PREDICTIVE CONTROL OF NONLINEAR PROCESSES USING FUZZY HAMMERSTEIN MODEL

Zuzana Dideková and Slavomír Kajan
Institute of Robotics and Cybernetics, ,
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology,
Bratislava, Slovak Republic
zuzana.didekova@stuba.sk
slavomir.kajan@stuba.sk

Štefan Kozák
Institute of Automotive Mechatronics
Faculty of Electrical Engineering and Information Technology
Slovak University of Technology, Bratislava,
Slovak Republic
stefan.kozak@stuba.sk

Abstract
In this paper a novel methodology for modeling and control of a class of nonlinear systems is proposed. The non-linear system is modeled by the so-called fuzzy Hammerstein model, and the control strategy based on generalized predictive control (GPC) algorithm is applied. Simulation of a modified DC motor with a nonlinear block demonstrates effectiveness of the proposed approach. The proposed control is compared with other predictive control approaches based on artificial neural network models. Obtained results confirm that the proposed methodology can be used for modeling and control of different types of nonlinear processes in industry applications.

Key words
Hammerstein model, fuzzy logic, DC motor

1 Introduction

The automatic control (AC) field is more than eighty years old. Development of the field has been very dynamic; AC principles pervaded such diverse fields as biology, medicine, banking, economics etc., and their applications can be encountered practically everywhere in our daily life, e.g. in home devices, communication systems, modern types of vehicles, in banking, health care services, etc. Conventional standard control strategies have been widely used in industries for several decades. A vast majority of automatic control loops in the process industries (90%) still rely on various forms of the ubiquitous PID controller commercially available for over seventy years [Kozák, 2002]. However, common control methods of are not available for highly non-linear processes. In the last years, research activity has been oriented to solving modeling and control tasks for non-linear systems using computing intelligence methods.

Fuzzy sets fall into effective methods of computing intelligence where it is possible to improve expert methods to modeling of non-linear processes. Artificial neural networks (ANNs), genetic algorithms (GA), fuzzy logic, hybrid and adaptive soft methods allow transforming experience of an expert into a control algorithm design [Hypiusová and Kajan, 2013].

The Hammerstein model is one of the simplest and most popular members of the general family of block-oriented non-linear dynamic models constructed from series and/or parallel combinations of linear dynamic models and static non-linearities [Narendra and Gallman, 1966]. More specifically, the Hammerstein model structure consists of a single static non-linearity *f(.)*, connected in cascade with a single linear dynamic model defined by the transfer function $G(z^{-1})$ [Doyle et al., 2002]. Block scheme of a discrete fuzzy Hammerstein (FH) model is shown in Fig. 1. FIS block represents fuzzy static non-linearity and $A(z^{-1})$, $B(z^{-1})$ are nominator and denominator polynomials, respectively, of the process transfer function $G(z^{-1})$.

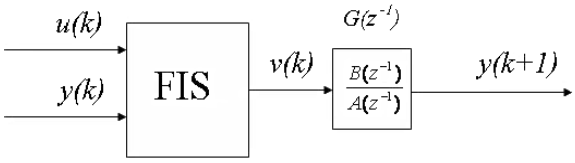

Figure 1. Block scheme of a fuzzy Hammerstein model with fuzzy static non-linearity (FIS) and linear dynamic model

2 Identification Problem Formulation

Consider discrete representation of the fuzzy Hammerstein (FH) model

$$y(k+1)=\sum_{i=1}^{na} a_i y(k-i+1) + \sum_{i=1}^{n_b} b_i \frac{\sum_{j=1}^{N_R} \beta_j(u(k-i-n_d+1), y(k-i-n_d+1)) d_j}{\sum_{j=1}^{N_R} \beta_j(u(k-i-n_d+1), y(k-i-n_d+1))} \quad (1)$$

