

Discrete PID Controller Tuning Using Piecewise-Linear Neural Network

Petr Doležel, Ivan Taufer and Jan Mareš
*University of Pardubice & Institute of Chemical Technology Prague
 Czech Republic*

1. 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 (Bennett, 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 et al., 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. Artificial Neural Networks have traditionally enjoyed considerable attention in process control applications, especially for their universal approximation abilities (Montague et al., 1994), (Dwarapudi, et al., 2007). In next sections, there is to be explained how to use artificial neural networks with piecewise-linear activation functions in hidden layer in controller design. To be more specific, there is described technique of controlled plant linearization using nonlinear neural model. Obtained linearized model is in a shape of linear difference equation and it can be used for PID controller parameters tuning.

2. Continuous-time and discrete PID controller

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

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

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \quad (1)$$

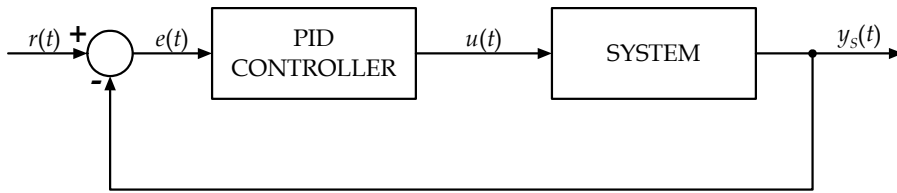


Fig. 1. Conventional feedback control loop

The control variable is a sum of three parts: proportional one, integral one and derivative one - see Fig. 2. The controller parameters are proportional gain K_p , integral time T_i and derivative time T_d .

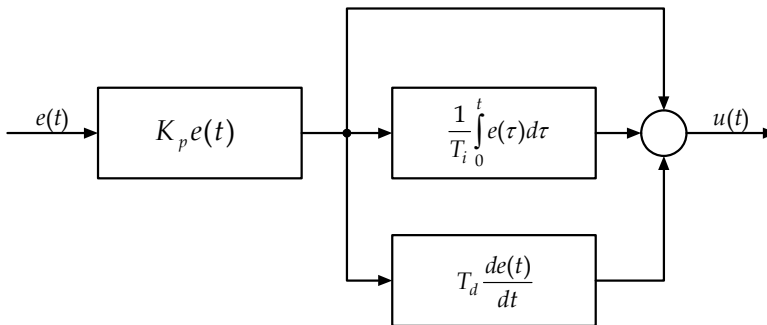


Fig. 2. Continuous-time PID controller

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. Many tuning techniques have been published in recent decades, some of them experimental, the others theoretically based.

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 $k \cdot T$ (where T is sampling 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. 3 ($k \cdot T$ is replaced by k for formal simplification).

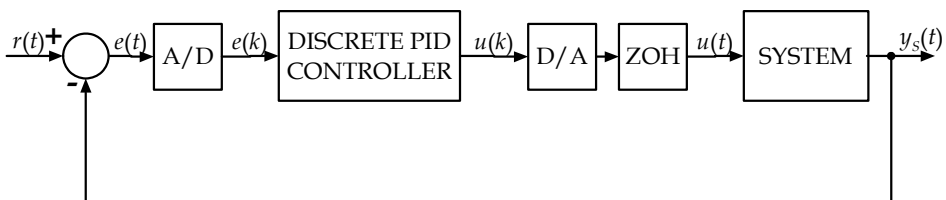


Fig. 3. Feedback control loop with discrete PID controller

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 (Eq. 2) and derivative part by (Eq. 3).

$$\int_0^t e(\tau) d\tau \approx T \sum_{i=1}^k \frac{e(i) + e(i-1)}{2} \quad (2)$$

$$\frac{de(t)}{dt} \approx \frac{e(k) - e(k-1)}{T} \quad (3)$$

Then, discrete PID controller is defined by (Eq. 4).

$$u(k) = K_p \left(e(k) + \frac{T}{T_i} \sum_{i=1}^k \frac{e(i) + e(i-1)}{2} + \frac{T_d}{T} (e(k) - e(k-1)) \right) \quad (4)$$

For practical application, incremental form of discrete controller is more suitable. Let us assume

$$\Delta u(k) = u(k) - u(k-1) \quad (5)$$

Then, with respect to (Eq. 4)

$$u(k) - u(k-1) = q_0 e(k) + q_1 e(k-1) + q_2 e(k-2) \quad (6)$$

where

$$q_0 = K_p \left(1 + \frac{T}{2T_i} + \frac{T_d}{T} \right)$$

$$q_1 = -K_p \left(1 - \frac{T}{2T_i} + \frac{2T_d}{T} \right)$$

$$q_2 = K_p \frac{T_d}{T}$$

In the Z domain (Isermann, 1991), discrete PID controller has the following transfer function.

$$\frac{Q(z^{-1})}{P(z^{-1})} = \frac{q_0 + q_1 z^{-1} + q_2 z^{-2}}{1 - z^{-1}} \quad (7)$$

As well as for continuous-time PID controller, there have been introduced several methods for q_0 , q_1 , q_2 tuning (Isermann, 1991). Most of them require mathematical model of controlled system (either first principle or experimental one) and if the system is nonlinear, the model has to be linearized around one or several operating points.

In next paragraph, the way how to tune discrete PID controller using Pole Assignment technique is described.

3. Discrete PID controller tuning using Pole Assignment technique

Suppose conventional feedback control loop with discrete PID controller (7) and controlled system described by nominator $B(z^{-1})$ and denominator $A(z^{-1})$ – see Fig. 4.

