Piecewise-linear artificial neural networks for PID controller tuning

Petr Doležel[1] *and* ***Ivan Taufer***[2]

A new algorithm of PID controller tuning is presented in this paper. It is well known that there have been introduced many techniques for PID controller tuning, both theoretical and experimental ones. However, this algorithm is suitable especially for highly nonlinear processes. It uses a model of the controlled process in the shape of piecewise-linear neural network which is linearized continuously and resulting linearized model is used for PID controller online tuning. While at the beginning of the paper the algorithm is described in theory, at the end there are mentioned some practical applications.

Keywords: *Process Control, PID Controller Tuning, Piecewise-Linear Neural Model*

Introduction

PID controller (which is an acronym to "proportional, integral and derivative") is a type of device used for process control. As first practical use of PID controller dates to 1890s (Bennet 1993), PID controllers are spread widely in various control applications till these days. In process control today, more than 95% of the control loops are PID type (Astrom and Hagglund 1995). PID controllers have experienced many changes in technology, from mechanics and pneumatics to microprocessors and computers. Especially microprocessors have influenced PID controllers applying significantly. They have given possibilities to provide additional features like automatic tuning or continuous adaptation - and continuous adaptation of PID controller via neural model of controlled system (which is considered to be significantly nonlinear) is the aim of this contribution.

PID controllers

The basic structure of conventional feedback control using PID controller is shown in Fig. 1 (Astrom and Hagglund 1995)(Doyle et al. 1990). In this figure, the PLANT is the object to be controlled. The aim of the control is to make controlled system output variable y_S follow the set-point r using the manipulated variable u changes. Variable e is control error and is considered as PID controller input and t is continuous time.

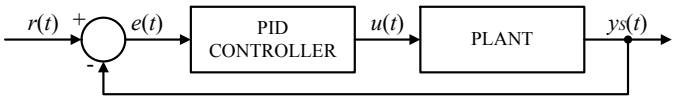

Fig. 1. Conventional feedback control loop.

Continuous-time PID controller itself is defined by several different algorithms (Astrom and Hagglund 1995)(Doyle et al. 1990). Let us use the common version defined by Eq. (1).

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau)\,\mathrm{d}\tau + T_d \frac{\mathrm{d}e(t)}{\mathrm{d}t} \right) \tag{1}$$

The control variable is a sum of three parts: proportional one, integral one and derivative one. The controller parameters are proportional gain K_p, integral time T_i and derivative time T_d. In applications, all three parameters have to be tuned to solve certain problem most appropriately while both stability and quality of control performance are satisfied. As microprocessors started to set widely in all branches of industry, discrete form of PID controller was determined. Discrete PID controller computes output signal only at discrete time instants kT (where T is sapling interval and k is an integer). Thus, conventional control loop (Fig. 1) has to be upgraded with zero order hold (ZOH), analogue-digital converter (A/D) and digital-analogue converter (D/A) - see Fig. 2 (kT is replaced by k for formal simplification).

[1] *Ing. Petr Doležel, Ph.D.*, University of Pardubice, Faculty of Electrical Engineering and Informatics, petr.dolezel@upce.cz

[2] *Prof. Ing. Ivan Taufer, DrSc.*, University of Pardubice, Faculty of Electrical Engineering and Informatics, ivan.taufer@upce.cz

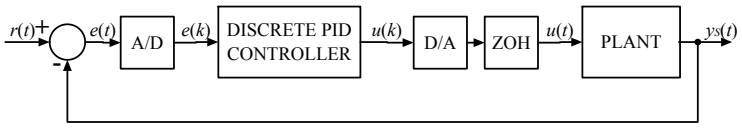

Fig. 2. Discrete feedback control loop.

Formula of discrete PID controller can be obtained by discretizing of Eq. (1). From a purely numerical point of view, integral part of controller can be approximated by sum of control errors and derivative part by difference of two consecutive control errors - see Eq. (2).

$$u(k) = K_p \left(e(k) + \frac{T}{T_i} \sum_{i=1}^{k} \frac{e(i) + e(i-1)}{2} + T_d(e(k) - e(k-1)) \right) \tag{2}$$

For practical application, incremental form of discrete controller is more suitable. Increment of manipulated variable is defined by difference of two consecutive values. Using Eq. (2), we can express

$$u(k) - u(k-1) = q_0 e(k) + q_1 e(k-1) + q_2 e(k-2) \tag{3}$$

Parameters q_0 ... q_2 depend on original values of K_p, T_i, T_d and sampling interval T.

