Predictive Control of Nonlinear Plant Using Piecewise-Linear Neural Model

Daniel Honc
Faculty of Electrical Engineering and Informatics
University of Pardubice
Pardubice, Czech Republic
Email: daniel.honc@upce.cz

Petr Dolezel
Faculty of Electrical Engineering and Informatics
University of Pardubice
Pardubice, Czech Republic
Email: petr.dolezel@upce.cz

Lumir Gago
Faculty of Electrical Engineering and Informatics
University of Pardubice
Pardubice, Czech Republic
Email: lumir.gago@student.upce.cz

***Abstract*—A special form of a predictive controller is presented in this paper. Based on previous authors' work, a piecewise-linear neural model of nonlinear plant to be controlled is adopted to local linearization. The linearized model is then used for control action evaluation using a predictive controller. Although the linearization using piecewise-linear neural network is simple and efficient, it provides the model in a nonstandard form. Therefore, the proposed predictive controller is designed in order to handle that nonstandard model without any customization. At the end of the paper, the illustrative example demonstrates the main features of the introduced solution.**

I. Introduction

If a system can be described by a finite set of individual linear subsystems, where each subsystem is valid in a distinguished region of state space, this system is then called a piecewise-linear system. Piecewise-linear systems have received a lot of attention recently for their equivalence to other classes of the systems [1], but mainly for their practicality. Namely, they provide a structure which is capable of approximating nonlinear systems. This structure can be consequently dealt with using techniques which were originally proposed for linear systems. Besides, some specialized tools for a piecewise-linear system analysis were published, too [2].

Identification of the nonlinear systems using piecewise-linear model is generally not a simple problem. In a very rare case, when the number of linear regions is known, the issue transforms into a classical linear system identification; the parameters of the subsystems can be estimated from the corresponding input-output data. Nevertheless, the significant challenge occurs in the situation when the state space division is not known. Many contributions dealing with the identification of piecewise-linear systems (or more general piecewise-affine systems) have been proposed over time; Vidal et al. propose an algebraic approach [3], while others use clustering approach [4] or the bounded error method [5].

Once a piecewise-linear model of the nonlinear system is designed, it can be advantageously applied in stability investigation, prediction, fault diagnosis and especially in process control. Ameur et. al [6] use piecewise-linear and piecewise-affine approach to electropneumatic systems control; authors of [7] propose an adaptive controller for piecewise-linear systems; and in [8], uncertain parameters of piecewise-linear systems are even considered, when a controller is designed.

In 2012, a novel way of piecewise-linear model design was proposed [9]. Authors used a special topology of feedforward neural network, which could ingenuously provide a piecewise-linear model of nonlinear system to any degree of accuracy. This approach was then used for nonlinear systems control using PID controller [10], [11], and tested in industrial environment [12]. The issue is, that control techniques applied in the mentioned contributions used classical control approaches. In so doing, the piecewise-linear model derived in [9] had to be transformed a bit to fit to those classical approaches. However, the family of control laws is much richer these days and some of them can be applied directly without any transformation. To be more specific, a suitable version of predictive controller is introduced in this paper. This controller can directly utilize the model of a nonlinear system provided by [9] and, apparently, it offers much larger tools for control response tuning, than the classical approaches.

The paper is organized as follows. In section II, the aim of this paper is properly formulated. Then, the particular predictive controller, which fits to the problem defined in section II, is suggested (section III) and its features are demonstrated on a laboratory system (section IV).

II. Problem formulation

In [9], a novel and computationally simple technique, which provides a piecewise-linear model of a nonlinear system, is presented. Functionally, it works as illustrated in Fig. 1.

The deterministic form of a provided piecewise-linear model is as follows.

$$\begin{aligned} &A^1(q^{-1})y(k) = B^1(q^{-1})u(k) + c^1, \text{if } \mathbf{x} \in \mathbf{X_1} \\ &A^2(q^{-1})y(k) = B^2(q^{-1})u(k) + c^2, \text{if } \mathbf{x} \in \mathbf{X_2} \\ &\vdots \\ &A^R(q^{-1})y(k) = B^R(q^{-1})u(k) + c^R, \text{if } \mathbf{x} \in \mathbf{X_R} \end{aligned} \quad (1)$$