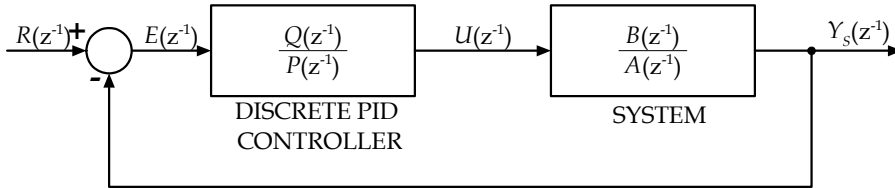


Fig. 4. Feedback control loop with discrete PID controller

Then, Z - transfer function of closed control loop is

$$\frac{Y(z^{-1})}{R(z^{-1})} = \frac{B(z^{-1})Q(z^{-1})}{A(z^{-1})P(z^{-1}) + B(z^{-1})Q(z^{-1})} \quad (8)$$

Denominator of Z - transfer function (8) is the characteristic polynomial

$$D(z^{-1}) = A(z^{-1})P(z^{-1}) + B(z^{-1})Q(z^{-1}) \quad (9)$$

It is well known that dynamics of the closed loop behaviour is defined by the characteristic polynomial (9). It has three tuneable variables which are PID controller parameters q_0, q_1, q_2 . The roots of the polynomial (9) are responsible for control dynamics and one can assign those roots (so called poles) (see Fig. 5) by suitable tuning of the parameters q_0, q_1, q_2 . Thus, discrete PID controller tuning using Pole Assignment means choosing desired control dynamics (desired definition of characteristic polynomial) and subsequent computing of discrete PID controller parameters.

Let us show an example: suppose we need control dynamics defined by characteristic polynomial (10), where d_1, d_2, \dots are integers (there are many ways how to choose those parameters, one of them is introduced in the case study at the end of this contribution).

$$D(z^{-1}) = 1 + d_1z^{-1} + d_2z^{-2} + \dots \quad (10)$$

So we have to solve Diophantine equation (11) to obtain all controller parameters.

$$1 + d_1z^{-1} + d_2z^{-2} + \dots = A(z^{-1})P(z^{-1}) + B(z^{-1})Q(z^{-1}) \quad (11)$$

If any solution exists, it provides us expected set of controller parameters.

Comprehensive foundation to pole assignment technique is described in (Hunt, 1993).

4. Continuous linearization using artificial neural network

The tuning technique described in section 3 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 with operating point shifting. Except some classical techniques of continuous linearization (Gain Scheduling, Recurrent Least Squares Method, ...), there has

been introduced new technique (Doležel et al., 2011), recently. It is presented in next paragraphs.

4.1 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 ANN with arbitrary precision. Topology of that ANN is depicted in Fig. 6. Input layer brings external inputs x_1, x_2, \dots, 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.

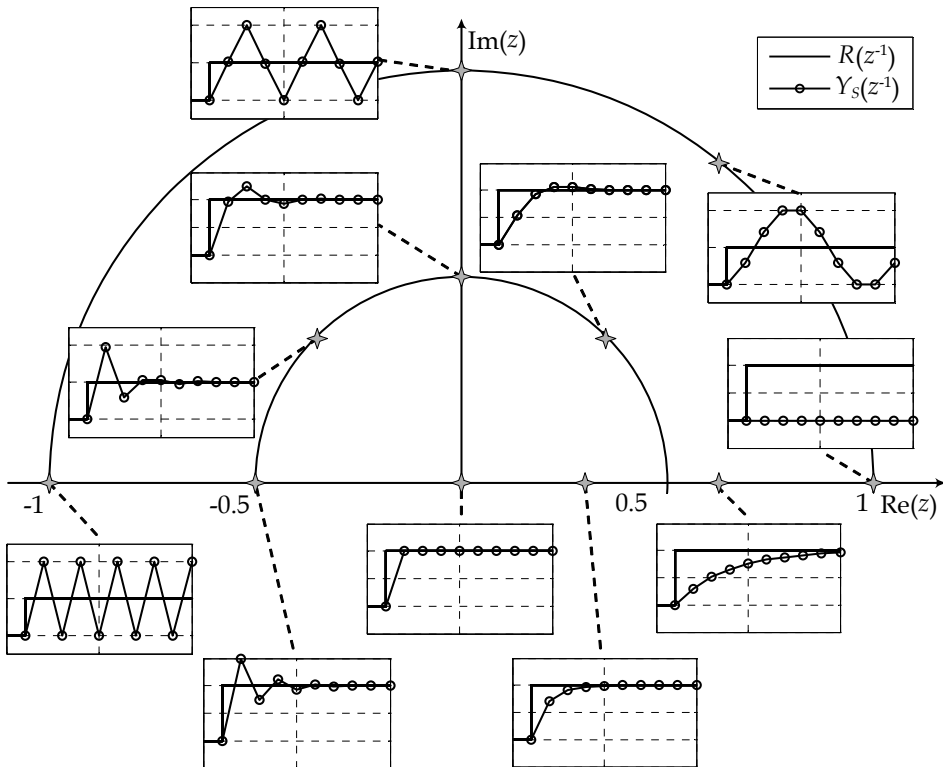


Fig. 5. The effect of characteristic polynomial poles to the control dynamics

So ANN in Fig. 6 takes P inputs, those inputs 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 k and output neuron is gained by weight $w^2_{1,k}$. Output of the network can be expressed by following equations.

$$y_{a^1_j} = \sum_{i=1}^p w^1_{j,i} \cdot x_i + w^1_j \tag{12}$$

$$y^1_j = \varphi^1(y_{a^1_j}) \tag{13}$$

$$y_{a^2_1} = \sum_{i=1}^s w^2_{1,i} \cdot y^1_i + w^2_1 \tag{14}$$

$$y = \varphi^2(y_{a^2_1}) \tag{15}$$

In equations above, $\varphi^1(\cdot)$ means activation functions of hidden neurons and $\varphi^2(\cdot)$ means output neuron activation function.