PID controller Tuning

The problem of PID controller tuning is solved often either using the Laplace Transform (continuous-time) or the Z Transform (discrete-time) (Bracewell 2000). However, there are some experimental methods, e.g. Nichols-Ziegler technique (Ziegler and Nichols 1942) which uses ultimate gain of proportional controller, or Cohen-Coon technique (Cohen and Coon 1954) which uses step test of controlled process. For this contribution, polynomial approach of discrete PID controller tuning is important. In the Z domain (Isermann 1991), discrete PID controller has the following transfer function.

$$\frac{Q(z^{-1})}{P(z^{-1})} = \frac{q_0 + q_1 z^{-1} + q_2 z^{-2}}{1 - z^{-1}} \tag{4}$$

where z is complex variable. For a bit more complicated systems, discrete PID controller with integrated filter (5) is sometimes used.

$$\frac{Q(z^{-1})}{P(z^{-1})} = \frac{q_0 + q_1 z^{-1} + q_2 z^{-2}}{(1 - z^{-1})(1 + \gamma z^{-1})} \tag{5}$$

Then, suppose Z domain conventional feedback control loop with discrete PID controller (4) and linear controlled system described by numerator $B(z^{-1})$ and denominator $A(z^{-1})$ - see Fig. 3.

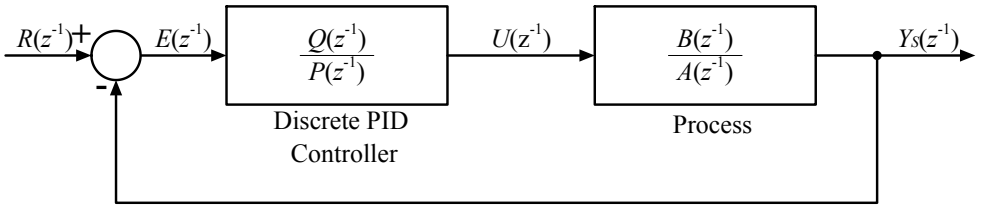

Fig. 3. Feedback control loop with discrete PID controller in Z domain.

Z transfer function of closed loop in Fig. 3 is

$$\frac{Y_S(z^{-1})}{R(z^{-1})} = \frac{B(z^{-1})Q(z^{-1})}{A(z^{-1})P(z^{-1}) + B(z^{-1})Q(z^{-1})} \tag{6}$$

Denominator of Z transfer function (6) is the characteristic polynomial

$$D(z^{-1}) = A(z^{-1})P(z^{-1}) + B(z^{-1})Q(z^{-1}) \tag{7}$$

It is well known that dynamics of the closed loop is defined by the characteristic polynomial (7) (Isermann 1991). The polynomial has three tuneable variables which are PID controller parameters q_0, q_1, q_2 (and γ for more complex systems). The roots of the polynomial (7) are responsible for control dynamics and one can assign those roots (so called poles) (see Fig. 4) by suitable tuning of the parameters q_0, q_1, q_2.