In equation above, k is a discrete time, $y(k)$ is the output of the model, $u(k)$ is the input to the model and q denotes the forward shift operator, i.e. $q^{-1}y(k) = y(k-1)$. The vector $\mathbf{x}$ defines the current state of the model,

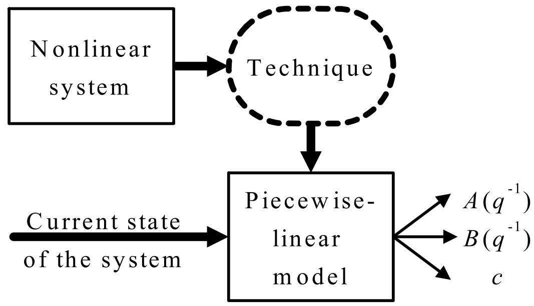

Fig. 1. Technique.

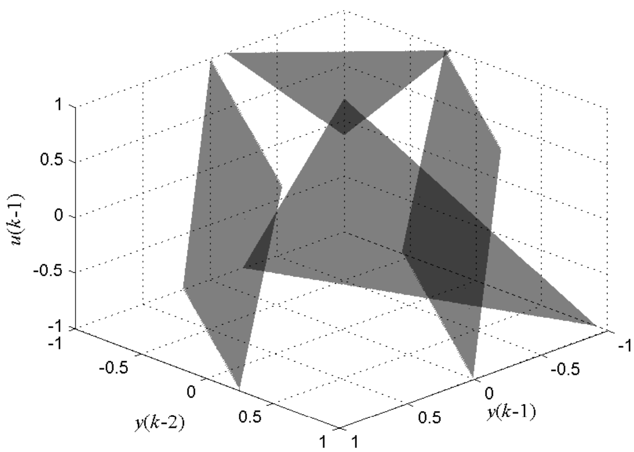

Fig. 2. State space partition of demonstrative model.

$\mathbf{x} = [u(k-1), \cdots, u(k-m), y(k-1), \cdots, y(k-n)]^{\mathrm{T}}$. Then, $\mathbf{X} = \bigcup_{i \in \{1,2,\cdots\},R} \mathbf{X_i}$ denotes the state space partition into closed regions. Moreover, $A^i(q)$, $B^i(q)$, $i = 1, 2, \cdots, R$, are the linear filters, which, together with the constant c^i, determine the linear subsystem valid for the particular region i. The filters are defined as follows.

$$B^i(q^{-1}) = [0 + b_1^i q^{-1} + b_2^i q^{-2} + ... + b_m^i q^{-m}], \quad (2)$$

$$A^i(q^{-1}) = [1 + a_1^i q^{-1} + a_2^i q^{-2} + ... + a_n^i q^{-n}], m \leq n. \quad (3)$$

where m is the order of the filter $B^i(q^{-1})$ and n is the order of the filter $A^i(q^{-1})$.

For an illustration, state space partition of a particular piecewise-linear model for $m = 1$ and $n = 2$ provided by the mentioned technique can be figured as shown in Fig. 2.

Obviously, the constant value c_i does not correspond with the ordinary specification of linear systems and consequent controller design. Thus, the piecewise-linear model (1) was transformed in [9] in order to remove the constant value c_i. The aim of this paper is different. In following sections, a type of predictive controller is proposed here to deal with the system described by model (1) directly without any transformation.

III. Predictive controller derivation

Model predictive control is a modern technique of process control. In comparison to other modern techniques, model predictive controllers minimize weighted future control errors and control efford taking into acount constraints and using concept of receeding horizon. The comprehensive review of the family of predictive control approaches can be found in [13], [14] or [15].

In our approach, the cost function is defined as follows.

$$J = \sum_{j=1}^{N} (\hat{y}(k+j) - w(k+j))^2 + r \sum_{j=1}^{N-1} (u(k+j) - u(k+j-1))^2, \quad (4)$$

where k is the current discrete time slot, $y(k)$ is the output of the system to be controlled, $w(k)$ is its reference value, $\hat{y}(k+j)$ is its predicted value j time slots ahead, $u(k)$ is the current control action, N is the length of the time horizon and r is the control action change penalization ($r > 0$). Tuning of this parameter affects the dynamics of the control action.