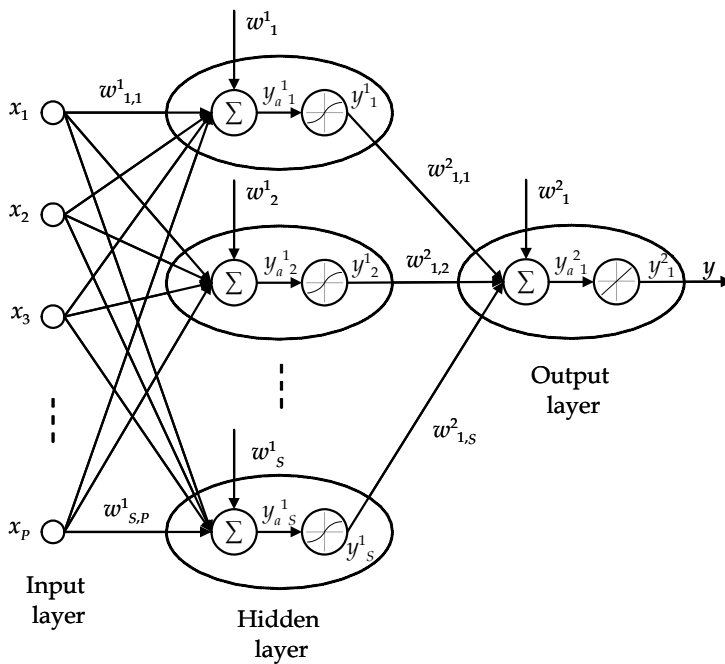


Fig. 6. Three-layered ANN

As it has been mentioned, there are some conditions applicable for activation functions. To satisfy those conditions, there is used mostly hyperbolic tangent activation function (Eq. 16) for neurons in hidden layer and identical activation function (Eq. 17) for output neuron.

$$y^1_j = \tanh(y_{a^1_j}) \tag{16}$$

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

Mentioned theorem does not define how to set number of hidden neurons or how to tune weights. However, there have been published many papers which are focused especially on gradient training methods (Back-Propagation Gradient Descend Alg.) or derived methods (Levenberg-Marquardt Alg.) – see (Haykin, 1994).

4.2 System identification by artificial neural network

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.

$$y_s(k) = \psi[y_s(k-1), \dots, y_s(k-n), u(k-1), \dots, u(k-m)], m \leq n \tag{18}$$

In equation (18), $\psi(\cdot)$ is nonlinear function, k is discrete time (formally better would be $k \cdot T$) and n is difference equation order.

The aim of the identification is to design ANN which approximates nonlinear function $\psi(\cdot)$. Then, neural model can be expressed by (eq. 19).

$$y_M(k) = \hat{\psi}[y_M(k-1), \dots, y_M(k-n), u(k-1), \dots, u(k-m)], m \leq n \tag{19}$$

In (Eq. 19), $\hat{\psi}$ represents well trained ANN and y_M is its output. Formal scheme of neural model is shown in Fig. 7. It is obvious that ANN in Fig. 7 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 whole process of neural model design is described in detail in (Haykin, 1994) or (Nguyen et al., 2003).

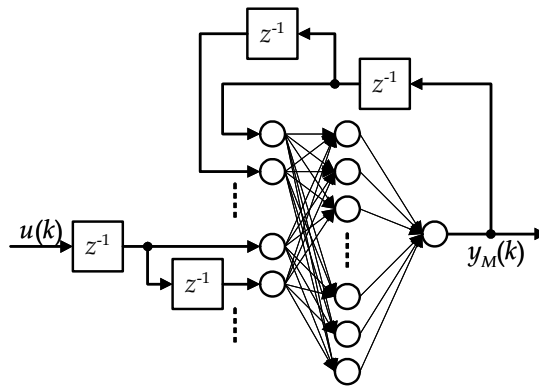


Fig. 7. Formal scheme of neural model

4.3 Piecewise-linear neural model for discrete PID controller tuning

As mentioned in section 4.1, there is recommended to use hyperbolic tangent activation function for neurons in hidden layer and identical activation function for output neuron in

ANN used in neural model. However, if linear saturated activation function (Eq. 20) is used instead, ANN features stay similar because of resembling courses of both activation functions (see Fig. 8).

$$y^1_j = \begin{cases} 1 & \text{for } y_a^1_j > 1 \\ y_a^1_j & \text{for } -1 \leq y_a^1_j \leq 1 \\ -1 & \text{for } y_a^1_j < -1 \end{cases} \quad (20)$$