The weight $0 \le \beta_j(u, y) \le 1$ is the overall truth value of the *j*-th rule and is computed as

$$\beta_j(u,y) = \beta_{(i_u,i_y)}(u,y) = A_{(u,i_u)}(u) \times A_{(y,i_y)}(y)\,, \quad (2)$$

where $A_{(u,i_u)}$, $A_{(y,i_y)}$ are i_u-th and i_y-th Gaussian functions of the input u and input y, respectively; $N_R = M_u.M_y$ denotes the number of fuzzy model rules, M_u and M_y are numbers of fuzzy sets of input u and input y, $i_u = 1, 2, ..., M_u$ and $i_y = 1, 2, ..., M_y$, respectively; n_a , n_b are orders of the denominator and numerator, respectively of the ARX (autoregressive with exogenous terms) model, n_d is the discrete time delay; a_i, b_i are nominator and denominator coefficients of the transfer function $G(z^1)$ of the linear dynamic model, d_j are output coefficients of the fuzzy static non-linearity [Abonyi *et al*. 2000].

In the sequel, coefficients a_i and b_i, of the linear dynamic model will be denoted „linear parameters", while coefficients d_j belonging to the fuzzy model will be called "non-linear parameters".

In this paper a one-step iterative FH model identification is proposed. The class of one-step iterative solutions includes techniques that alternately refine estimates of the linear dynamics and the static non-linearity [Doyle et al. 2002]. Only dynamic data are used for identification. This method is a modification of the algorithm proposed by Narendra and Gallman [1966] for identification of Hammerstein models with a polynomial static non-linearity. In this case, model parameters are obtained by separating the estimation of the linear dynamics from the static non-linear part. As none of these parts is known beforehand it results in an iterative identification algorithm. For the sake of simplicity it is assumed that the antecedent part of the fuzzy model (the fuzzy sets) is designed manually based on a priori knowledge. For this task one could also employ numerical optimization techniques such as fuzzy clustering. The identification algorithm then determines the consequent parameters a_i, b_i and d_j. This is a clearly non-linear optimization problem. Two approaches can be used: on-line or off-line identification. For simplicity, an iterative off-line procedure is used in the sequel.

Suppose that the parameters a_i, b_i of the linear dynamic model are known. Then, parameters $\bar{d} = [d_1,\ldots,d_{N_R}]^{\mathrm{T}}$ of the nonlinear part can be estimated by solving the regression problem using following equation.

$$\bar{y}_d = \bar{\bar{\phi}}_d \bar{d} + \bar{\varepsilon}\,, \quad (3)$$

where $\bar{\varepsilon}$ denotes the zero-mean, normally-distributed modeling error. For N measured data pairs, the $\bar{y}_d$ vector and the $\bar{\bar{\phi}}_d$ matrix are given by:

$$\bar{y}_d = \begin{bmatrix} y_d(1) \\ y_d(2) \\ \vdots \\ y_d(N) \end{bmatrix}, \quad \bar{\bar{\phi}}_d = \begin{bmatrix} \bar{\phi}_d(1) \\ \bar{\phi}_d(2) \\ \vdots \\ \bar{\phi}_d(N) \end{bmatrix} \quad (4)$$

with $y_d(k) = y(k) - \sum_{i=1}^{n_a} a_i y(k-i)$,

$$\bar{\phi}_d(k) = [\sum_{i=1}^{n_b} b_i \beta_1(u(k-i-n_d)), \ldots, \sum_{i=1}^{n_b} b_i \beta_{N_R}(u(k-i-n_d))] / \sum_{i=1}^{N_R} \beta_i(u(k-i-n_d)) \quad (5)$$

The least-squares estimate of coefficients in (5) is

$$\bar{d} = \left[\bar{\bar{\phi}}_d^T \bar{\bar{\phi}}_d\right]^{-1} (\bar{\bar{\phi}}_d)^T \bar{y}_d\,. \quad (6)$$