For further processing, it is convenient to transform the cost function to its matrix form. Thus,

$$J = (\mathbf{y} - \mathbf{w})^{\mathrm{T}} (\mathbf{y} - \mathbf{w}) + \mathbf{u}^{\mathrm{T}} \mathbf{R} \mathbf{u}, \quad (5)$$

where $\mathbf{y} = \begin{bmatrix} \hat{y}(k+1) \\ \hat{y}(k+2) \\ \vdots \\ \hat{y}(k+N) \end{bmatrix}$, $\mathbf{w} = \begin{bmatrix} w(k+1) \\ w(k+2) \\ \vdots \\ w(k+N) \end{bmatrix}$, $\mathbf{u} = \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N-1) \end{bmatrix}$, and $\mathbf{R} = \begin{bmatrix} r & -r & 0 & \cdots & 0 \\ -r & 2r & -r & \cdots & 0 \\ 0 & -r & 2r & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & r \end{bmatrix}$,

Apparently, it is necessary to determine a control action course $u(k), u(k+1), \cdots, u(k+N-1)$ which keeps the cost function (4) to be minimal. In the following paragraphs, the intuitive way of this course determination is shown for the piecewise-linear model (1), where $m = 2, n = 2$. The generalization, however, would be obvious.

Thus, suppose an i^{th} linear submodel of a piecewise-linear model (1) as follows. The upper indices are removed for the simplicity of the notation.

$$y(k) + a_1 y(k-1) + a_2 y(k-2) = b_1 u(k-1) + b_2 u(k-2) + c. \quad (6)$$

According to the previous equation, the following set of predictive equations can be provided.

$$\hat{y}(k+1) + a_1 y(k) + a_2 y(k-1) = b_1 u(k) + b_2 u(k-1) + c, \quad (7)$$

$$\hat{y}(k+2) + a_1 \hat{y}(k+1) + a_2 y(k) = b_1 u(k+1) + b_2 u(k) + c, \quad (8)$$

$$\hat{y}(k+3) + a_1 \hat{y}(k+2) + a_2 \hat{y}(k+1) = b_1 u(k+2) + b_2 u(k+1) + c, \quad (9)$$

$$\vdots$$

$$\hat{y}(k+N) + a_1 \hat{y}(k+N-1) + a_2 \hat{y}(k+N-2) = b_1 u(k+N-1) + b_2 u(k+N-2) + c, \quad (10)$$

The symbol $\hat{y}$ means that the value is predicted using the model, while y means measured value of system output.

Again, the set of equations above is transformed into a matrix form. Note that the predicted and measured parts of the equations are separated.

$$\mathbf{A}_p \mathbf{y} = \mathbf{B}_p \mathbf{u} + \mathbf{A}_m \begin{bmatrix} y(k) \\ y(k-1) \end{bmatrix} + \mathbf{B}_m u(k-1) + \mathbf{C}_m c, \quad (11)$$

where $\mathbf{A}_p = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ a_1 & 1 & 0 & \cdots & 0 \\ a_2 & a_1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$,

$\mathbf{B}_p = \begin{bmatrix} b_1 & 0 & 0 & \cdots & 0 \\ b_2 & b_1 & 0 & \cdots & 0 \\ 0 & b_2 & b_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & b_1 \end{bmatrix}$, $\mathbf{A}_m = \begin{bmatrix} -a_1 & -a_2 \\ -a_2 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix}$,

$\mathbf{B}_m = \begin{bmatrix} b_2 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$, $\mathbf{C}_m = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$.

Now, a predictor equation can be derived.

$$\mathbf{y} = \mathbf{A}_p^{-1} \mathbf{B}_p \mathbf{u} + \mathbf{A}_p^{-1} \mathbf{A}_m \begin{bmatrix} y(k) \\ y(k-1) \end{bmatrix} + \mathbf{A}_p^{-1} \mathbf{B}_m u(k-1) + \mathbf{A}_p^{-1} \mathbf{C}_m c, \quad (12)$$

and after a particular substitution,

$$\mathbf{y} = \mathbf{G}\mathbf{u} + \mathbf{F}_p \mathbf{x}_p. \quad (13)$$

In equation above, $\mathbf{G} = \mathbf{A}_p^{-1} \mathbf{B}_p$, $\mathbf{F}_p$ consists of three matrices in a row, i.e. $\mathbf{F}_p = \begin{bmatrix} \mathbf{A}_p^{-1} \mathbf{A}_m & \mathbf{A}_p^{-1} \mathbf{B}_m & \mathbf{A}_p^{-1} \mathbf{C}_m \end{bmatrix}$, and $\mathbf{x}_p = \begin{bmatrix} y(k) & y(k-1) & u(k-1) & c \end{bmatrix}^{\mathrm{T}}$.