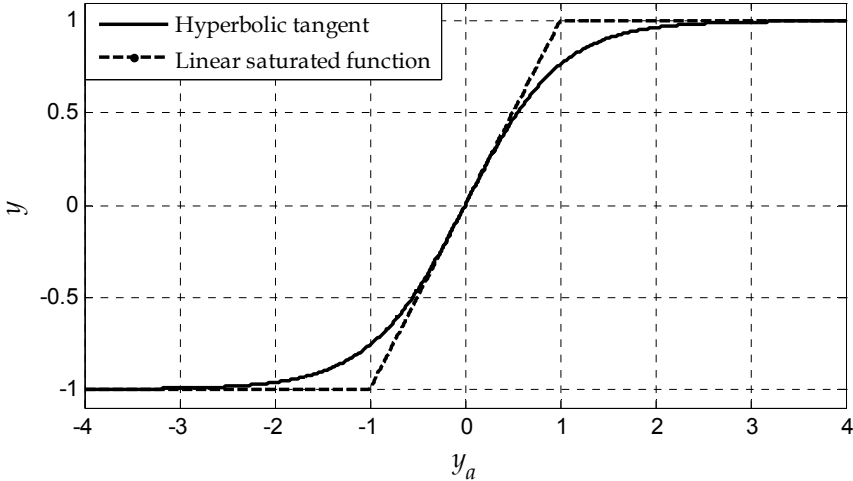


Fig. 8. Activation functions comparison

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. 9. Let us also presume $m = n = 2$ for making process plainer. ANN output can be computed using Eqs. (12), (13), (14), (15). However, another way for ANN output computing is useful. Let us define saturation vector \mathbf{z} of S elements. This vector indicates saturation states of hidden neurons – see (Eq. 21).

$$z_i = \begin{cases} 1 & \text{for } y^1_i > 1 \\ 0 & \text{for } -1 \leq y^1_i \leq 1 \\ -1 & \text{for } y^1_i < -1 \end{cases} \quad (21)$$

Then, ANN output can be expressed by (Eq. 22).

$$y_M(k) = -a_1 \cdot y_M(k-1) - a_2 \cdot y_M(k-2) + b_1 \cdot u(k-1) + b_2 \cdot u(k-2) + c \quad (22)$$

where

$$a_1 = -\sum_{i=1}^S w_{1,i}^2 \cdot (1 - |z_i|) \cdot w_{i,1}^1$$

$$a_2 = -\sum_{i=1}^S w_{1,i}^2 \cdot (1 - |z_i|) \cdot w_{i,2}^1$$

$$b_1 = \sum_{i=1}^S w_{1,i}^2 \cdot (1 - |z_i|) \cdot w_{i,3}^1$$

$$b_2 = \sum_{i=1}^S w_{1,i}^2 \cdot (1 - |z_i|) \cdot w_{i,4}^1$$

$$c = w_1^2 + \sum_{i=1}^S (w_{1,i}^2 \cdot z_i + (1 - |z_i|) \cdot w_{1,i}^2 \cdot w_i^1)$$

Thus, difference equation (22) defines ANN output and it is linear in some neighbourhood of actual state (in that neighbourhood, where saturation vector \mathbf{z} stays constant). Difference equation (22) can be clearly extended into any order.

In other words, if the neural model of any nonlinear system in form of Fig. 9 is designed, then it is simple to determine parameters of linear difference equation which approximates

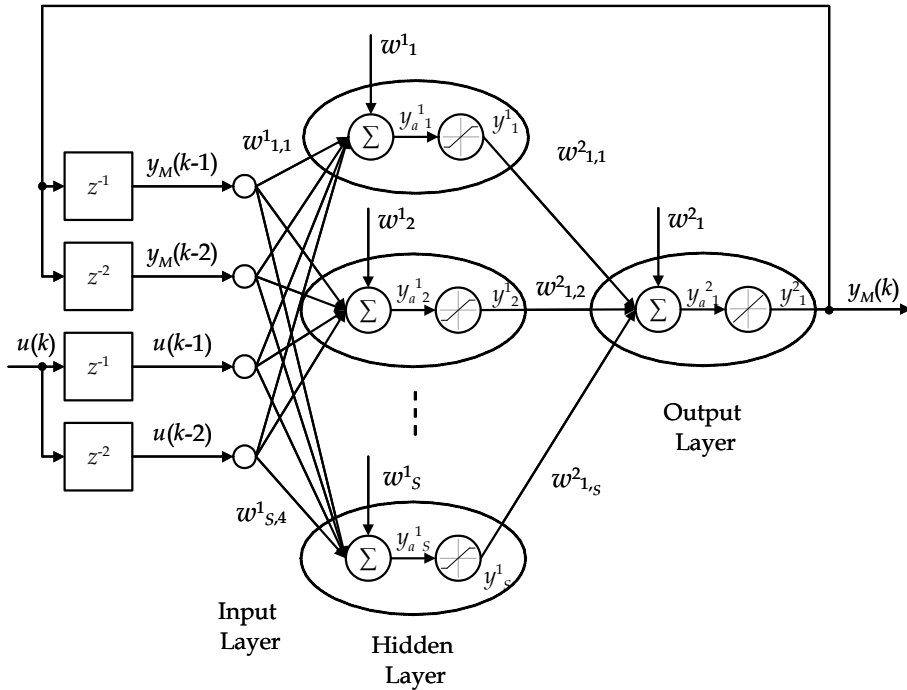


Fig. 9. Piecewise-linear neural model

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 (22) should be transformed in following way. Let us define

$$\tilde{u}(k) = u(k) - u_0 \tag{23}$$

where u_0 is constant. Then, (Eq. 22) turns into

$$y_M(k) = -a_1 \cdot y_M(k-1) - a_2 \cdot y_M(k-2) + b_1 \cdot \tilde{u}(k-1) + b_2 \cdot \tilde{u}(k-2) + c + (b_1 + b_2) \cdot u_0 \tag{24}$$

Equation (24) becomes constant term free, if (Eq. 25) is satisfied.

$$u_0 = -\frac{c}{b_1 + b_2} \tag{25}$$

In Z domain, model (24) with respect to (Eq. 25) is defined by Z - transfer function (26).

$$\frac{Y_M(z^{-1})}{\tilde{U}(z^{-1})} = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{26}$$

5. 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. 10, too).