Parameters of the linear dynamic model $\bar{\theta}_l = \left[a_1,\ldots,a_{n_a},b_1,\ldots,b_{n_b}\right]^T$ can be estimated by solving the following regression problem

$$\bar{y}_l = \bar{\bar{\phi}}_l \bar{\theta}_l + \bar{\varepsilon} \quad (7)$$

where

$$\bar{y}_l = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix} \quad \bar{\bar{\phi}}_l = \begin{bmatrix} \bar{\phi}_l(1) \\ \bar{\phi}_l(2) \\ \vdots \\ \bar{\phi}_l(N) \end{bmatrix}, \quad (8)$$

and

$$\bar{\phi}_l(k) = [y(k-1),\ldots,y(k-n_a), \frac{\sum_{j=1}^{N_R} \beta_j(u(k-1-n_d))d_j}{\sum_{j=1}^{N_R} \beta_j(u(k-1-n_d))}, \ldots, \frac{\sum_{j=1}^{N_R} \beta_j(u(k-n_b-n_d))d_j}{\sum_{j=1}^{N_R} \beta_j(u(k-n_b-n_d))}] \quad (9)$$

The least-squares estimate of the linear parameters can be computed by

$$\bar{\theta}_l = \left[\bar{\bar{\phi}}_l^T \bar{\bar{\phi}}_l\right]^{-1} \bar{\bar{\phi}}_l^T \bar{y}_l\,. \quad (10)$$

Finally, the nonlinear parameters $\bar{d}$ are estimated again by using (6) and the whole procedure is iteratively repeated according to the desired accuracy.

As the static gain of the FH model is determined by both static non-linearity and gain of the linear part, the nonlinear model is redundant. The unity gain of the identified linear model will be ensured by using constrained quadratic programming (QP) instead of (10).

$$\min_{\theta_l} \left\{ \frac{1}{2} \bar{\theta}_l^T \bar{\bar{H}} \bar{\theta}_l + \bar{c}^T \bar{\theta}_l \right\}, \quad (11)$$

with $\bar{\bar{H}} = 2\bar{\bar{\phi}}_l^T \bar{\bar{\phi}}_l$ and $\bar{c} = -2(\bar{\bar{\phi}}_l)^T \bar{y}_l$.

The constraint on $\bar{\theta}_l$ can be expressed as

$$[1,1,\ldots,1]\bar{\theta}_l = 1 \quad (12)$$

The iterative algorithm is stopped when convergence in both $\bar{\theta}_l$ and $\bar{d}$ occurs; here, the term "convergence" means that the infinity norm of the difference in the parameters between two successive iterations is smaller than a predefined threshold. Note that the above algorithm is computationally expensive because it requires solving one least-squares problem and one quadratic program in each iteration. Moreover since it is an off-line algorithm, if new input-output data become available the whole algorithm has to be restarted.

3 Non-linear system control using the fuzzy Hammerstein model and the GPC algorithm

To control a nonlinear system, its FH model in combination with the generalized predictive control (GPC) algorithm have been used. As the GPC algorithm uses the ARX model of the plant, it is necessary to recast the FH model of the system to ARX model in each computation of control action. FH model linearized in the operating point is used. By its linearization we obtained a discrete-time model in the form

$$y(k+1)=\sum_{i=1}^{na} a_i y(k-i+1) + \sum_{i=1}^{n_b} b_i \frac{\sum_{j=1}^{N_R} \beta_j(u(k-i-n_d+1), y(k-i-n_d+1))d_j}{\sum_{j=1}^{N_R} \beta_j(u(k-i-n_d+1), y(k-i-n_d+1))} \tag{13}$$

In the operating point $(x, y(x)) = ((u(k), y(k)), y(k+1, u(k), y(k)))$ it becomes a linear equation standing for the ARX model

$$y(k+1)=\sum_{i=1}^{n_b} dyu(k-i-n_d+1)\Delta u(k-i-n_d+1) + (1+dyy(k))y(k) +$$

$$+ \sum_{i=1}^{n_{a2}-1}(dyy(k-i)-dyy(k-i+1))y(k-i) - dyy(k-n_{a2}+1)y(k-n_{a2})$$

$$n_{a2} = n_a + 1 \quad \text{if } n_a > n_b$$
$$= n_b + 1 \quad \text{if } n_a \le n_b$$

$$dyu(k-i) = b_{i+1}\left.\frac{\partial v(k,u(k),y(k))}{\partial u(k)}\right|_{\substack{u(k)=u(k-i-1)\\ y(k)=y(k-i)}}$$