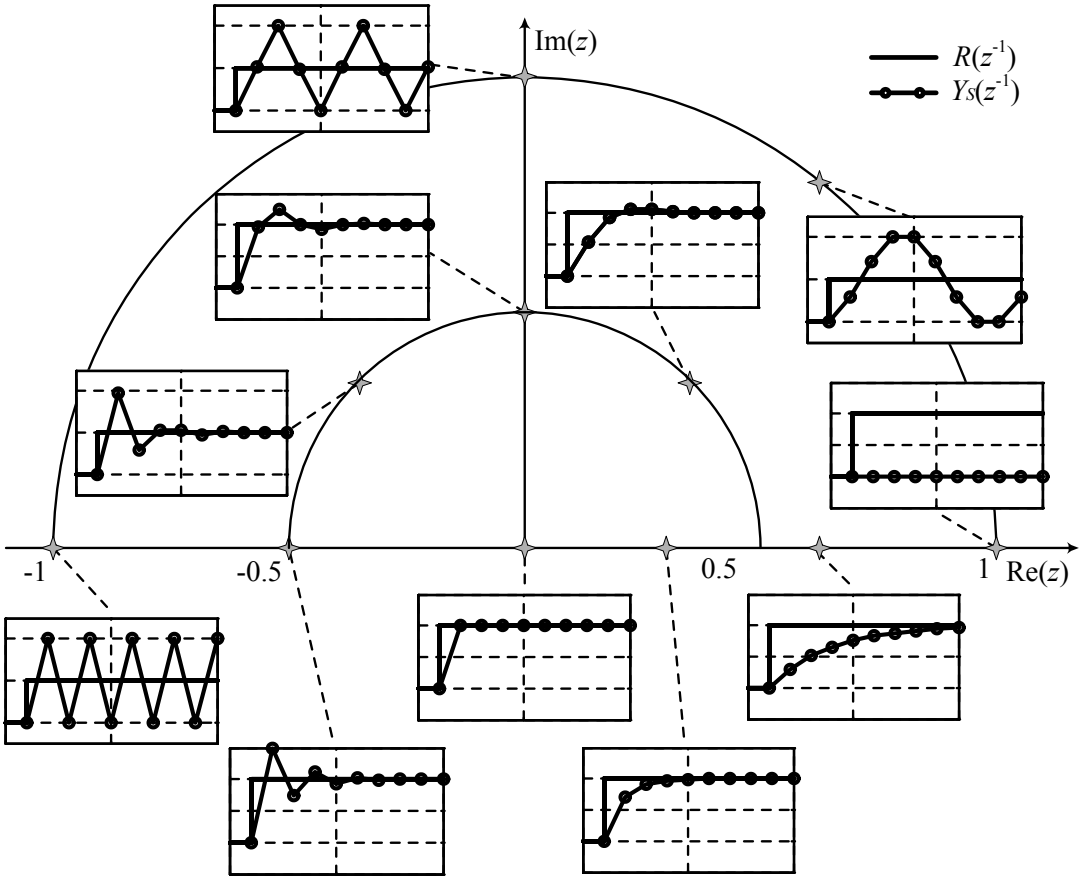

Fig. 4. The effect of characteristic polynomial poles to the control dynamics.

Thus, discrete PID controller tuning using pole assignment means to choose desired control dynamics (desired definition of characteristic polynomial) and to compute subsequently discrete PID controller parameters. The only eligible parameter is polynomial $D(z^{-1})$ (see Eq. 8) which is to be chosen. According to (Hunt 1993), there are four ways to do it.

$$D(z^{-1}) = 1 + \sum_i d_i z^{-i} \tag{8}$$

1. Dead beat is achieved
2. Quadratic criterion is satisfied
3. Control dynamics of closed loop equals to dynamics of defined second order system
4. Special dynamics of closed control loop (defined by customer) is achieved

Then, following Diophantine equation is to be solved.

$$1 + \sum_i d_i z^{-i} = A(z^{-1})P(z^{-1}) + B(z^{-1})Q(z^{-1}) \tag{9}$$

If any solution exists, it provides us expected set of controller parameters. Comprehensive foundation to pole assignment technique is described in (Hunt 1993).

Continuous linearization using piecewise-linear artificial neural network

The tuning technique described in previous section requires linear model of controlled system in form of Z transfer function. If controlled system is highly nonlinear process, linear model has to be updated continuously (online) with operating point shifting. Except some classical techniques of continuous linearization (Gain Scheduling, Recurrent Least Squares Method, ...), there has been introduced new technique (Dolezel et al. 2011), recently. It is presented in next paragraphs.

Artificial neural network for approximation

According to Kolmogorov's superposition theorem, any real continuous multidimensional function can be evaluated by sum of real continuous one-dimensional functions (Hecht-Nielsen 1987). If the theorem is applied to artificial neural network (ANN), it can be said that any real continuous multidimensional function can be approximated by certain three-layered feedforward ANN with arbitrary precision. Topology of that ANN is depictured in Fig. 5. Input layer brings external inputs $x_1, x_2, ..., x_P$ into ANN. Hidden layer contains S neurons, which process sums of weighted inputs using continuous, bounded and monotonic activation function. Output layer contains one neuron, which processes sum of weighted outputs from hidden neurons. Its activation function has to be continuous and monotonic.

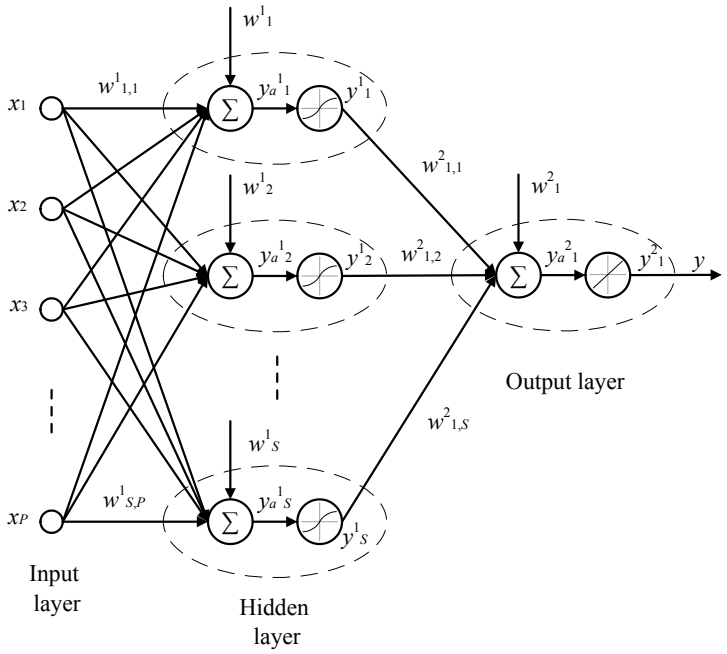

Fig. 5. Three-layered ANN.