The predictor is compounded of two components; the term $\mathbf{G}\mathbf{u}$ is called the forced response and the remaining term $\mathbf{F}_p \mathbf{x}_p$ is titled as the free response. The free response is the system response assuming that the current and future control actions are zero. The forced response is the system response due to the nonzero current and future control actions. Therefore, a controller is able to affect only the forced response part of the predictor. For the simplicity of the notation, the free response is labeled as $\mathbf{f}$ in the following equations; i.e. $\mathbf{f} = \mathbf{F}_p \mathbf{x}_p$.

Using (13), the future course $\mathbf{y}$ can be predicted with respect to the control action course $\mathbf{u}$. Thus, eq. (5) can be written as follows.

$$J = (\mathbf{G}\mathbf{u} + \mathbf{f} - \mathbf{w})^{\mathrm{T}} (\mathbf{G}\mathbf{u} + \mathbf{f} - \mathbf{w}) + \mathbf{u}^{\mathrm{T}} \mathbf{R} \mathbf{u} \quad (14)$$

Now, using some basic operations of matrix algebra, the formulation above can be transformed as follows.

$$J = \left(\mathbf{u}^{\mathrm{T}} \mathbf{G}^{\mathrm{T}} + \mathbf{f}^{\mathrm{T}} - \mathbf{w}^{\mathrm{T}}\right) (\mathbf{G}\mathbf{u} + \mathbf{f} - \mathbf{w}) + \mathbf{u}^{\mathrm{T}} \mathbf{R} \mathbf{u} = \mathbf{u}^{\mathrm{T}} \mathbf{G}^{\mathrm{T}} \mathbf{G} \mathbf{u} + \mathbf{u}^{\mathrm{T}} \mathbf{G}^{\mathrm{T}} \mathbf{f} - \mathbf{u}^{\mathrm{T}} \mathbf{G}^{\mathrm{T}} \mathbf{w} + \mathbf{f}^{\mathrm{T}} \mathbf{G} \mathbf{u} + \mathbf{f}^{\mathrm{T}} \mathbf{f} - \mathbf{f}^{\mathrm{T}} \mathbf{w} - \mathbf{w}^{\mathrm{T}} \mathbf{G} \mathbf{u} - \mathbf{w}^{\mathrm{T}} \mathbf{f} + \mathbf{w}^{\mathrm{T}} \mathbf{w} + \mathbf{u}^{\mathrm{T}} \mathbf{R} \mathbf{u}. \quad (15)$$

The previous equation can be formally simplified as follows.

$$J = \mathbf{u}^{\mathrm{T}} \mathbf{H} \mathbf{u} + \mathbf{u}^{\mathrm{T}} \mathbf{g} + \mathbf{g}^{\mathrm{T}} \mathbf{u} + k = \mathbf{u}^{\mathrm{T}} \mathbf{H} \mathbf{u} + 2 \mathbf{u}^{\mathrm{T}} \mathbf{g} + k, \quad (16)$$

where $\mathbf{H} = \mathbf{G}^{\mathrm{T}} \mathbf{G} + \mathbf{R}$, $\mathbf{g} = \mathbf{G}^{\mathrm{T}} (\mathbf{f} - \mathbf{w})$ and $k = (\mathbf{f} - \mathbf{w})^{\mathrm{T}} (\mathbf{f} - \mathbf{w})$.

