mathbf{x} - \boldsymbol{\mu}^p\|) \rightarrow 0 \quad \text{as} \quad \|\mathbf{x} - \boldsymbol{\mu}^p\| \rightarrow \infty. \quad (15)$$

Output of a radial basis function network is a linear combination of basis functions.^[5]

$$y_k(\mathbf{x}) = \sum_{p=1}^N w_{kp} \phi(\|\mathbf{x} - \boldsymbol{\mu}^p\|) \quad (16)$$

or, in other words, it is a superposition of radial basis functions with their heights set by output weights \mathbf{w}_k .

The training procedure of a RBFN can be divided into two stages:

- training of hidden layer weights,
- training of output layer weights.

Hidden layer can be trained by unsupervised methods, e.g. random selection of fixed centers, orthogonal least squares or K -means clustering. The result is determination of centers and widths of radial basis functions in hidden layer neurons. In general, number N of basis functions (hidden units) should be much less than the number of training data points.

Output layer has linear activation, thus the output weights are determined analitically by solving a set of linear equations. Values of output layer weights are determined as values, at which the sum-squared output error,^[5]

$$E = \frac{1}{2} \sum_p \sum_k \left(y_k(\boldsymbol{\mu}_p) - t_k^p \right)^2, \quad (17)$$

reaches its minimum. In Equation (17) t_{kp} are target values.

It is also possible to perform supervised training of basis function parameters. Given results are generally better than those of unsupervised learning, but the procedure is computationally inefficient.

2.2 Radial basis function networks in Matlab

In Matlab, the function *newrb* creates a radial basis function network. The first layer has neurons with *radbas* transfer function (Equation (18)) and calculates its weighted inputs with *dist* and its net input with *neprod*. The second layer has neurons with *purelin* transfer function and calculates its weighted input with *dotprod* and its net inputs with *netsum*. Both layers have biases. Architecture of a RBFN is represented in Figure 6.^[7]

Initially the hidden layer has no neurons. The following steps are repeated until the network's mean square error falls below goal:^[6]

- The network is simulated.
- The input vector with the greatest error is found.
- A *radbas* neuron is added with weights equal to that vector.
- The linear output layer weights are redesigned to minimize error.

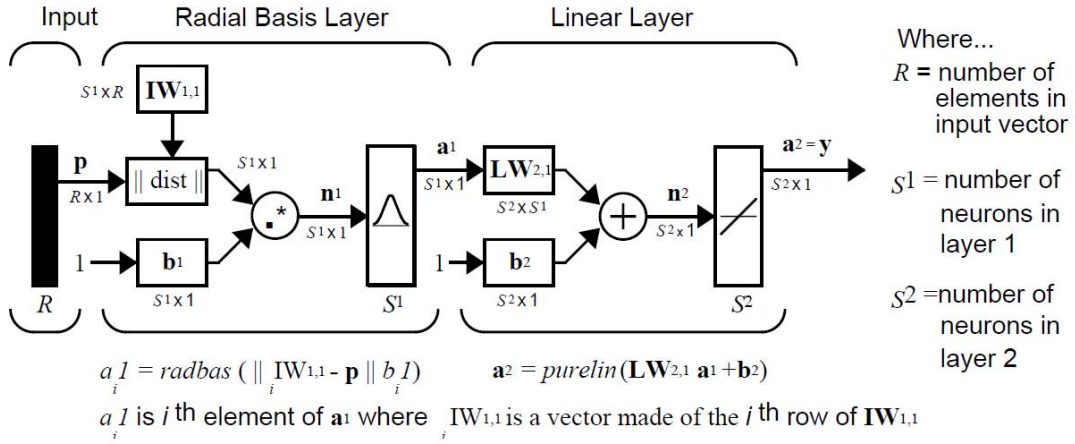


Figure 6: Architecture of radial basis function network.^[7]

One of the arguments of the function *newrb* is *spread*, which determines the width of radial basis functions. The larger is the *spread*, the smoother is the function approximation. For too large *spread* a lot of neurons are required to fit a fast-changing function. Too small *spread* means many neurons are required to fit a smooth function, and the network might not generalize well.^[6]

The net input to the *radbas* transfer function is the vector distance between its weight vector μ^p ($IW^{1,1}$ in Figure 6) and the input vector x (p in Figure 6), multiplied by the bias, b .

The transfer function for a radial basis neuron is:^[7]

$$radbas(\|x - \mu^p\| b) = \exp\left(-\left(\|x - \mu^p\| b\right)^2\right). \quad (18)$$

The bias allows the sensitivity of the the *radbas* neuron to be adjusted.

We found out that the relation between bias, b , and *spread* of the function *radbas* is:

$$b = \frac{0,8326}{spread}. \quad (19)$$

Thus standard deviation σ of the Gaussian function (Equation (14)) is related to *spread* and bias as follows:

$$\sigma = \frac{spread}{0,8326\sqrt{2}} = \frac{1}{b\sqrt{2}}. \quad (20)$$

3 Identification of the measurement characteristic of the CFVN mass flow meter

3.1 Designing the training data

As mentioned in Section 1.6, every data cluster consists of $n = 10$ measurement points. There were $N = 45$ clusters spreading as equally as possible over typical pressure and temperature ranges for the CFVN measurement system (Section 1.6). Input data, i.e. measured values of pressure and temperature, are collected into the matrix \mathbf{M}_{PT} :

$$\mathbf{M}_{PT}^T = \begin{pmatrix} p_1 & p_2 & \dots & p_n & \dots & p_{(i_c-1)n+1} & p_{(i_c-1)n+2} & \dots & p_{i_c \cdot n} & \dots & p_{(N-1)n+1} & p_{(N-1)n+2} & \dots & p_{N \cdot n} \\ T_1 & T_2 & \dots & T_n & \dots & T_{(i_c-1)n+1} & T_{(i_c-1)n+2} & \dots & T_{i_c \cdot n} & \dots & T_{(N-1)n+1} & T_{(N-1)n+2} & \dots & T_{N \cdot n} \end{pmatrix}. \quad (21)$$

$\underbrace{\hspace{10em}}_{1^{\text{st}} \text{ cluster}}$
 $\underbrace{\hspace{10em}}_{i_c\text{-th cluster}}$
 $\underbrace{\hspace{10em}}_{N\text{-th cluster}}$

Targets, i.e. measured values of the mass flow rate, are collected into the vector \mathbf{q}_m :

$$\mathbf{q}_m^T = \left(\underbrace{q_{m,1} \quad q_{m,2} \quad \dots \quad q_{m,n}}_{1^{\text{st}} \text{ cluster}} \quad \underbrace{q_{m,(i_c-1)n+1} \quad q_{m,(i_c-1)n+2} \quad \dots \quad q_{m,i_c \cdot n}}_{i_c\text{-th cluster}} \quad \underbrace{q_{m,(N-1)n+1} \quad q_{m,(N-1)n+2} \quad \dots \quad q_{m,N \cdot n}}_{N\text{-th cluster}} \right). \quad (22)$$

All measured values of pressure, temperature and mass flow-rate are further collected into the $(N \cdot n) \times 3$ matrix \mathbf{M}_{PTQ} , which represents the training data set.

3.2 Partitioning of the training set

The training data set is randomly partitioned into two subsets:

- an estimation subset that is used to build a model and
- a validation subset that is used to validate the built model.

Number of clusters in each of subsets may be selected freely. We decided to partition the training set with respect to the ratio 2 : 1, which means that estimation and validation subsets contain 2/3 and 1/3 of clusters (or 300 and 150 data points), respectively.