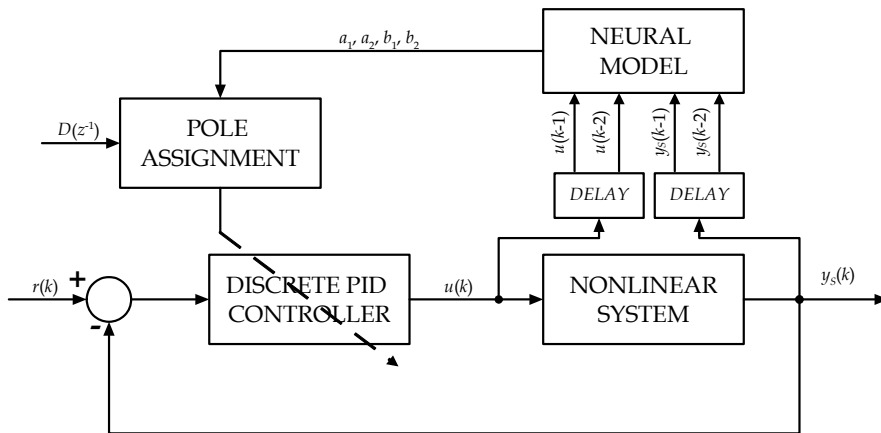


Fig. 10. Control algorithm scheme for second order nonlinear system

1. Create neural model of controlled plant in form of Fig. 9.
2. Determine polynomial $D(z^{-1})$ of (10).
3. Set $k = 0$.
4. Measure system output $y_s(k)$.
5. Determine the parameters a_i , b_i and c of difference equation (22).
6. Transform (Eq. 22) into Z - transfer function (26).
7. Determine discrete PID controller parameters by solving of (Eq. 11) where $A(z^{-1})$ and $B(z^{-1})$ are denominator and nominator of Z - transfer function (26), respectively.
8. Determine $\tilde{u}(k)$ using discrete PID controller tuned in previous step.
9. Transform $\tilde{u}(k)$ into $u(k)$ using (Eq. 23) and perform control action.
10. $k = k + 1$, go to 4.

Introduced algorithm is suitable to control of highly nonlinear systems, especially.

6. Case study

Discrete PID controller tuned continuously by technique introduced above is applied now to control of two nonlinear systems. Both of them are compiled by a combination of nonlinear static part and linear dynamical system – see Fig. 11.

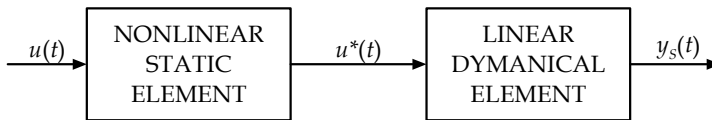


Fig. 11. System to control

6.1 First order nonlinear system

The static element of the first demo system is defined by (Eq. 27) and dynamical system is defined by differential equation (28).

$$u^*(t) = \left(\frac{2}{1 + e^{-2u(t)}} - 1 \right)^3 \quad (27)$$

$$y(t) + 10 \frac{dy(t)}{dt} = u^*(t) \quad (28)$$

Graphic characteristics of the system are shown in Fig. 12.

Control loop is designed as shown in paragraph 5. At first, dynamical piecewise-linear neural model in shape of Fig. 9 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 (Haykin, 1994), (Nguyen, 2003) it is not referred here in detail. Briefly, training set is gained by controlled system excitation by set of step functions with various amplitudes while both u and y_s are measured (sampling interval $T = 1$ s) – see Fig. 13. Then, order of the neural model is set: $n = 1$ (Eq. 19) because the controlled system is first order one, too. After that, artificial neural network is trained by Backpropagation Gradient Descent Algorithm repeatedly (see Fig. 14) while pruning is applied – optimal neural network topology is determined as two inputs,

four neurons in hidden layer and one output neuron. Finally, the neural model is validated (Fig. 15).