So ANN in Fig. 5 takes P inputs, which are processed by S neurons in hidden layer and then by one output neuron. Dataflow between input i and hidden neuron j is gained by weight $w^1_{j,i}$. Dataflow between hidden neuron j and output neuron is gained by weight $w^2_{1,j}$. Output of the network can be expressed by following equations.

$$ya^1_j = \sum_{i=1}^{P} w^1_{j,i}\, x_i + w^1_j \tag{10}$$

$$y^1_j = \phi^1\left(ya^1_j\right) \tag{11}$$

$$ya^2_1 = \sum_{j=1}^{S} w^1_{1,j}\, y^1_j + w^2_1 \tag{12}$$

$$y = \phi^2\left(y_a^2{}_1\right) \tag{13}$$

In equations above, $\phi^1(.)$ means activation functions of hidden neurons and $\phi^2(.)$ means output neuron activation function. The network should be well trained to achieve sufficient approximation qualities. In other words, the network is to learn associations between a specified set of input-output pairs (training set). As there were presented many training techniques from simple Back-propagation Algorithm (Rumelhart et al. 1986)(Haykin 1999) to some specialized hybrid techniques using evolutionary algorithms (Montana and Davis 1989), they are not defined here.

System identification by ANN

System identification means especially a procedure which leads to dynamic model of the system. ANN is used widely in system identification because of its outstanding approximation qualities. There are several ways to use ANN for system identification. One of them assumes that the system to be identified (with input u and output y_S) is determined by the following nonlinear discrete-time difference equation.

$$y_S(k) = \psi\left[y_S(k-1), ...y_S(k-n), u(k-1), u(k-m)\right] \tag{14}$$

In equation (14), $\psi(.)$ is nonlinear function, k is discrete time (formally better would be kT) and n is difference equation order. The aim of the identification is to design ANN which approximates nonlinear function $\psi(.)$. Then, neural model can be expressed by Eq. (15).

$$y_M(k) = \hat{\psi}\left[y_M(k-1), ...y_M(k-n), u(k-1), u(k-m)\right] \tag{15}$$

$\hat{\psi}$ represents well trained ANN and y_M is its output. Formal diagram of neural model is shown in Fig. 6. It is obvious that ANN in Fig. 6 has to be trained to provide y_M as close to y_S as possible. Existence of such a neural network is guaranteed by Kolmogorov's superposition theorem and the whole modelling procedure has been extensively discussed in (Haykin 1999) or (Nguyen et al. 2003), so it is not dealt with here.

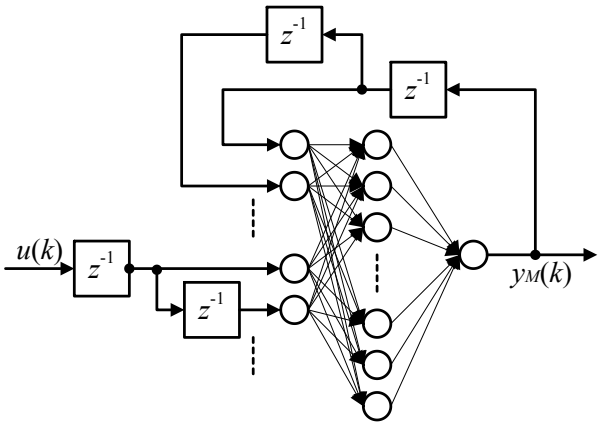

Fig. 6. Formal diagram of neural model.

Piecewise-linear neural model for discrete PID controller tuning

As mentioned in section above, there are some conditions to be held, so that ANN acquires universal approximation qualities. Above all, continuous, bounded and monotonic activation function should be used in hidden neurons, continuous and monotonic activation function then in output neuron. To satisfy those conditions, there is used mostly hyperbolic tangent activation function (16) for neurons in hidden layer and identical activation function (17) for output neuron.

$$y^1{}_j = \tanh\left(y_a^1{}_j\right) \tag{16}$$

$$y = y_{a\,1}^{2} \tag{17}$$

However, if linear saturated activation function (18) is used instead, ANN features stay similar because of resembling courses of both activation functions (see Fig. 7).

$$y^{1}{}_{j} = \begin{cases} 1 & for \quad y_{a}{}^{1}{}_{j} > 1 \\ y_{a}{}^{1}{}_{j} & for \quad -1 \leq y_{a}{}^{1}{}_{j} \leq 1 \\ -1 & for \quad y_{a}{}^{1}{}_{j} < 1 \end{cases} \tag{18}$$