Our own algorithm for partitioning of the training set into two subsets executes the following operations:

- Matlab function $randperm(N)$ returns randomly permuted integers from 1 to N . The argument represents N clusters of measured data.
- The first n_{est} elements of the vector containing randomly permuted integers represent indexes $\{i_{est}\} \subset \{i_c\}$ used to extract data clusters from the matrix \mathbf{M}_{PTQ} into the matrix \mathbf{E}_{PTQ} , that represents the estimation data subset.
- The last n_{val} elements of the vector containing randomly permuted integers represent indexes $\{i_{val}\} \subset \{i_c\}$ used to extract data clusters from the matrix \mathbf{M}_{PTQ} into the matrix \mathbf{V}_{PTQ} , that represents the validation data subset.

Partitioning of the training subset is schematically represented in Figure 7.

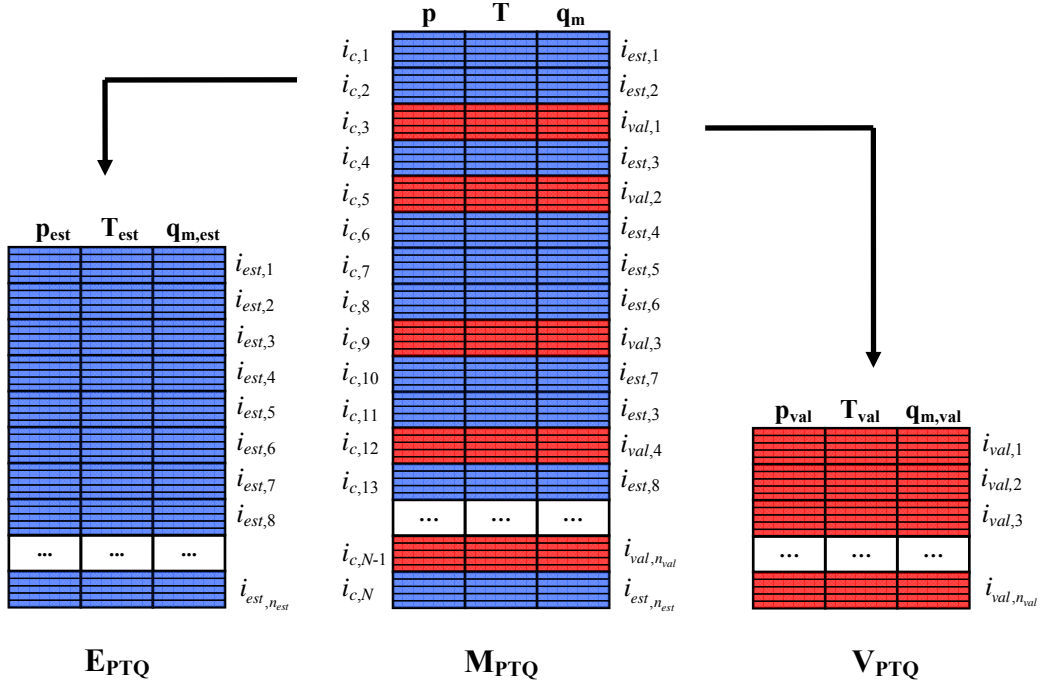


Figure 7: Partitioning of the training set (M_{PTQ}) into an estimation subset (E_{PTQ}) and a validation subset (V_{PTQ}).

3.3 Preparation of data

To obtain good neural network result some data transformation should be done. Input values of pressure and temperature are standardized by subtracting a mean and dividing by standard deviation.^[8] Consequently »standard normal« random variables with mean 0 and standard deviation 1 are achieved:

$$\bar{p}_{est,s} = 0, \sigma(p_{est,s}) = 1, \quad (23)$$

$$\bar{T}_{est,s} = 0, \sigma(T_{est,s}) = 1. \quad (24)$$

Standardization is essential for achieving good result due to the nature, i.e. radial symmetry, of radial basis function (Chapter 2.1). Original values of pressure and temperature have very different ranges (Figure 4), but after standardization they lie on similar intervals.

It is very important to standardize the validation data using mean and standard deviation of data from the estimation subset:

$$p_{val,s} = \frac{p_{val} - \bar{p}_{est}}{\sigma(p_{est})}, \quad (25)$$

$$T_{val,s} = \frac{T_{val} - \bar{T}_{est}}{\sigma(T_{est})}. \quad (26)$$

Standardized values of pressure, $p_{est,s}$ and $p_{val,s}$, and temperature, $T_{est,s}$ and $T_{val,s}$, from estimation and validation subsets are collected into corresponding matrices \mathbf{E}_{PT-s} and \mathbf{V}_{PT-s} , respectively.

3.4 Modelling of the measurement characteristic

Firstly, we approximate the function $q_m = f(p, T)$ with the linear regression model. The corresponding algorithm is described in Section 3.4.1. The approximative function, given as an output of linear regression model, represents the reference for estimation of performances of neural network models.

Secondly, we approximate the relationship between pressure, temperature and mass flow rate with two radial basis function network models. Within the second RBFN model prior knowledge regarding direct relationship between variables is incorporated. Both RBFN models are described in Sections 3.4.2 and 3.4.3.

3.4.1 Linear regression model

From the Equation (2) is obvious that if pressure is zero then the mass flow rate also equals zero. But on the other hand, chosen range of measurement points is distant from the origin of the coordinate system. Due to this fact, linear regression with bias, b^{LR} , was selected. The linear regression model has the form:

$$q_m^{LR} = k_p p + k_T T + b^{LR}, \quad (27)$$

where k_p and k_T are coefficients of the approximative plane.

The matrix \mathbf{E}_{PTO} is formed by collecting the input data from the estimation subset and a column of ones:

$$\mathbf{E}_{PTO}^T = \begin{pmatrix} p_{est,1} & p_{est,2} & \dots & p_{est,N \cdot n_{est}} \\ T_{est,1} & T_{est,2} & \dots & T_{est,N \cdot n_{est}} \\ 1 & 1 & \dots & 1 \end{pmatrix}. \quad (28)$$

Vector of measured values of the mass flow rate from the estimation subset is:

$$\mathbf{q}_{m,est}^T = (q_{m,est,1} \quad q_{m,est,2} \quad \dots \quad q_{m,est,N \cdot n_{est}}). \quad (29)$$

Both coefficients and bias,

$$\mathbf{K}^T = (k_p \quad k_T \quad b^{LR}), \quad (30)$$

could be obtained by solving the equation:

$$\mathbf{K} = \mathbf{E}_{\text{PTO}}^+ \cdot \mathbf{q}_{\text{m,est}}, \quad (31)$$

where standard pseudo inverse, $\mathbf{E}_{\text{PTO}}^+$, is calculated as:^[8]

$$\mathbf{E}_{\text{PTO}}^+ = (\mathbf{E}_{\text{PTO}}^T \cdot \mathbf{E}_{\text{PTO}})^{-1} \cdot \mathbf{E}_{\text{PTO}}^T. \quad (32)$$

Vectors of predicted values of the mass flow rate in data points from estimation and validation subsets are calculated as:

$$\mathbf{q}_{\text{m,est}}^{\text{LR}} = \mathbf{K} \cdot \mathbf{E}_{\text{PTO}}, \quad (33)$$

$$\mathbf{q}_{\text{m,val}}^{\text{LR}} = \mathbf{K} \cdot \mathbf{V}_{\text{PTO}}, \quad (34)$$

where the matrix \mathbf{V}_{PTO} is:

$$\mathbf{V}_{\text{PTO}}^T = \begin{pmatrix} P_{\text{val},1} & P_{\text{val},2} & \dots & P_{\text{val},N \cdot n_{\text{val}}} \\ T_{\text{val},1} & T_{\text{val},2} & \dots & T_{\text{val},N \cdot n_{\text{val}}} \\ 1 & 1 & \dots & 1 \end{pmatrix}. \quad (35)$$

For linear regression model it is not necessary to be built on standardized values.