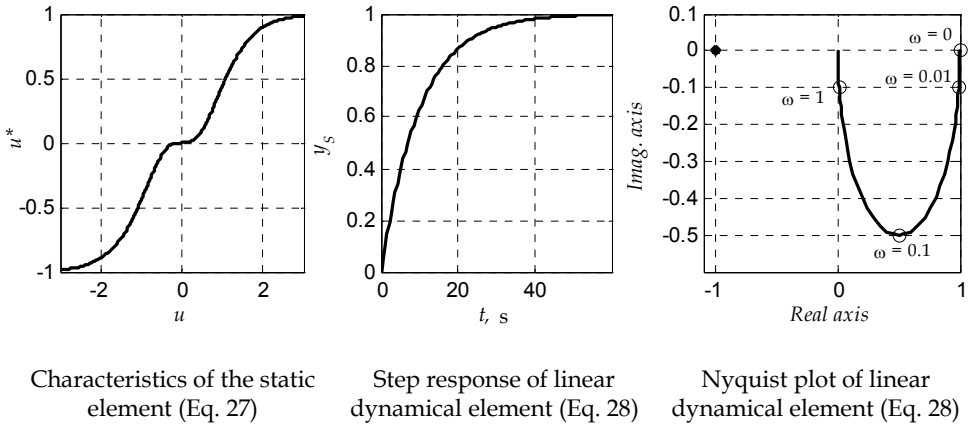


Fig. 12. Graphic characteristics of the first order nonlinear system

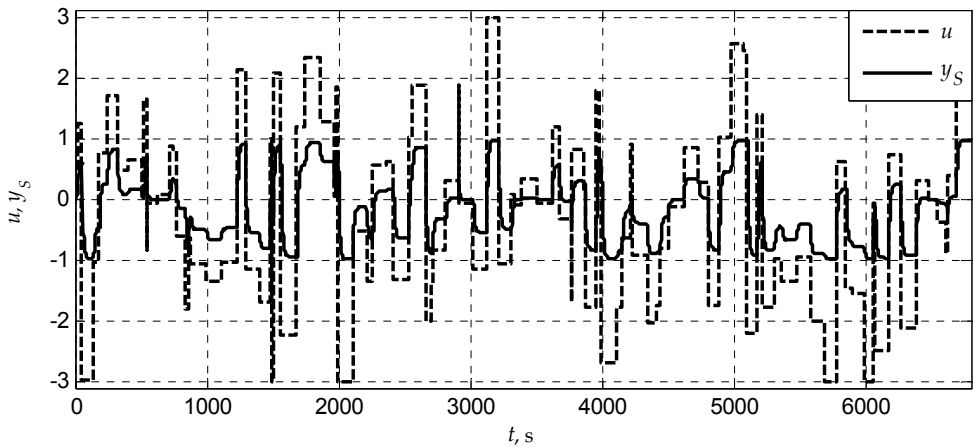


Fig. 13. Training set for the neural model

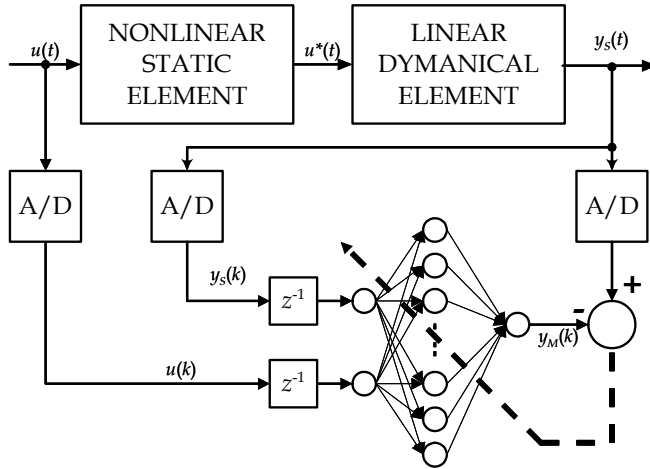


Fig. 14. Neural network training

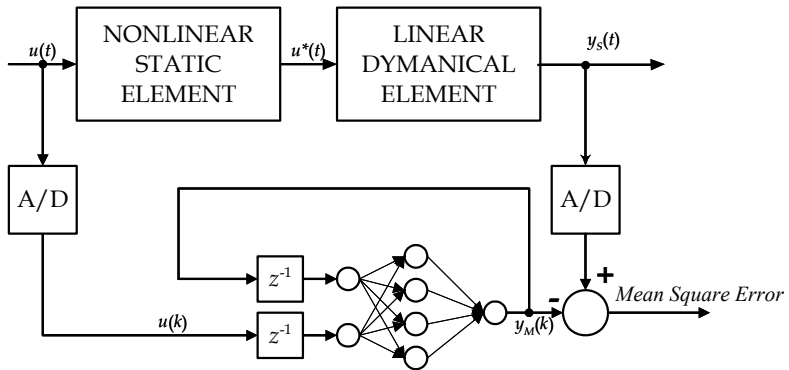


Fig. 15. Neural model validating

Next step is to determine polynomial $D(z^{-1})$. Common ways of $D(z^{-1})$ determination are mentioned below (Hunt, 1993).

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

Let us use the c) possibility and define the standard for control dynamics as second order system with Z - transfer function (29).

$$F(z^{-1}) = \frac{0.2642z^{-1} + 0.1353z^{-2}}{1 - 0.7358z^{-1} + 0.1353z^{-2}} \tag{29}$$

Thus,

$$D(z^{-1}) = 1 + d_1z^{-1} + d_2z^{-2} = 1 - 0.7358z^{-1} + 0.1353z^{-2} \tag{30}$$

Polynomial $D(z^{-1})$ is stable with double pole equal to 0.3679.

Essential part of next three steps of the control algorithm is to solve Diophantine equation (11). In this particular example, (Eq. 31) is to be solved.

$$1 + d_1z^{-1} + d_2z^{-2} = (1 + a_1z^{-1})(1 - z^{-1}) + b_1z^{-1}(q_0 + q_1z^{-1} + q_2z^{-2}) \quad (31)$$

Method of undetermined coefficients is one possibility how to solve this equation. The initial matrix equation is

$$\begin{bmatrix} b_1 & 0 & 0 \\ 0 & b_1 & 0 \\ 0 & 0 & b_1 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} d_1 - a_1 + 1 \\ d_2 + a_2 \\ a_2 \end{bmatrix} \quad (32)$$

And the solution is

$$\begin{aligned} q_0 &= \frac{d_1 - a_1 + 1}{b_1} \\ q_1 &= \frac{a_1 + d_2}{b_1} \\ q_2 &= 0 \end{aligned} \quad (33)$$

Now it is possible to perform control simulation. For defined reference variable course (combination of step functions and linearly descending and ascending functions), the control performance is shown in Fig. 16. Comparison of system output to standard (Eq. 29) is shown then in Fig. 17.