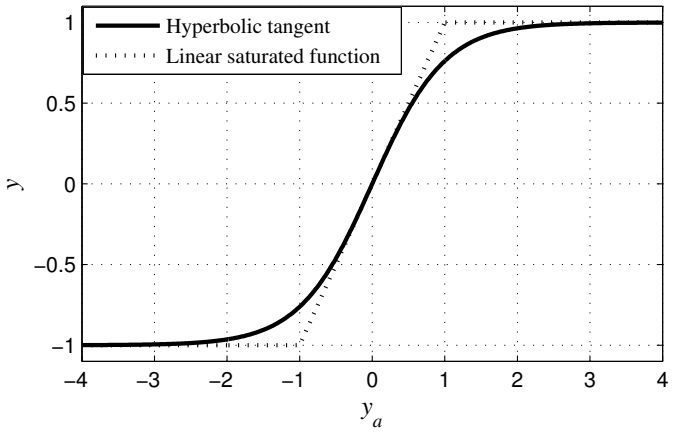

Fig. 7. Activation functions comparison.

The output of linear saturated activation function is either constant or equal to input so neural model which uses ANN with linear saturated activation functions in hidden neurons acts as piecewise-linear model. One linear submodel turns to another when any hidden neuron becomes saturated or becomes not saturated. Let us presume an existence of a dynamical neural model which uses ANN with linear saturated activation functions in hidden neurons and identic activation function in output neuron - see Fig. 8.

ANN output can be computed using Eqs. (10) - (13). However, another way for ANN output computing is useful. Let us define saturation vector **V** of S elements. This vector indicates saturation states of hidden neurons - see Eq. (19).

$$v_j = \begin{cases} 1 & for \quad y^{1}{}_{j} = 1 \\ 0 & for \quad -1 < y^{1}{}_{j} < 1 \\ -1 & for \quad y^{1}{}_{j} = -1 \end{cases} \tag{19}$$

Now, ANN output can be expressed by

$$y_M(k) = -\sum_{j=1}^{n} a_j y_M(k-j) + \sum_{j=1}^{m} b_j u(k-j) + c \tag{20}$$

where

$$a_j = -\sum_{i=1}^{S} w^{2}{}_{1,i}(1-|v_i|)w^{1}{}_{i,j} \tag{21}$$

$$b_j = \sum_{i=1}^{S} w^{2}{}_{1,i}(1-|v_i|)w^{1}{}_{i,j+n} \tag{22}$$

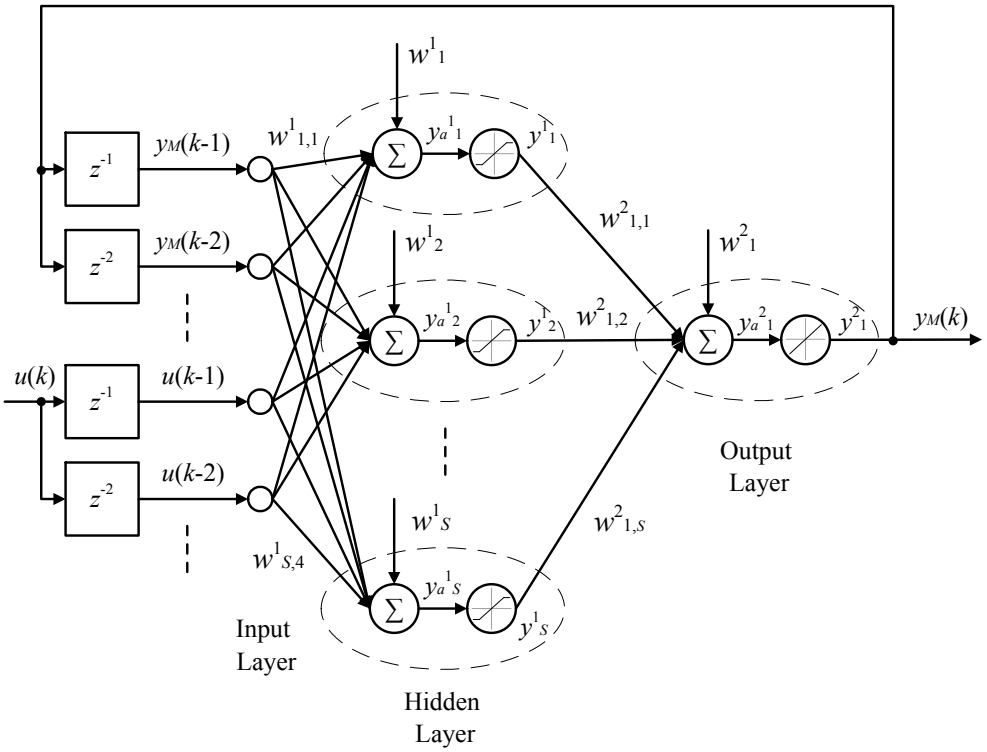

Fig. 8. Piecewise-linear neural model.