3.4.2 Radial basis function network model NN-q

The purpose of the radial basis function network model called NN-q is approximation of the function $q_m = f(p, T)$. In the process of training of this neural network, input vectors are columns from the matrix $\mathbf{E}_{\text{PT-s}}^T$ and targets are values from vector $\mathbf{q}_{\text{m,est}}^T$.

When the network is trained, the model NN-q takes the following form:

$$q_m^{\text{NN-q}} = f_q(p_s, T_s). \quad (36)$$

It is used to predict the values of mass flow rates, $q_{\text{m,est}}^{\text{NN-q}}$ and $q_{\text{m,val}}^{\text{NN-q}}$, based on standardized values of pressure and temperature from the estimation and validation subsets, respectively.

3.4.3 Radial basis function network model NN-C

Direct relationship between variables p , T and q_m is stated by equation (2). With this prior knowledge being incorporated, the function C may be calculated as:

$$C(p, T) = q_m \frac{\sqrt{T}}{p} = \frac{A_m C_d C_*}{\sqrt{R/M}}. \quad (37)$$

The purpose of the radial basis function network model called NN-C is approximation of this function. In the process of training of the neural network NN-C, the input vectors are columns from the matrix \mathbf{E}_{pT-s}^T and targets are values from the vector \mathbf{C}_{est}^T .

The neural network NN-C takes the following form:

$$C^{NN-C} = f_C(p_s, T_s) \quad (38)$$

and it is used to predict the values C_{est}^{NN-C} and C_{val}^{NN-C} on standardized inputs from estimation and validation subsets, respectively.

With the RBFN model NN-C it is possible to calculate the mass flow rate as follows:

$$q_m^{NN-C} = \frac{P_s}{\sqrt{T_s}} C^{NN-C}(p_s, T_s) \quad (39)$$

We expect that by applying prior knowledge better generalization ability of the neural network model will be obtained.

3.5 Multifold cross-validation based on random partitioning of the training set

Parameters of a radial basis function network are *the number of neurons in hidden layer* and *spread* (or width) of radial basis functions. In order to estimate the approximation and generalization abilities of a neural network model with selected parameters, it is N_{iter} -times trained and validated on different estimation and validation subsets, respectively. In the beginning of every iteration, data clusters from the training set (the matrix \mathbf{M}_{PTQ}) are randomly divided into estimation subset (2/3 of clusters) and validation subset (1/3 of clusters), as described in Section 3.2. Afterwards, each model is trained on estimation subset and its performance is checked on both data subsets. Chosen measure of error is mean absolute percentage error, *mape*, which is described in Section 3.5.1.

Performances of a selected model, i.e. its approximation and generalization abilities, are obtained by averaging the corresponding values of *mape*, gained over N_{iter} iterations. The procedure is described in Section 3.5.2.

3.5.1 Mean absolute percentage error

Mean absolute percentage error (*mape*) is a measure of error of model's predicted values compared to the targets. Mean absolute percentage error on the estimation subset,

$$e_{est} = \frac{1}{n \cdot n_{est}} \sum_{i=1}^{n \cdot n_{est}} \frac{|q_{m,est,i}^M - q_{m,est,i}|}{q_{m,max}}, \quad (40)$$

reflects the approximation ability of the model – accuracy of its predictions on the estimation data subset.

Mean absolute percentage error on validation subset,

$$e_{val} = \frac{1}{n \cdot n_{val}} \sum_{i=1}^{n \cdot n_{val}} \frac{|q_{m,val,i}^M - q_{m,val,i}|}{q_{m,max}}, \quad (41)$$

reflects the generalization ability of a model – accuracy of its predictions on validation data subset.

In Equations (40) and (41) the mass flow rate (with superscripted index M), q_m^M , denotes predicted value by the model M and $q_{m,max}$ denotes maximum value of measured mass flow rate in the training set.

3.5.2 Estimation and validation errors

For the model with selected parameters, the estimation and validation errors are defined as:

$$E_{est} = \frac{1}{N_{iter}} \sum_{i=1}^{N_{iter}} e_{est}, \quad (42)$$

$$E_{val} = \frac{1}{N_{iter}} \sum_{i=1}^{N_{iter}} e_{val}, \quad (43)$$

respectively. They represent averages of corresponding mean average percentage errors, e_{est} and e_{val} , over N_{iter} iterations. Since the training set is N_{iter} -times randomly partitioned into estimation and validation subsets, errors E_{est} and E_{val} are more appropriate estimators of model performances than particular errors e_{est} and e_{val} .

In the case of RBFN models, the validation error, E_{val} , also represents **the objective function** for selection of a RBFN model with optimal parameters.

3.6 Optimization of parameters of RBFN model

With respect to the training algorithm of radial basis function network, represented in Section 2.2, we expected that if only one neuron is added at every epoch there would consequently exist one center of RBF in each cluster (in one of its 10 data points). But results of this approach did not meet our expectations – in most cases happened that each cluster did not have its own center of RBF, or, in other words, there was more than one center in some clusters but none in some others. For an example, have a look at Figure 8.

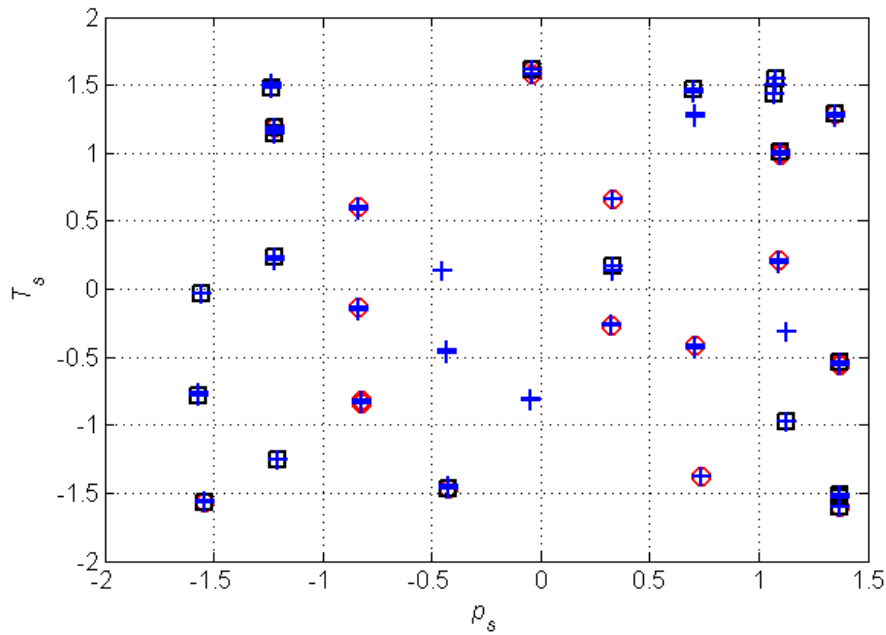


Figure 8: Positions of data clusters (blue crosses) from the estimation subset and centers of radial basis functions – for RBFN models NN-q (red circles) and NN-C (black squares). In this case the estimation subset consist of 30 clusters and *the number of neurons* is also 30.