for $i \in <0; n_b - 1>$

$$dyy(k-i) = a_{i+1} + b_{i+1}\left.\frac{\partial v(k,u(k),y(k))}{\partial y(k)}\right|_{\substack{u(k)=u(k-i-1)\\ y(k)=y(k-i)}}$$

for $i \in <0; n_{a2} - 1>$

$$v(k,u(k),y(k)) = \frac{\sum_{i_u=1}^{M_u}\sum_{i_y=1}^{M_y}\beta_{i_u i_y}(u(k),y(k)).d_{i_u i_y}}{\sum_{i_u=1}^{M_u}\sum_{i_y=1}^{M_y}\beta_{i_u i_y}(u(k),y(k))}$$

$$\beta_{i_u i_y}(u(k),y(k)) =$$

$$\exp\left(\frac{-(u(k)-c_{ui_u})^2}{2sig_{ui_u}^{\ 2}}\right).\exp\left(\frac{-(y(k)-c_{yi_y})^2}{2sig_{yi_y}^{\ 2}}\right) \tag{14}$$

M_u and M_y represent numbers of fuzzy sets for inputs $u(k)$ and $y(k)$, and c_{ui_u}, sig_{ui_u}, c_{yi_y} and sig_{yi_y} denote parameters of particular fuzzy sets.

Next we proceed according to the common GPC algorithm. [Kozák, 2006]

4 Case Study: DC Motor Control

4.1 Process Description

The controlled plant is a laboratory DC motor (Fig. 2). Controlled variable is the angular velocity which is manipulated by the input voltage within 0-10V; it is sensed by optical electronics which gives the output voltage in the range 0-10V.

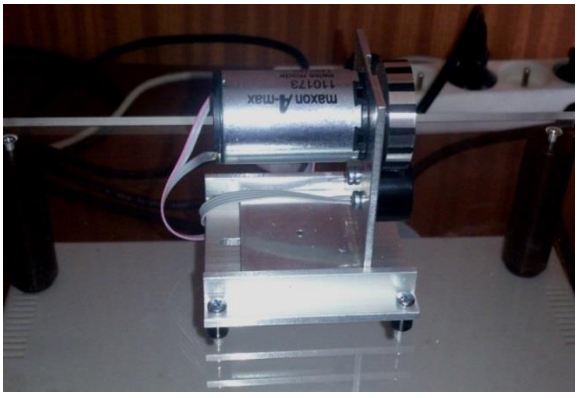

Figure 2. Laboratory DC motor

The laboratory DC motor with nonlinearity was simulated in Matlab-Simulink (Fig. 3). The whole system can be divided into the linear part of DC motor system (LS) and a nonlinear block (NB) in the feedback part which simulates system behavior with a load p and nonlinearity (Fig. 4).

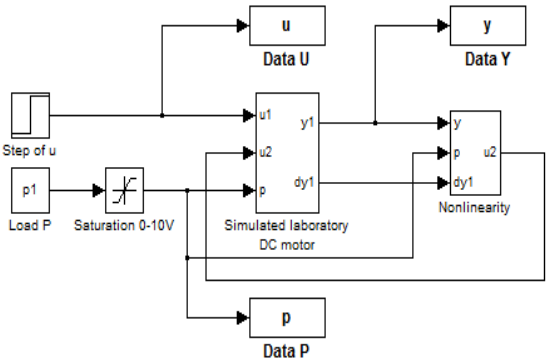

Figure 3. Simulation block scheme of a DC motor with nonlinearity (input u, output y and load p)

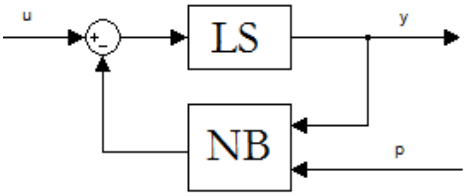

Figure 4. Block scheme of the linear DC motor (LS) with the nonlinear feedback block (NB)

According to the Fig. 4 the output value *y* and the load *p* enter to the nonlinear block; its output is connected with the linear part of the DC motor via feedback loop [Hypiusová and Kajan, 2013]. In this paper, we consider a constant load $p = 1V$.