The cost function (16) can be efficiently minimized by completing the square, i.e.

$$J = \left(\mathbf{u} + \mathbf{H}^{-1} \mathbf{g}\right)^{\mathrm{T}} \mathbf{H} \left(\mathbf{u} + \mathbf{H}^{-1} \mathbf{g}\right) - \mathbf{g}^{\mathrm{T}} \mathbf{H}^{-1} \mathbf{g} + k. \quad (17)$$

Now, assuming that $\left(\mathbf{H}^{-1}\right)^{\mathrm{T}} = \mathbf{H}^{-1}$ and no constrains, the cost function (17) can be minimized analytically. The mentioned assumption is fulfilled - see the definitions of matrices $\mathbf{H}$ and $\mathbf{R}$. Thus,

$$\mathbf{u} = -\mathbf{H}^{-1} \mathbf{g} = \left(\mathbf{G}^{\mathrm{T}} \mathbf{G} + \mathbf{R}\right)^{-1} \mathbf{G}^{\mathrm{T}} (\mathbf{w} - \mathbf{F}_p \mathbf{x}_p), \quad (18)$$

or

$$\mathbf{u} = \mathbf{L} (\mathbf{w} - \mathbf{F}_p \mathbf{x}_p), \quad (19)$$

where $\mathbf{L} = \left(\mathbf{G}^{\mathrm{T}} \mathbf{G} + \mathbf{R}\right)^{-1} \mathbf{G}^{\mathrm{T}}$.

Since only the current control action value is required, the term (19) can be simplified to

$$u(k) = \mathbf{l} (\mathbf{w} - \mathbf{F}_p \mathbf{x}_p), \quad (20)$$

where $\mathbf{l}$ is the first row of the matrix $\mathbf{L}$.

Considering (20), the control loop using a piecewise-linear neural model of the system and the derived predictive controller can be illustrated as seen in Fig. 3.

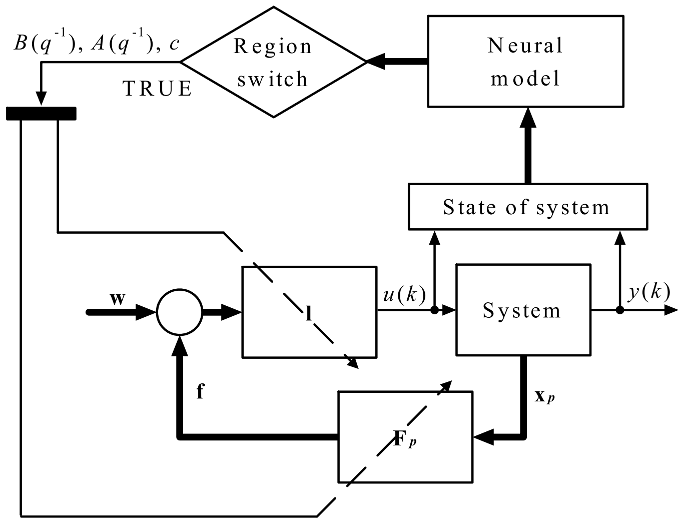

Fig. 3. Control loop.

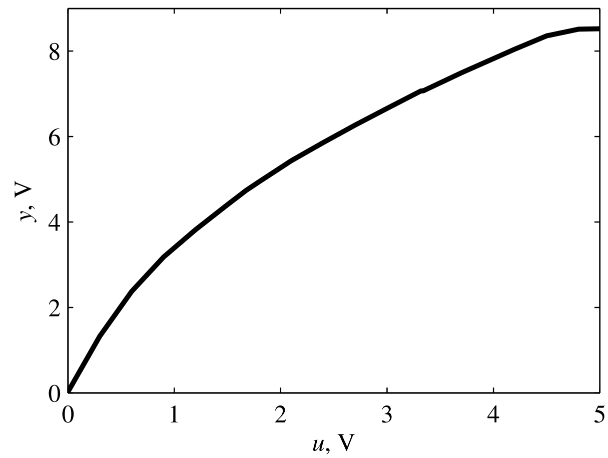

Fig. 5. Static characteristic of the system.

Fig. 4. GUNT RT 050 speed control laboratory system.

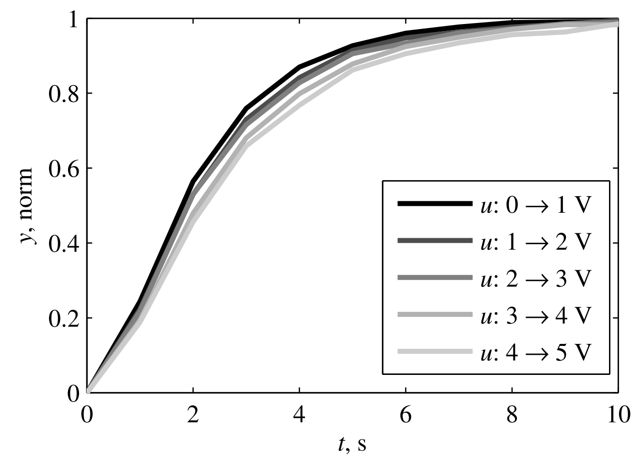

Fig. 6. Normalized response to step functions along various working points.