3.7 Algorithm for determination of models and selection of optimal RBFN models

We developed a Matlab code with purpose to train and validate all three models. Our aim is also determination of the smallest but powerful enough RBFN models NN-q and NN-C.

The following operations are executed within the program:

- Settings: vector of *spreads*, vector of *number of neurons*, number of iterations, N_{iter}, \dots
- Loading of measured data.
- Loop for *number of neurons*.
 - Loop for *spread*.
 - Loop for iterations for random partitioning of the training set and training and validation of models.

- ◆ Random partitioning of the training set into estimation and validation subsets (Section 3.2).
 - ◆ Standardization of pressure and temperature from estimation and validation subsets (Section 3.3).
 - ◆ Calculation of values of the function C (Equation (37)).
 - ◆ Determination of the linear regression model (Section 3.4.1).
 - ◆ Training and validation of the RBFN model NN-q (Section 3.4.2).
 - ◆ Training and validation of the RBFN model NN-C (Section 3.4.3).
 - ◆ Simulation of models over entire input range (not only for measured (and standardized) values of pressure and temperature).
 - ◆ Calculation of errors e_{est} and e_{val} in the end of each iteration (Section 3.5.1).
 - ◆ Plots of measured data and models' outputs within each iteration.
- End of loop for iterations.
 - Calculation of the estimation and validation errors, E_{est} and E_{val} , for models with selected parameters (Section 3.5.2).
 - Plots of models' performances: E_{est} vs. *spread* and E_{val} vs. *spread* for fixed *number of neurons*.
- End of loop for *spread*.
- End of loop for *number of neurons*.
 - Evaluation and plot of performance of each model:

$$E_{est} = f(\text{number of neurons, spread of RBFs}). \quad (44)$$

$$E_{val} = f(\text{number of neurons, spread of RBFs}). \quad (45)$$

The linear regression model is independent of *number of neurons* and *spread*, thus errors E_{est}^{LR} and E_{val}^{LR} theoretically do not change due to variation of these parameters.

In general, parameters at which the objective function (E_{est} in our case) reaches its minimum are selected as optimal parameters of the neural network.

4 Results

4.1 Models' predictions of the mass flow rate

Diagrams in Figure 9 represent the mass flow rates and values of the function C_* , predicted by the models. Results of:

- the linear regression (LR) model are plotted in first column,
- the RBFN model NN-q are plotted in second column and
- the RBFN model NN-C are plotted in third column.

Data points from estimation and validation subsets are represented by blue and red stars, respectively, and black circles represent the predicted values by the corresponding model.

Within each iteration (represented as a row of diagrams), all models were built on the same estimation subset. In case of RBFN models, the values of pressure and temperature were standardized. However, their corresponding original values are presented in Figure 9. In the first and in the last two rows of diagrams, models with *spread* equal to 1 and 10, respectively, are presented. As mentioned before, the linear regression model is independent of *spread*.

At first sight the linear regression model predicts the values of mass flow rate from validation subset quite accurately. In general, neural networks are not good at extrapolation. This fact is confirmed by responses of models NN-q and NN-C for *spread* = 1 (Figure 9(a)) – in this case some of validation data lie outside of the range of estimation data and consequently predicted values are quite inaccurate. For *spread* = 10 (Figure 9(b)) both RBFN models predict values of mass flow rate more accurately, even though there are also some validation data outliers with respect to the ranges of estimation data. As expected, approximative surface over complete input range is quite smooth.

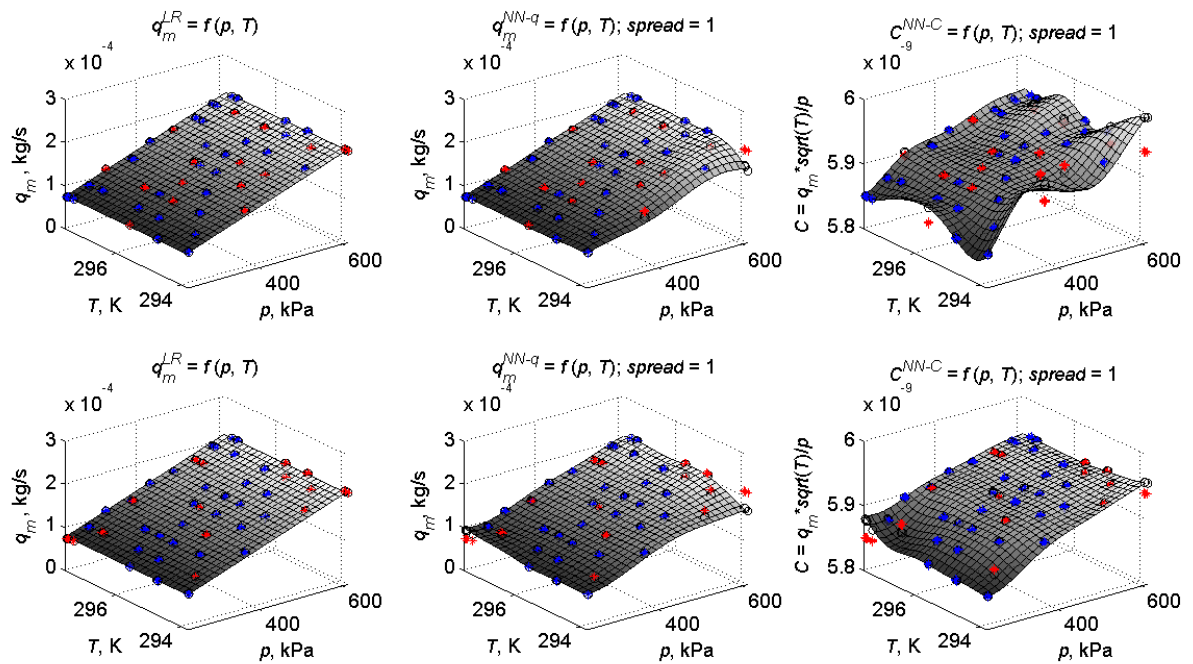
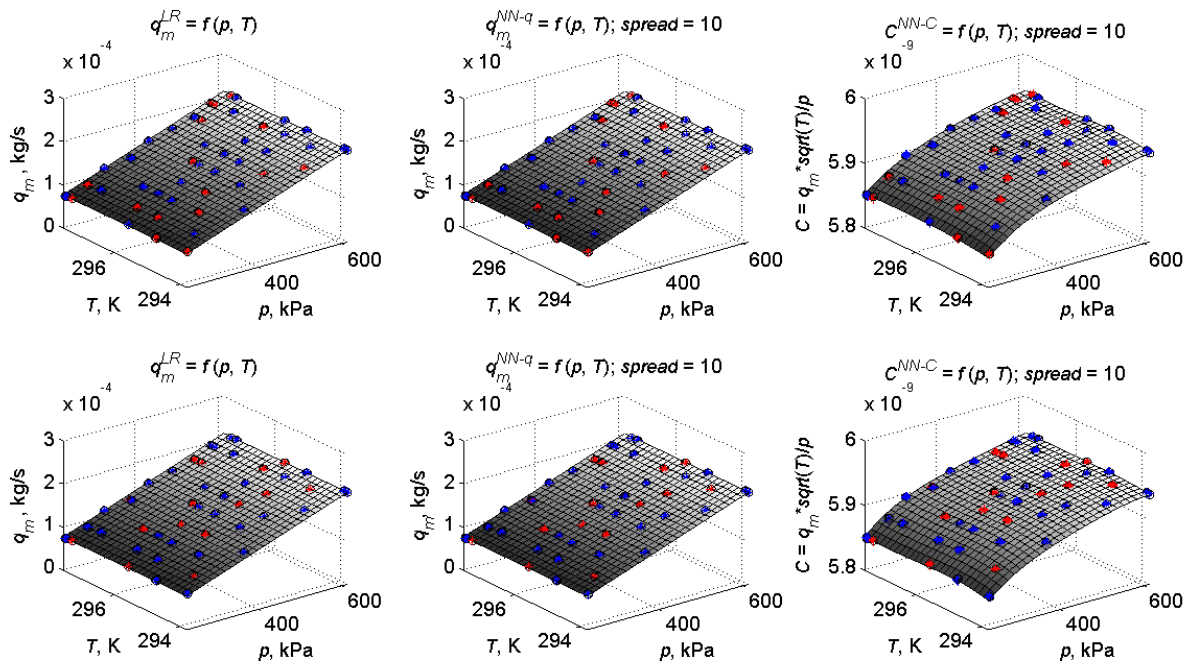