Input-output characteristics of the DC motor with load $p=1V$ is depicted in Fig. 5 with 3 operating points marked: OP1 (0.5 V, 2.0828 V), OP2 (1.5 V, 4.7972 V) and OP3 (6.0 V, 8.2342 V). For these 3 operating points output variable time responses U_{out} for 0.5V step changes in input variable U_{in} are depicted in Fig. 6. As it can be seen from the pictures, the nonlinearity is static (Fig. 5) and also dynamic (Fig. 6).

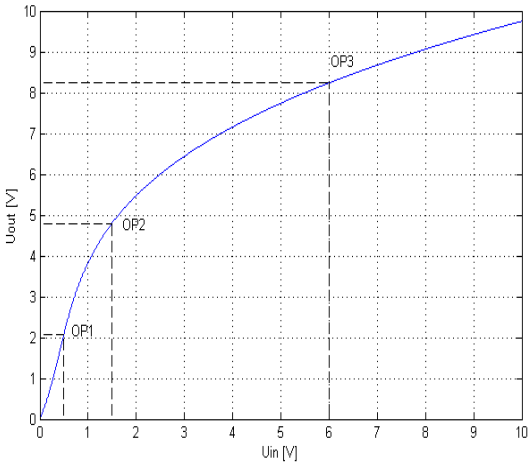

Figure 5. Input-output characteristics of the modified DC motor with load $p = 1V$: OP1 (0.5 V, 2.0828 V), OP2 (1.5 V, 4.7972 V) and OP3 (6.0 V, 8.2342 V)

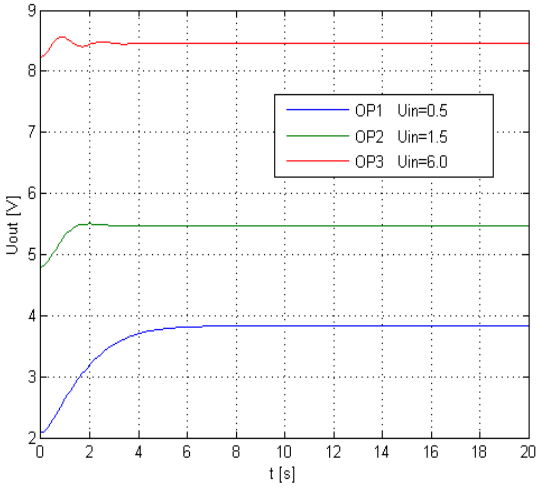

Figure 6. Time responses of output variable U_{out} to 0.5V step changes in the input variable U_{in} in individual operating points: OP1 (0.5 V, 2.0828 V), OP2 (1.5 V, 4.7972 V) and OP3 (6.0 V, 8.2342 V)

4.2 Process Modeling

Training data for the fuzzy Hammerstein model (FH) identification are depicted in Fig.7. Sampling period $T_s = 0.1$s was used, and the input variable was changing every 5 seconds.

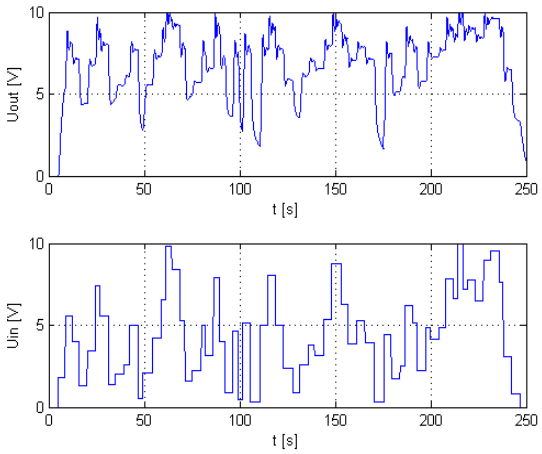

Figure 7. Training data

For the FH model, a second order ARX model was selected.