$$c = w^2_1 + \sum_{i=1}^{S} \left(w^2_{1,i} v_i + (1 - |v_i|) w^2_{1,i} w^1_i \right) \tag{23}$$

Thus, difference equation (20) defines ANN output and it is linear in some neighbourhood of actual state (in that neighbourhood, where saturation vector **V** stays constant).

In other words, if the neural model of any nonlinear system in form of Fig. 8 is designed, then it is simple to determine parameters of linear difference equation which approximates system behaviour in some neighbourhood of actual state. This difference equation can be used then to the actual control action setting due to many of classical or modern control techniques.

In following examples, discrete PID controller with parameters tuned according to algorithm introduced in paragraph 3 is studied. As it is mentioned above, controlled system discrete model in form of Z transfer function is required. So first, difference equation (20) should be transformed in following way. Let us define

$$\tilde{u}(k) = u(k) - u_0 \tag{24}$$

where u_0 is constant. Then, Eq. (20) turns into

$$y_M(k) = -\sum_{j=1}^{n} a_j y_M(k-j) + \sum_{j=1}^{m} b_j \tilde{u}(k-j) + c + u_0 \sum_{j=1}^{m} b_j \tag{25}$$

Equation (25) becomes constant term free, if Eq. (26) is satisfied.

$$u_0 = -\frac{c}{\sum_{j=1}^{m} b_j} \tag{26}$$

In Z domain, model (25) witch respect to Eq. (26) is defined by Z transfer function

$$\frac{Y_M(z^{-1})}{\tilde{U}(z^{-1})} = \frac{\sum_{j=1}^{m} b_j z^{-j}}{1 + \sum_{j=1}^{m} a_j z^{-j}} \tag{27}$$

Algorithm of discrete PID controller tuning using piecewise-linear neural network

Whole algorithm of piecewise-linear neural model usage in PID controller parameters tuning is summarized in following terms (see Fig. 9, too).

1. Create neural model of controlled plant in form of Fig. 8
2. Determine polynomial $D(z^{-1})$
3. Set $k = 0$
4. Measure system output $y_S(k)$
5. Determine the parameters a_i, b_i and c of difference equation (20)
6. Transform Eq. (20) into Z transfer function (27)
7. Determine discrete PID controller parameters by solving of Eq. (9) where $A(z^{-1})$ and $B(z^{-1})$ are denominator and nominator of Z transfer function (27), respectively
8. Determine $\tilde{u}(k)$ using discrete PID controller tuned in previous step
9. Transform $\tilde{u}(k)$ into $u(k)$ using Eq. (24) and perform control action
10. $k = k + 1$, go to 4

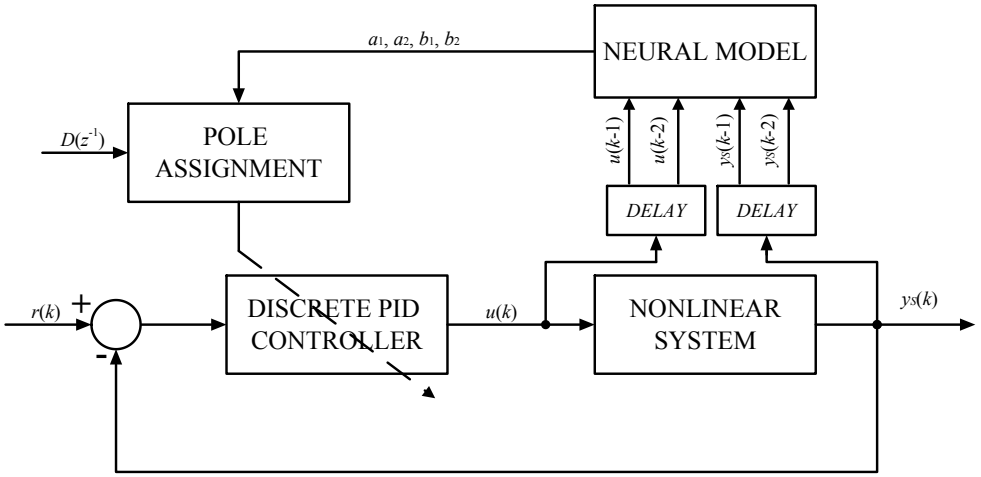

Fig. 9. Control algorithm diagram for second order nonlinear system (m = n = 2).