 (a) $spread = 1$

 (b) $spread = 10$

Figure 9: Predicted mass flow rates by the linear regression model (first column of diagrams) and by the RBFN model NN-q (second column); and predicted values of the function C by the RBFN model NN-C (third column). For RBFN models, the number of neurons is 25 and $spread$ of RBFs is 1 (first two rows of diagrams – (a)) or 10 (last two rows of diagrams – (b)).

4.2 Performances of models

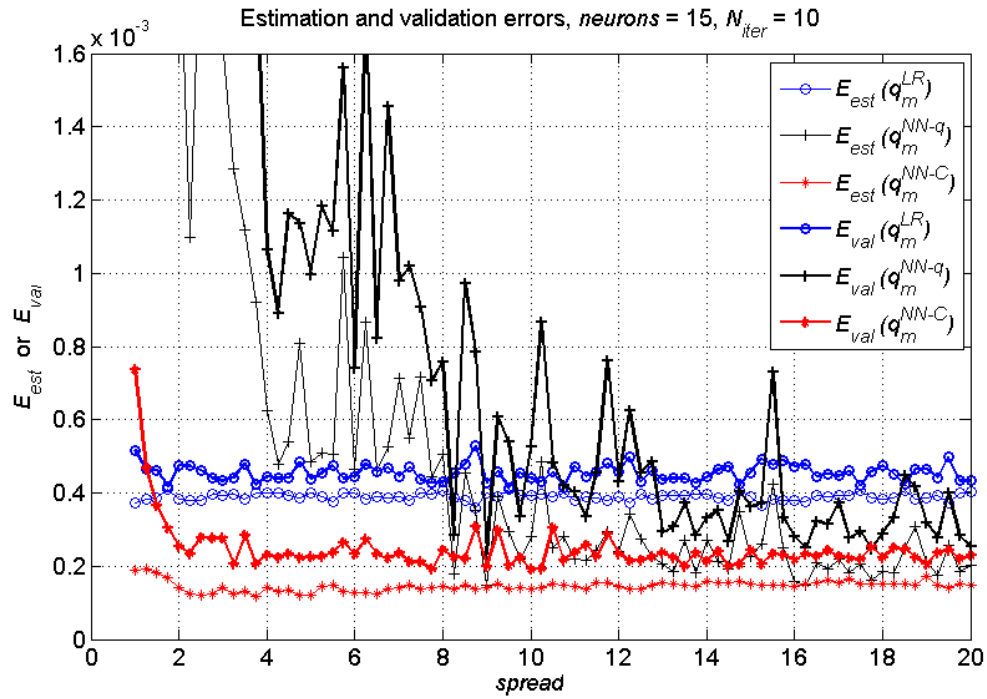
In Figure 10 are plotted two diagrams (for RBFN models with 15 and 25 *neurons in hidden layer*) representing estimation and validation errors for all models in dependence of *spread*.

The estimation and validation errors of linear regression model (thin and thick blue lines with circles, respectively) are quite constant. Even though this model is independent of *spread*, its error functions are plotted with the purpose of being the references for RBFN models. In our opinion, variations of errors of the linear regression model are caused due to more or less favourable (random) partitioning of the training data set. If the RBFN model NN-q consists of only 15 neurons in the hidden layer, its estimation and generalization abilities are not satisfactory (Figure 10(a)). On the other hand, the model NN-C approximates the estimation data the most accurately of all models. The same holds also for performance on validation data if *spread* is larger than 1. In comparison to the model NN-q with 15 neurons, the same model with 25 neurons possesses better approximative and predictive accuracies at smaller values of *spread* (Figure 10(b)). For *spread* > 8, its performance on validation data is nearly the same as performance of the model NN-C.

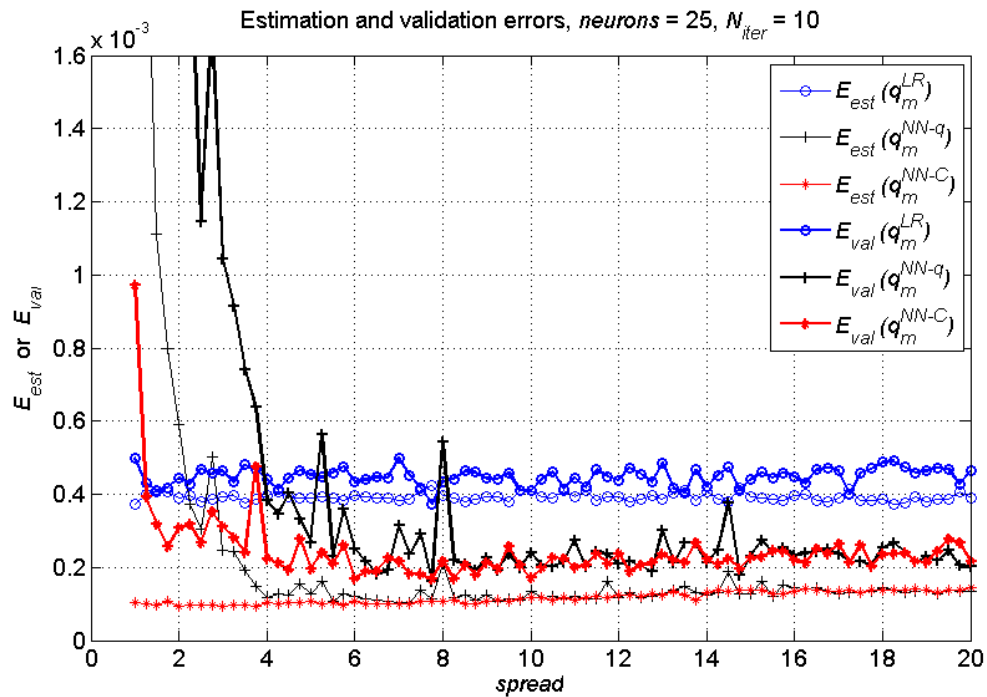
In Figure 11 there are two diagrams representing values of the objective function, i.e. the validation error, in dependence of *number of neurons* and *spread* of radial basis function networks. In comparison to the model NN-C, the region with minimal and almost constant validation error is for the model NN-q reached at larger values of *the number of neurons* and *spread*.