IV. Illustrative Example

As an illustrative example, a predictive control of GUNT RT 050 speed control laboratory system (Fig. 4) is proposed in the following paragraphs. In simple words, the DC motor of this laboratory system drives a mass flywheel. The speed is measured inductively using a speed sensor. The power of the motor is controlled by the input voltage (0-5 V), while the speed sensor generates the voltage directly corresponding with the angular speed of the flywheel (0-10 V) [16]. The static characteristic of this system is shown in Fig. 5. Consequently, a set of normalized responses to step functions along various working points is depicted in Fig. 6 in order to show the system nonlinearity in dynamics.

A. Piecewise-linear neural model

To transform the nonlinear system described above into linear submodels according to the algorithm described in [9], it is necessary to design a neural model of the system, where

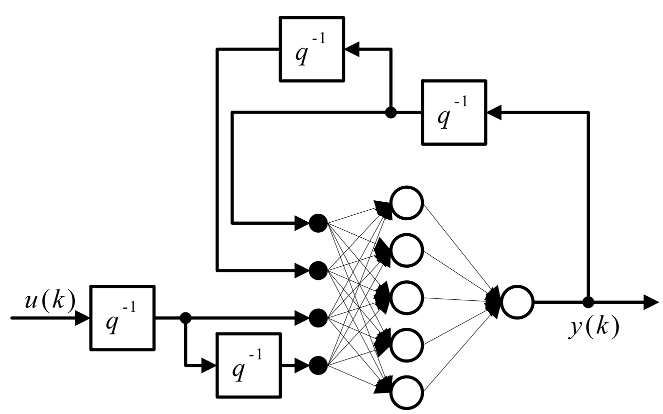

Fig. 7. Designed neural model.

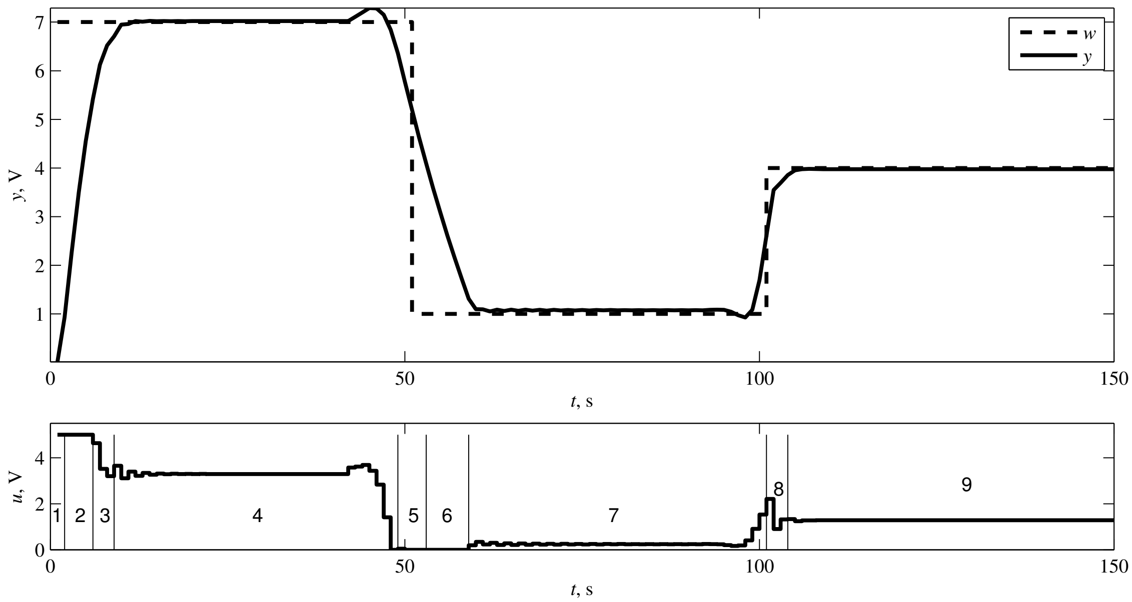

Fig. 8. Control response.

neurons in the hidden layer of the used feedforward neural network contain a linear saturated activation function and the output neuron contains a linear (identical) activation. The existence of this neural model is guaranteed by [17] and the process of a dynamic neural model design was standardized in many sources [18], [19], [20] or [21]. The process generally consists of training and testing set acquisition, neural network training and pruning, and neural model validating.