Case study

The algorithm introduced above is applied now to control of simulated nonlinear system compiled by a combination of nonlinear static part and linear dynamical system - see Fig. 10.

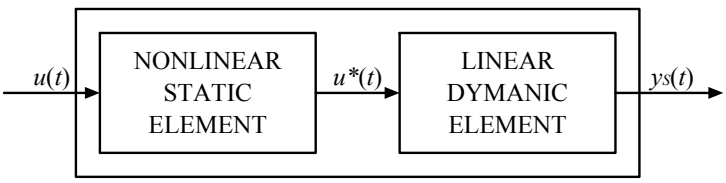

Fig. 10. System to be controlled.

The static element is defined by Eq. (28), while the dynamic system by (29).

$$u^* = \tanh(u^3) \tag{28}$$

$$y_S(t) + 5\frac{\mathrm{d}y(t)}{\mathrm{d}t} + 50\frac{\mathrm{d}^2y(t)}{\mathrm{d}t^2} = u^*(t) \tag{29}$$

Graphic characteristics of the system are shown in Fig. 11.

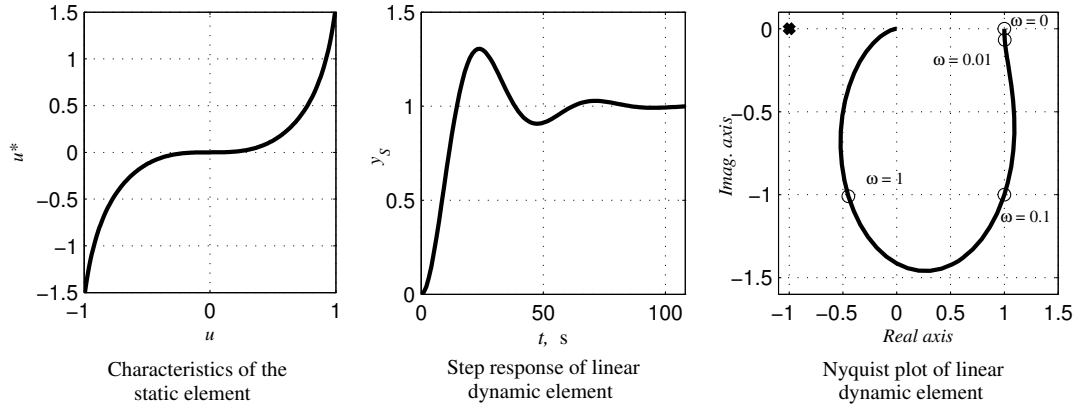

Fig. 11. Graphic characteristics of the system.

System input is limited to an interval $u \in < -1; 1 >$. Control loop is designed as shown in paragraph 5. At first, dynamical piecewise-linear neural model in shape of Fig. 8. is created. This procedure involves training and testing set acquisition, neural network training and pruning and neural model validating. As this sequence of processes is illustrated closely in many other publications (Nguyen et al. 2003)(Haykin 1999), it is not referred here.

Next step is to decide polynomial $D(z^{-1})$. In this example, two second order systems (one conservative, the other more aggressive one) are defined as standards for closed loop dynamics to determine polynomial $D(z^{-1})$. In other words, the point is to force some defined performance to the closed loop - Fig. 12.

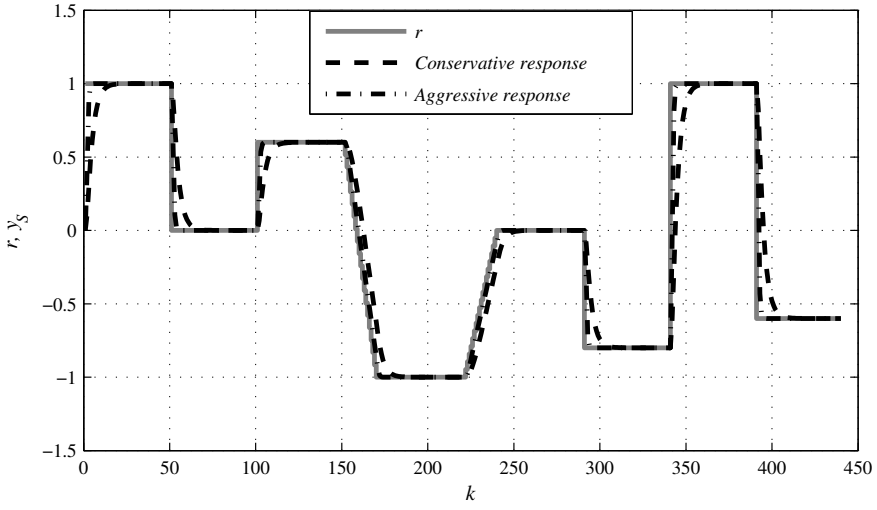

Fig. 12. Desired dynamics of the closed loop.