As the result of evaluation of the objective function, we may choose the smallest but power enough RBFN models:

- the optimal model NN-q: 25 *neurons* in hidden layer and *spread* of RBFs equal to 10,
- the optimal model NN-C: 15 *neurons* in hidden layer and *spread* of RBFs equal to 5.

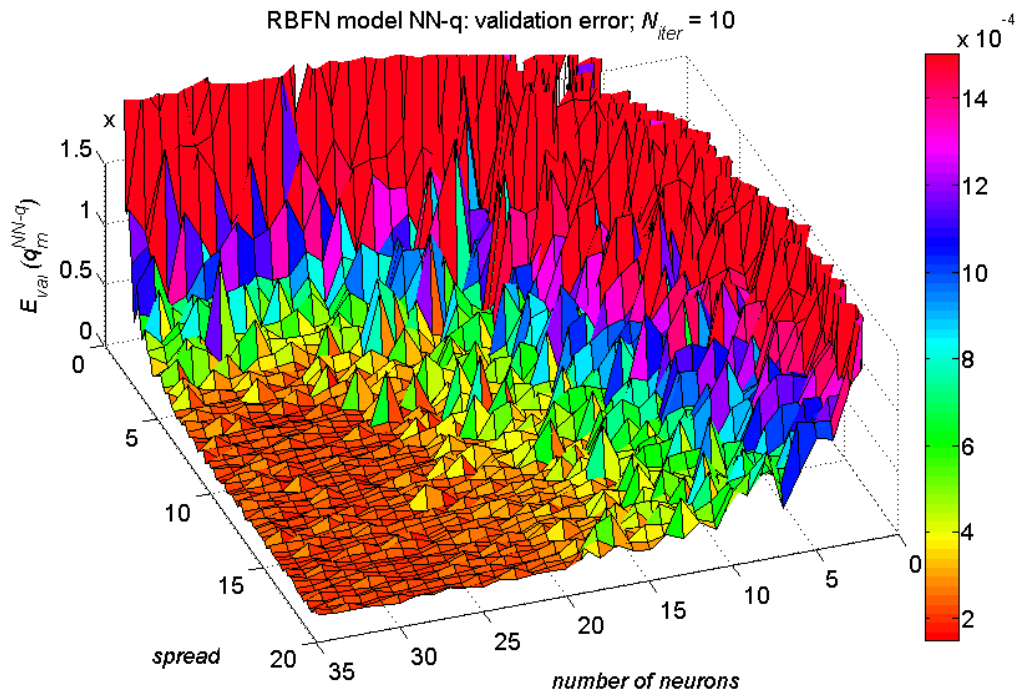


(a) number of neurons = 15.

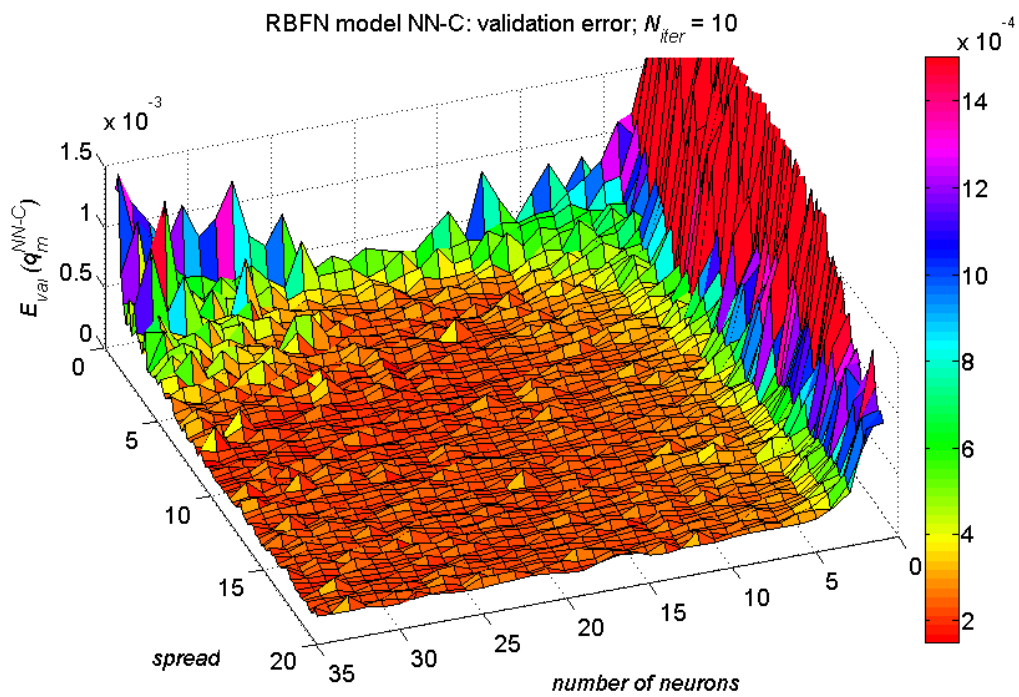


(b) number of neurons = 25.

Figure 10: Estimation and validation errors, E_{est} and E_{val} , for all models. RBFN models consist (a) of 15 or (b) of 25 neurons.



(a) RBFN model NN-q.



(b) RBFN model NN-C.

Figure 11: The objective function, i.e. the validation error, of RBFN models NN-q (a) and NN-C (b).

4.3 Response of RBFN models at large values of spread

It is expected that if $spread \rightarrow \infty$ (width of Gaussian radial basis function $\rightarrow \infty$), there will be little difference between predictions of linear regression model and RBFN models. Convergence of models' responses (and also errors) is not present for the values of $spread$ up to 20, which is shown in Figures 10 and 11. Due to this reason we simulated the models and calculated values of error functions, E_{est} and E_{val} , for much larger range of $spread$ – from 1 to 10^5 . Errors are plotted on diagrams in Figures 13 and 14.

In the vicinity of $spread$ equal to $2 \cdot 10^3$, validation errors for both RBFN models increase on the level of validation error of linear regression model, which confirms the expectations. Global minima of both validation errors are approximately equal ($\approx 0,02\%$) and lie within the range of expected variations. Besides this, the validation error of the model NN-q falls below the error of the model NN-C for values of $spread$ approximately from the interval $(50, 10^3)$, where also its minimum is reached. In the case of model NN-q, the difference between the value of $spread$ and the ranges of standardized inputs (zero mean and standard deviation equal to 1) is very large. That means that only narrow parts of $radbas$ (or Gaussian) functions in the vicinity of their peaks are used in the model. In our opinion, the reason for this is the difference in rates of nonlinearity of functions $q_m = f_q(p, T)$ and $C = f_c(p, T)$, that are approximated by the models NN-q and NN-C, respectively. By multiplying the mass flow rate, q_m , by \sqrt{T}/p , the dominating influence of pressure is excluded and consequently the function $f_c(p, T)$ expresses larger rate of nonlinearity than the function $f_q(p, T)$ – have a look at Figure 9. The larger is the standard deviation of the Gaussian function, more flat is this function in the vicinity of its peak and it also includes smaller rate of nonlinearity over the range of standardized inputs. For examples of $radbas$ functions with $spread$ equal to 10 and 50, see Figure 12. If the rate of nonlinearity of a function that we want to approximate is smaller, e.g. the function $q_m = f_q(p, T)$, then the corresponding model will consist of RBFs with larger $spread$.