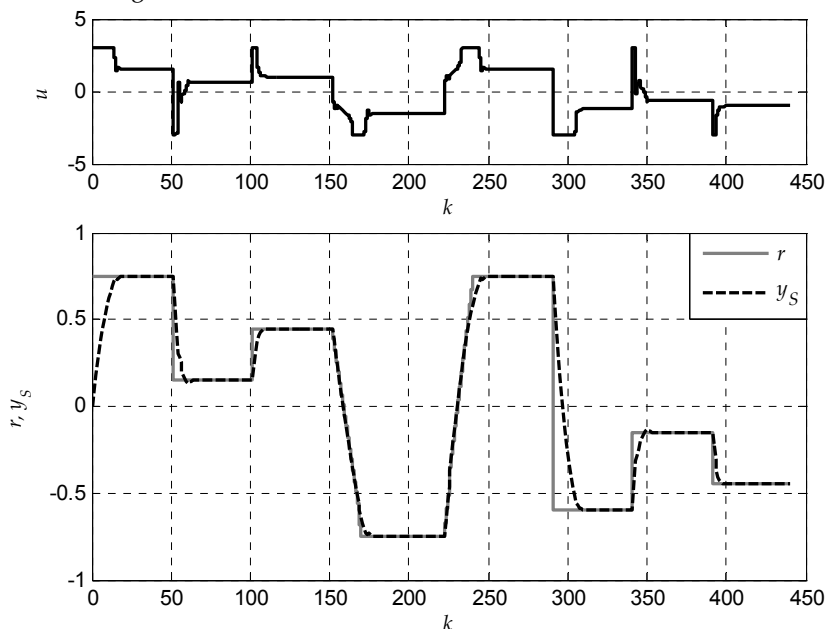


Fig. 16. Control performance – first order nonlinear system

As shown in Figs. 16 and 17, control performance is stable and desired dynamics of the closed loop is close to defined standard.

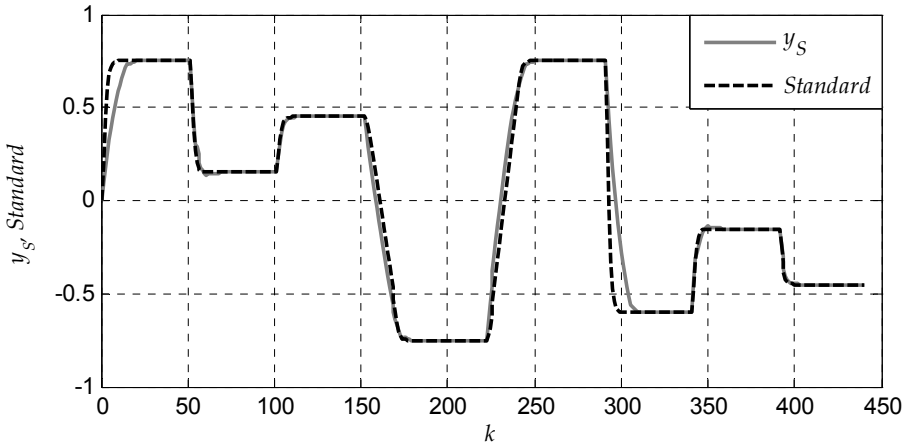


Fig. 17. Comparison to standard – first order nonlinear system

6.2 Second order nonlinear oscillative system

Second demo system is structurally identical as the previous one (Fig. 11). Even the static element is the same. However, the dynamic system is defined now by differential equation (34).

$$y(t) + 5 \frac{dy(t)}{dt} + 50 \frac{d^2y(t)}{dt^2} = u^*(t) \tag{34}$$

Graphic characteristics of the system are shown in Fig. 18.

The system is controlled on equal terms as previous one. However, the neural model now has four inputs as original system is second order one. Thus, Diophantine equation (35) should be solved.

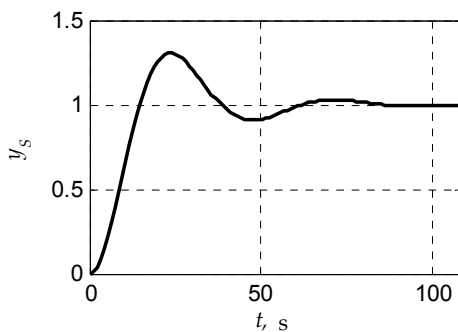
$$1 + d_1z^{-1} + d_2z^{-2} = (1 + a_1z^{-1} + a_2z^{-2})(1 - z^{-1}) + (b_1z^{-1} + b_2z^{-2})(q_0 + q_1z^{-1} + q_2z^{-2}) \tag{35}$$

However, equation (35) is unsolvable. Thus, algorithm of discrete PID controller has to be extended into Z - transfer function (36) which is kind a filtered discrete PID controller.

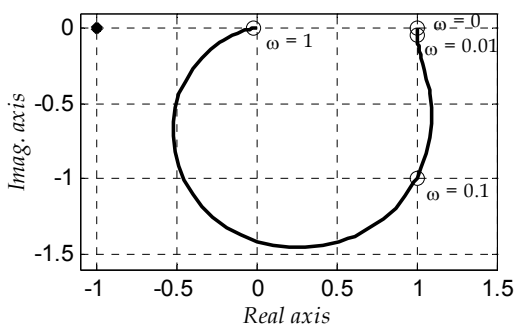
$$\frac{Q(z^{-1})}{P(z^{-1})} = \frac{q_0 + q_1z^{-1} + q_2z^{-2}}{(1 - z^{-1})(1 + \gamma z^{-1})} \tag{36}$$

Now, Diophantine equation (11) turns to (Eq. 37).

$$1 + d_1z^{-1} + d_2z^{-2} = (1 + a_1z^{-1} + a_2z^{-2})(1 - z^{-1})(1 + \gamma z^{-1}) + (b_1z^{-1} + b_2z^{-2})(q_0 + q_1z^{-1} + q_2z^{-2}) \tag{37}$$



Step response of linear dynamical element (Eq. 34)



Nyquist plot of linear dynamical element (Eq. 34)

Fig. 18. Graphic characteristics of the second order nonlinear oscillative system

After applying of method of undetermined coefficients, solution can be obtained by solving of following matrix equation.