However, actual control responses are a little different (especially the aggressive one) - see Fig. 13. The aggressive response follows set point similarly to the conservative one and it involves some oscillations, in addition. The differences are caused by not ideally precise neural model, system input boundaries and mainly by simple fact, that one cannot expect to force such a smooth dynamics to complex nonlinear oscillative system. In other words, standard dynamics defined by polynomial $D(z^{-1})$ should be feasible considering complexity of controlled system.

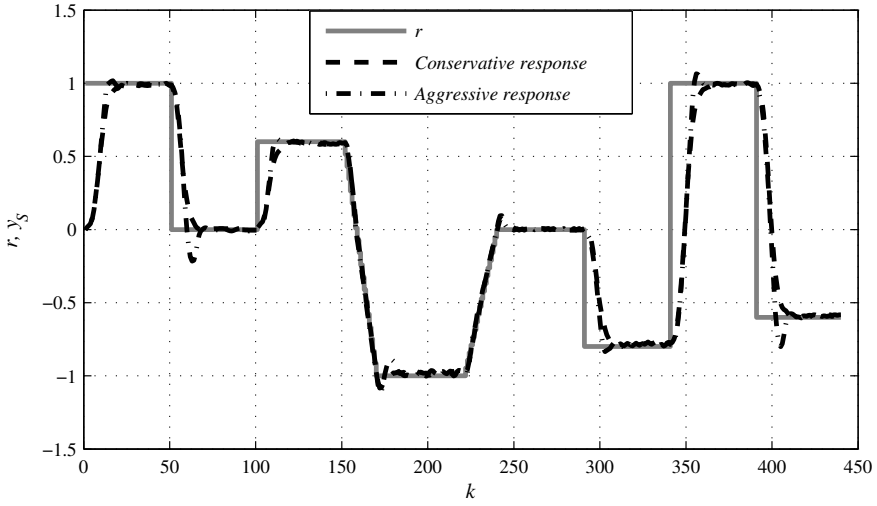

Fig. 13. Control responses.

Conclusion

The new algorithm of PID controller online tuning is introduced and theoretically described in the paper. It is suitable especially for highly nonlinear processes control, as it is shown in case study. The algorithm uses piecewise-linear neural model as plant model, which is great advantage, because the linearization of such a model is not computationally demanding. Thus, linearization and controller parameters adaptation can be easily performed every time sample.

Acknowledgement

The work has been supported by the funds No. SGFEI03/2012 of the University of Pardubice, Czech Republic. This support is very gratefully acknowledged.

References

Astrom, K.J., Hagglund, T., 1995. PID controllers: theory, design and tuning. International Society for Measurement and Control. ISBN: 1556175167.

Bennet, S., 1993. A history of control engineering, 1930-1955. Institution of Engineering and Technology. ISBN: 0863410472.

Bracewell, R.N., 2000. The Fourier Transform and Its Applications (3rd ed.). McGraw-Hill. ISBN: 0073039381.

Cohen, G.H., Coon, G.A., 1954. Theoretical consideration of retarded control. Trans. of the American Society of Mechanical Engineers .

Dolezel, P., Taufer, I., Mares, J., 2011. Piecewise-linear neural models for process control, in: Proceedings of the 18th International Conference on Process Control '11.

Doyle, J.C., Francis, B.A., Tannenbaum, A.R., 1990. Feedback Control Theory. Macmillan Publishing. ISBN: 0486469336.

Haykin, S., 1999. Neural Networks: A Comprehensive Foundation. Prentice Hall. ISBN: 0132733501.

Hecht-Nielsen, R., 1987. Kolmogorov's mapping neural network existence theorem, in: Proc. I987 IEEE International Conference on Neural Networks.

Hunt, K.J., 1993. Polynomial methods in optimal control and filtering. Peter Peregrinus Ltd. ISBN: 3540107282.

Isermann, R., 1991. Digital Control Systems. Springer-Verlag. ISBN: 0863412955.

Montana, D.J., Davis, L., 1989. Training feedforward neural networks using genetic algorithms, in: Proceeding of the Eleventh (1989) International Joint Conference on Artificial Intelligence.

Nguyen, H., Prasad, N., Walker, C., 2003. A First Course in Fuzzy and Neural Control. Chapman and Hall/CRC. ISBN: 1584882441.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. Nature .

Ziegler, J.G., Nichols, N.B., 1942. Optimum settings for automatic controllers. Trans. of the American Society of Mechanical Engineers .