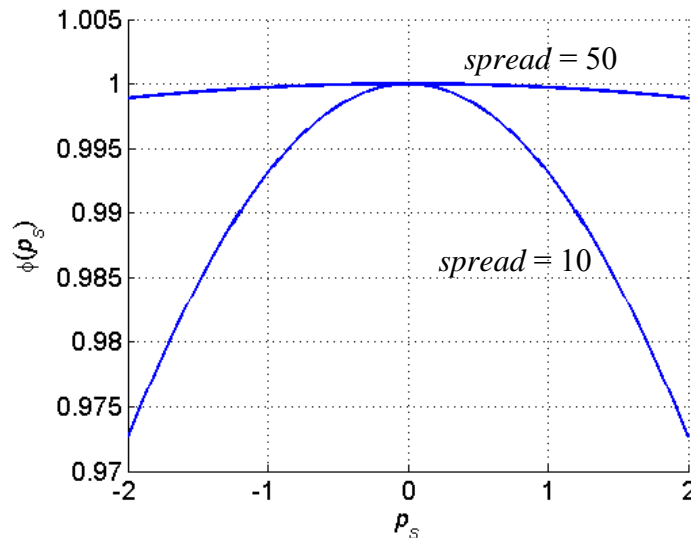


Figure 12: Two $radbas$ functions with $spread$ equal to 10 and 50, in dependence of standardized variable p_s .

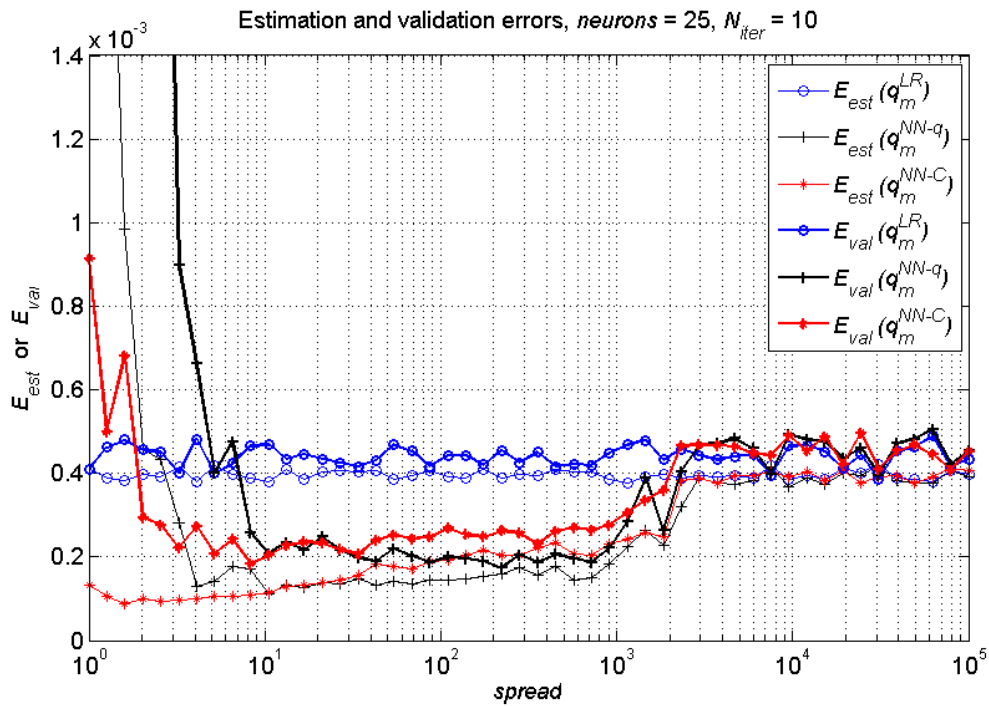
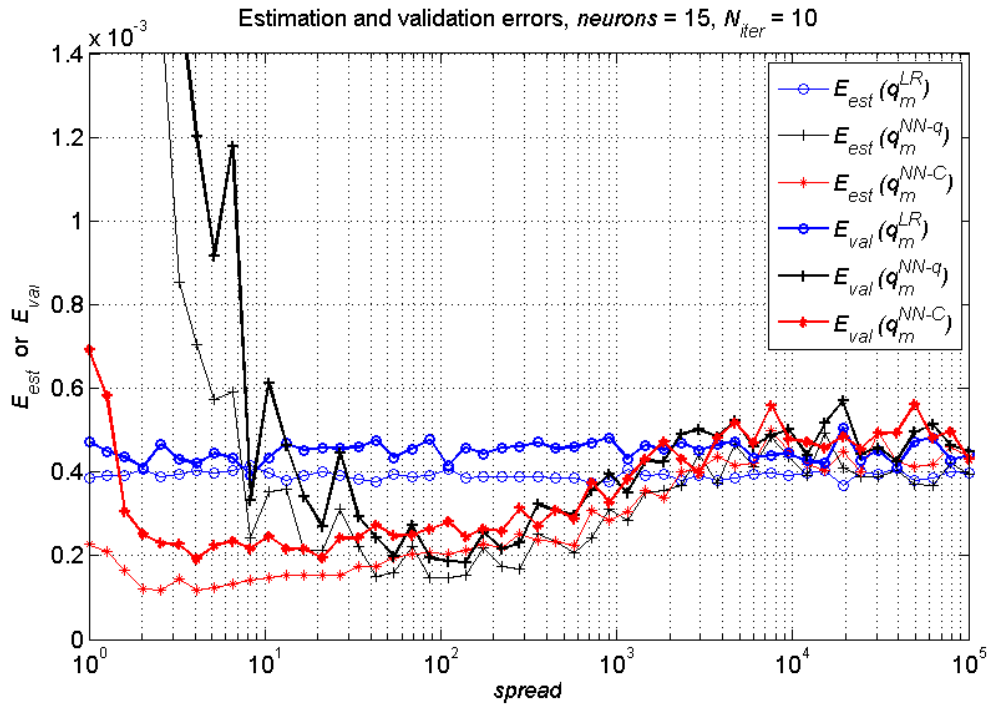
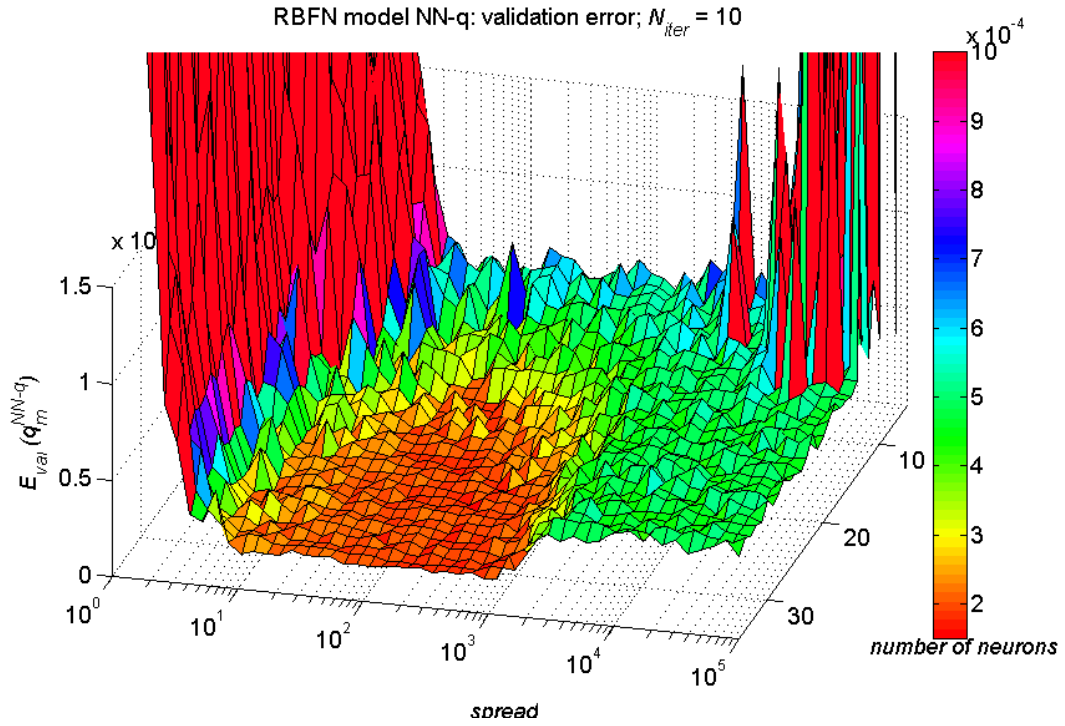
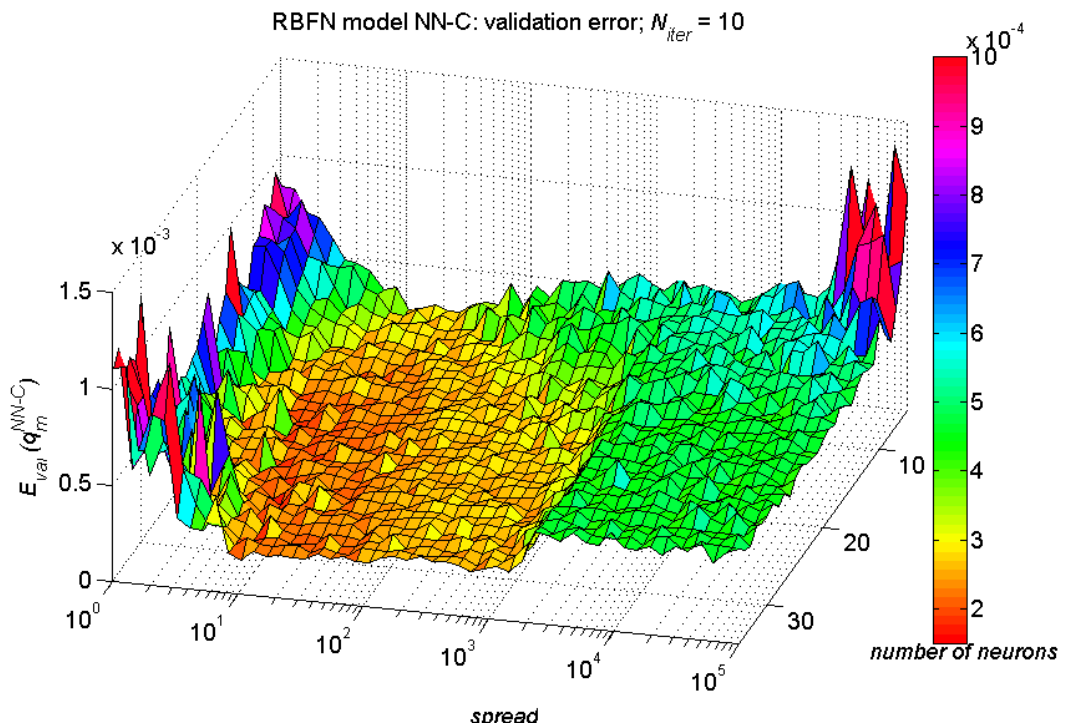