Initial ARX model was identified by Matlab function *arx* and fuzzy model by Matlab functions *genfis*2 (with *RADII* = 0.5) and *anfis* (with 100 epochs). For ending the iteration, threshold was set to 10^{-10} and maximum number of iterations to 500.

Final computed ARX model is:

$$G(z^{-1}) = \frac{(-0.1309z^{-1} - 0.2595z^{-2}).10^{-5}}{1 - 0.5000z^{-1} - 0.5000z^{-2}} \qquad (15)$$

And the output fuzzy part of model is:

$\vec{d}$ = [-255.9273 35.1927 753.3018 33.3217 -5.8297 -48.0981 1725.0176 -145.0510 -8379.8437] ^{T}

Inputs to the fuzzy part of model are depicted in Fig. 8 and Fig. 9.

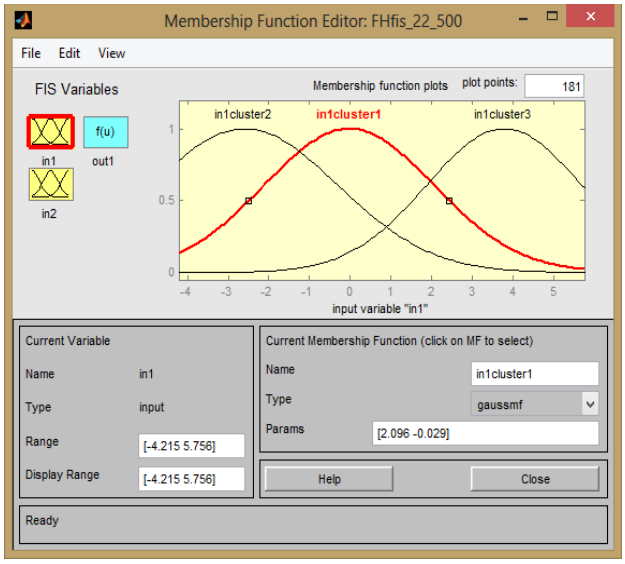

Figure 8. Membership functions for the first input: input voltage (U_{in}) to the fuzzy part of FH model

Comparison of the training and the testing data with the FH model output are shown in Fig. 10, Fig. 11 (training data) and Fig. 12 and Fig. 13 (testing data). The proposed FH model is useful for the control under the GPC predictive algorithm. The parameters of GPC

algorithm are: number of prediction steps N = 10 and weighting coefficient λ= 0.5 .

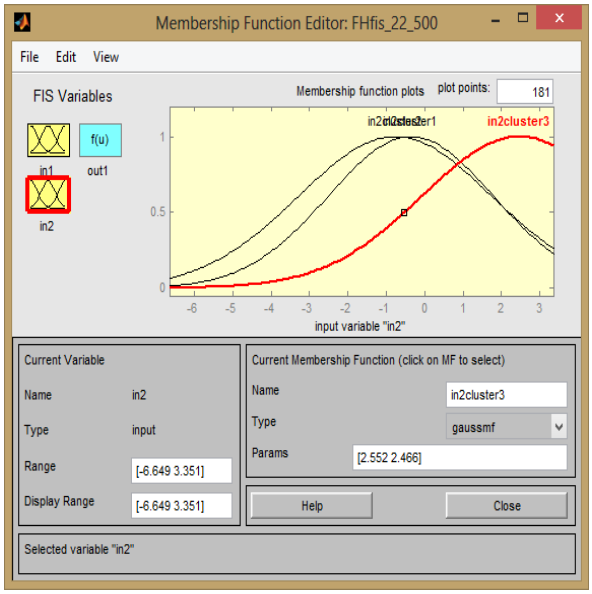

Figure 9. Membership functions for the second input: output voltage representing angular velocity (U_{out}) to the fuzzy part of FH model.