In this paper, the process of a neural model design is not described there since it is a generic procedure. As the end of this procedure, the neural model with the structure shown in Fig. 7 is designed. Thus, the neural network used inside this model consists of four inputs ($m = 2$, $n = 2$), five neurons with linear saturated activation functions in hidden layer, and one output neuron with linear (identical) activation function. The sampling rate is set to $T_S = 1$s.

B. Control experiments

As mentioned above, the piecewise-linear neural model provides the parameters a_1, a_2, b_1, b_2 and c of the linear submodel, which is currently valid. Using these parameters, the vector $\mathbf{l}$ and the matrix $\mathbf{F}_p$ of the predictive controller can be determined and the control action $u(k)$ can be computed. Apparently, the current state of the system as well as the future course of the reference variable should be known.

One control response is measured for control horizon $N = 10$ and control action change penalization $r = 0.2$. The courses are shown in Fig. 8. The system output y and its reference w are situated at the top part of the figure, while control action is at the bottom. In addition, the regions, in which particular linear models are used, are also marked and numbered. The numbering corresponds with Table I.

TABLE I
LINEAR SUBMODELS USED FOR PREDICTIVE CONTROLLER, SEE FIG 8

Number	a_1	a_2	b_1	b_2	c
1	0.0780	-0.1520	0.1720	0.0060	0.0450
2	-1.2900	0.4890	0.0530	0.0120	0.8680
3	-1.2980	0.3920	0.1100	0.0090	0.3220
4	-1.2980	0.3920	0.1100	0.0090	0.2610
5	-1.2190	0.2400	0.2820	0.0150	-0.3520
6	-1.3160	0.3720	0.3950	-0.0250	-0.1300
7	-1.3160	0.3720	0.3950	-0.0250	-0.0690
8	-1.2120	0.3370	0.2260	0.0180	0.2550
9	-1.2120	0.3370	0.2260	0.0180	0.1940

Considering Fig. 8, the control response is quick and smooth. Switching between linear submodels does not bring any significant disturbances. The small steady-state control error is caused by the combination of the non-ideal model of the system and missing integral part of the predictive controller.

Naturally, the control response can be tuned by the control action change penalization r. In Fig. 9, the control responses for several values of r are shown, keeping the other features unmodified.

V. CONCLUSION

As it is presented in this paper, a locally valid linear submodel provided by a piecewise-linear neural model can be used for control action evaluation without any customization. The family of predictive control techniques offers many tools these days to deal with various kinds of special and nonstandard systems and models. One particular predictive controller is developed in this paper. Its special feature is that it deals

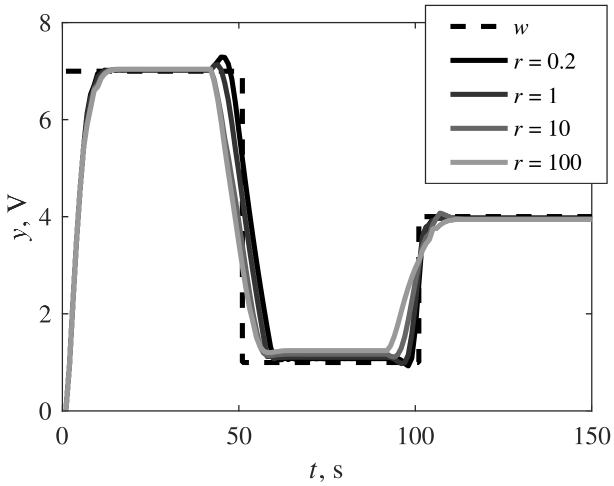

Fig. 9. Control responses for various values of q.

with a particular linear submodel provided by a piecewise-linear neural model in a very satisfactory way, as it is shown in illustrative example above. Note, that the point of the development was to keep the controller as simple as possible to keep the contribution synoptic. However, the controller can be improved on other interesting features such as an integral part, prediction error evaluation, constraint satisfaction, etc.