Figure 13: Estimation and validation errors, E_{est} and E_{val} , over for all models and for large range of $spread$. RBFN models consist (a) of 15 or (b) of 25 neurons.



(a) RBFN model NN-q.



(b) RBFN model NN-C.

Figure 14: The objective function, i.e. the validation error, of RBFN models NN-q (a) and NN-C (b) for large range of $spread$.

5 Conclusions

In this seminar some research in the field of identification of the measurement characteristic of the CFVN mass flow meter by radial basis function network models was done. The mass flow rate is determined by gas pressure and temperature upstream of the critical flow Venturi nozzle. Values of these variables were measured in 450 measurement points, which represented the training set of 45 data clusters with 10 data points each. With purpose to approximate the relationship between variables p , T and q_m , three models were trained and validated:

- the linear regression model (Equation (27)):

$$q_m^{LR} = k_1 p + k_2 T + b^{LR},$$

- the RBFN model NN-q (Equation (36)):

$$q_m^{NN-q} = f_q(p_s, T_s),$$

- the RBFN model NN-C with integrated prior knowledge regarding direct relationship between variables (Equations (38) and (39)):

$$q_m^{NN-C} = C^{NN-C} \frac{p_s}{\sqrt{T_s}},$$

where

$$C^{NN-C} = f_C(p_s, T_s).$$

The validation error was selected as the objective function for determination of optimal parameters of RBFN models. The optimal model is defined as the smallest but powerful enough model that reaches the minimum of its objective function. By variation of parameters of RBFN models and their multifold training and validation on different subsets of randomly partitioned training data set, we selected RBFN models with the following parameters as optimal:

- **the RBFN model NN-q:** 25 *neurons* in hidden layer and *spread* of RBFs equal to 10,
- **the RBFN model NN-C:** 15 *neurons* in hidden layer and *spread* of RBFs equal to 5.

Both optimal RBFN models reach better generalization abilities than the linear regression model. This is a consequence of the fact that even though the rate of nonlinearity of the measurement characteristic is small, it have to be involved in the models.

The model NN-C reaches the minimum of the objective function with RBFN structure, consisting of less neurons than the model NN-q. Based on the results, it is possible to conclude that in our case by incorporating the prior knowledge into the model, the same generalization ability can be achieved with »smaller« RBFN model.

It was also proven that if *spread* $\rightarrow \infty$ there is little difference between the predicted mass flow rates by the linear regression model and the RBFN models.

6 References

- [1] ISO 9300:2005: Measurement of gas flow by means of critical flow Venturi nozzles.
- [2] Kutin, J.: Postopek LMPS-PP-02: Umerjanje kritičnih oz. zvočnih merilnih šob, Laboratorij za meritve v procesnem strojništvu, Faculty of Mechanical Engineering, University of Ljubljana, Ljubljana, 2010.
- [3] Critical Flow Nozzles According to DIN EN 9300, datasheet, TetraTec Instruments GmbH, 2006. Available on website: http://tetrathec-instruments.com/English/Download/SNZ_data_e.pdf.
- [4] Picard, A., Ravis, R. S., Gläser, M., Fujii, K.: Revised formula for the density of moist air (CIPM-2007), Metrologia 45 (2008) 149–155.
- [5] Potočnik, P.: Practical Considerations, Lectures at the course Neural Network, Laboratory of Synergetics, Faculty of Mechanical Engineering, University of Ljubljana, Ljubljana, 2010.
- [6] Matlab Help, Matlab R2010b, The MathWorks Inc., 2010.
- [7] Hudson Beale, M., Hagan, M. T., Demuth, H. B.: Neural Network Toolbox 7, User's Guide, The MathWorks, Inc., 2010. Available on website: http://www.mathworks.com/help/pdf_doc/nnet/nnet.pdf
- [8] Potočnik, P.: Radial Basis Function Networks, Lectures at the course Neural Network, Laboratory of Synergetics, Faculty of Mechanical Engineering, University of Ljubljana, Ljubljana, 2010.