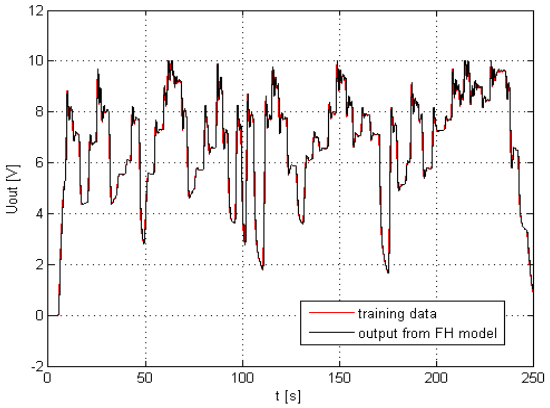

Figure 10. Comparison of training data with the FH model output

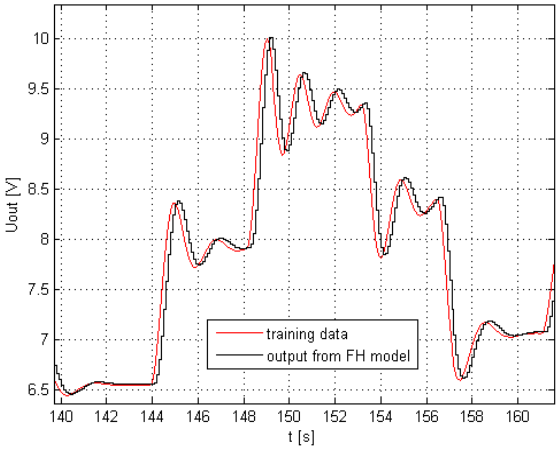

Figure 11. Comparison of training data with the FH model output – a detail

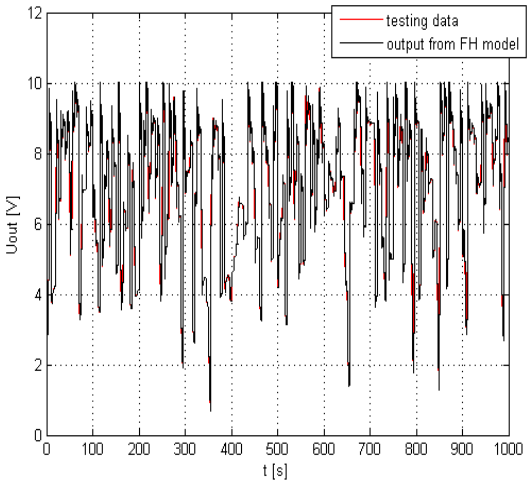

Figure 12. Comparison of testing data with the FH model output

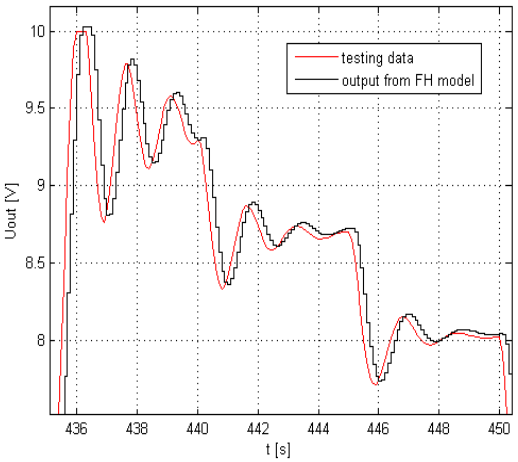

Figure 13. Comparison of testing data with the FH model output – a detail

4.3 Comparison of the Predictive Control algorithms based on FH and MLP Models

In this section, the predictive control algorithms based on FH and MLP (multi layer perceptron) models are compared.

In the predictive control algorithm based on FH model, the parameters of GPC algorithm are: number of prediction steps N = 10 and weighting coefficient λ= 0.5 .

Predictive control with MLP model is realized using artificial neural network as a plant model. The model is represented by a three layer ANN with one hidden layer of 7 neurons. The inputs to the model are 3 previous values of U_{in} and 3 previous values of U_{out} and the output is current value of U_{out}.

Fig. 14 and Fig. 15 show time responses of controlled output (U_{out}) and control action (U_{in}) under predictive control based on FH and MLP models to step change in reference variable w from w_s = 4 V to w_e = 5V (Fig. 14) and several different step changes of w within 4 V and 8 V (Fig. 15).