ACKNOWLEDGMENT

The work has been supported by the Funds of University of Pardubice, Czech Republic. This support is very gratefully acknowledged.

REFERENCES

[1] W. P. M. H. Heemels, B. D. Schutter, and A. Bemporad, "On the equivalence of classes of hybrid dynamical models," in *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No.01CH37228)*, vol. 1, 2001, pp. 364–369 vol.1.

[2] "PWLTool, a Matlab Toolbox for Piecewise Linear System," Department of Automatic Control, Lund Institute of Technology (LTH), Tech. Rep., 1999.

[3] R. Vidal, S. Soatto, Y. Ma, and S. Sastry, "An algebraic geometric approach to the identification of a class of linear hybrid systems," in *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 1, Dec 2003, pp. 167–172 Vol.1.

[4] G. Ferrari-Trecate, M. Muselli, D. Liberati, and M. Morari, "A clustering technique for the identification of piecewise affine systems," *Automatica*, 2003.

[5] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino, "Data classification and parameter estimation for the identification of piecewise affine models," in *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 1, Dec 2004, pp. 20–25 Vol.1.

[6] O. Ameur, P. Massioni, G. Scorletti, X. Brun, and M. Smaoui, "A piecewise-affine approach to the analysis of non-linear control laws for pneumatic systems," in *53rd IEEE Conference on Decision and Control*, 2014, pp. 3963–3969.

[7] Q. Sang and G. Tao, "Adaptive control of piecewise linear systems with output feedback for output tracking," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, 2012, pp. 5422–5427.

[8] N. Lin, X. Wang, Y. Zhang, C. Wang, and S. Diao, "Adaptive control design of uncertain piecewise-linear systems," in *Proceedings of 2012 UKACC International Conference on Control*, 2012, pp. 817–820.

[9] P. Dolezel and I. Taufer, "Piecewise-linear artificial neural networks for pid controller tuning," *Acta Montanistica Slovaca*, vol. 17, no. 3, pp. 224–233, 2012.

[10] P. Dolezel and P. Skrabanek, "Piecewise-linear neural network - possible tool for nonlinear process control," in *International Conference on Soft Computing MENDEL*, 2012, pp. 245–250.

[11] P. Dolezel, P. Rozsival, M. Mariska, and L. Havlicek, "Pid controller design for nonlinear oscillative plants using piecewise linear neural network," in *2013 International Conference on Process Control (PC)*, June 2013, pp. 19–24.

[12] P. Dolezel and L. Gago, "Piecewise linear neural network for process control in industrial environment," in *2016 17th International Carpathian Control Conference (ICCC)*, 2016, pp. 161–165.

[13] E. F. Camacho and C. A. Bordons, *Model Predictive Control in the Process Industry*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1997.

[14] J. Rossiter, *Model-Based Predictive Control: A Practical Approach*. London, UK: CRC Press, 2003.

[15] C. Wade, *Model Predictive Control: Theory, Practices and Future Challenges*. Hauppauge, New York, USA: Nova Science Publishers, 2015.

[16] D. Honc and F. Dusek, "Matlab/simulink support for GUNT control units," in *2013 International Conference on Process Control (PC)*, 2013, pp. 534–539.

[17] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[18] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New Jersey, USA: Prentice Hall, 1999.

[19] J. Korbicz and A. Janczak, "A neural network approach to identification of structural systems," in *Proceedings of IEEE International Symposium on Industrial Electronics*, vol. 1, 1996, pp. 98–103.

[20] K. S. Narendra and K. Parthasarathy, "Adaptive identification and control of dynamical systems using neural networks," in *Proceedings of the 28th IEEE Conference on Decision and Control,*, 1989, pp. 1737–1738.

[21] M. H. F. Rahiman, M. N. Taib, and Y. M. Salleh, "Black box modeling of steam distillation essential oil extraction system using nnarx structure," in *2009 First Asian Conference on Intelligent Information and Database Systems*, 2009, pp. 324–329.