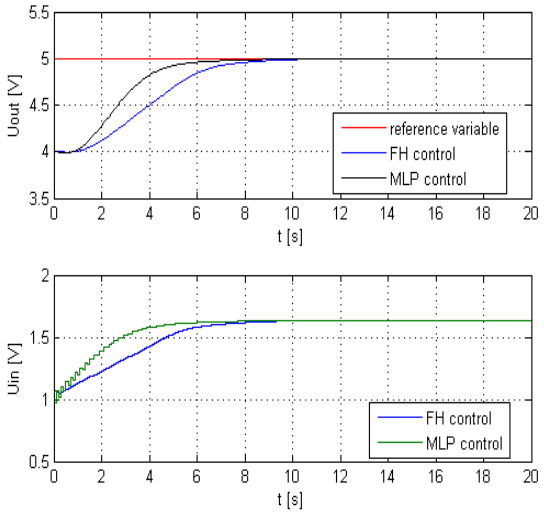

Figure 14. Time responses of the controlled output (U_{out}) and control action (U_{in}) under predictive control based on FH and MLP models to a step change in reference variable (from w_s = 4 V to w_e = 5 V)

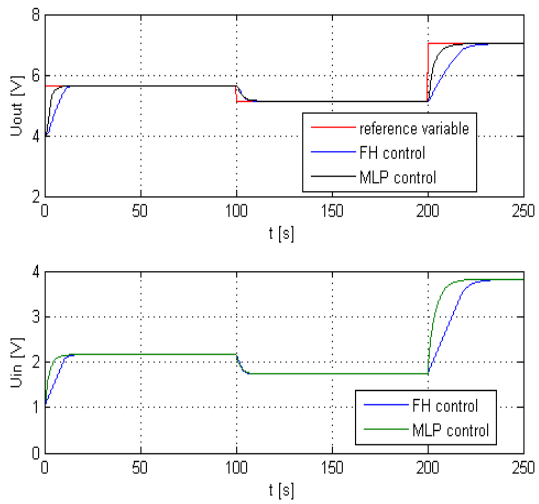

Figure 15. Time responses of controlled output (U_{out}) and control action (U_{in}) under predictive control based on FH and MLP models to several different step changes in w between 4 V and 8 V

It can be seen, that both applied control algorithms provide stability in the range between 4 V and 7 V.

5 Conclusion

The proposed non-linear FH model based predictive controller was verified on a laboratory plant - a DC motor; the results were generalized for modeling and control of a selected class of non-linear systems. The designed control algorithm was implemented on the DC motor using Matlab-Simulink. From the obtained results we can state that the fuzzy Hammerstein model is effective mainly for modeling of non-linear plants with static non-linearity.

Developed control algorithms provide high performance of time responses to step changes in reference variable. The generalized predictive control algorithm based on the fuzzy Hammerstein model is suitable for the control of uncertain plants showing robust properties.

Acknowledgements

This work has been supported by the grants of the Slovak Scientific Grant Agency (VEGA) No. 1/0937/14 and No. 1/1241/12.

References

Abonyi, J., Babuška, B., Botto, M., Avala, Szeifert, F., Doyle, F.J., Pearson, R.K., Ogunnaike, B. A. (2002). Identification and Control Using Volterra Models, Springer London.

Hypiusová, M, Kajan, S. (2013). Robust controller design using edge theorem and genetic algorithm, International Review of Automatic Control.194-200.

Kozák, Š. (2002). Advanced Control Methods, FEI STU, Bratislava.

Kozák, Š. (2006). Lectures on Advance Theory of Automatic Control. FEI STU. Bratislava. Internal materials.

Nagy, F. (2000). Identification and control of nonlinear systems using fuzzy Hammerstein models, Industrial and Engineering Chemistry Research, 4302 – 4314.

Narendra, K. S., Gallman, P. G. (1966). An iterative method for the identification of nonlinear systems using the Hammerstein model, IEEE Trans. Automatic Control. 546 – 550.