$$\begin{bmatrix} b_1 & 0 & 0 & 1 \\ b_2 & b_1 & 0 & a_1 - 1 \\ 0 & b_2 & b_1 & a_2 - a_1 \\ 0 & 0 & b_2 & -a_2 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ \gamma \end{bmatrix} = \begin{bmatrix} d_1 - a_1 + 1 \\ d_2 + a_1 - a_2 \\ a_2 \\ 0 \end{bmatrix} \tag{38}$$

And the solution is

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ \gamma \end{bmatrix} = \begin{bmatrix} b_1 & 0 & 0 & 1 \\ b_2 & b_1 & 0 & a_1 - 1 \\ 0 & b_2 & b_1 & a_2 - a_1 \\ 0 & 0 & b_2 & -a_2 \end{bmatrix}^{-1} \begin{bmatrix} d_1 - a_1 + 1 \\ d_2 + a_1 - a_2 \\ a_2 \\ 0 \end{bmatrix} \tag{39}$$

Now it is possible to perform control simulation. For defined reference variable course, the control performance is shown in Fig. 19. Comparison of system output to standard (Eq. 29) is shown then in Fig. 20.

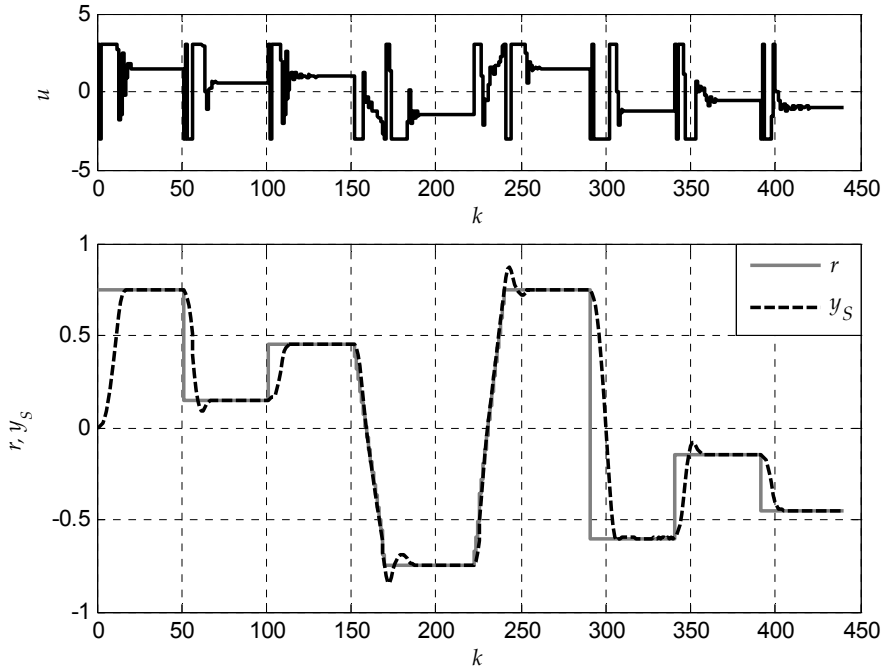


Fig. 19. Control performance – Second order nonlinear oscillative system

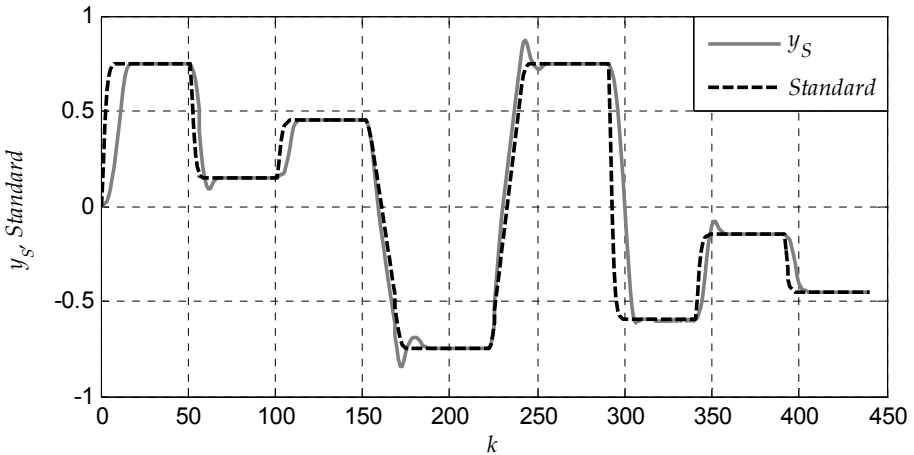


Fig. 20. Comparison to standard – Second order nonlinear oscillative system

As shown in Figs. 19 and 20, control performance is stable and satisfying. On the other hand, oscillative nature of the controlled system is not fully stifled.

7. Conclusion

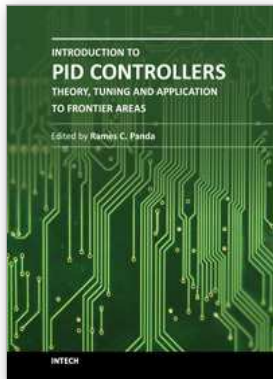
There is introduced the technique above, which performs continuous adaptation of PID controller via neural model of controlled system. Neural model is used for controlled system continuous linearization and that linearized model is used for discrete PID controller tuning using pole assignment. The technique is suitable for highly nonlinear systems control, while it brings no advantages to control of the systems which are close to linear ones.

8. Acknowledgement

The work has been supported by the funds of the framework research project MSM 0021627505, MSM 6046137306 and by the funds of the project of University of Pardubice SGFEI06/2011 „Artificial Intelligence Control Toolbox for MATLAB“. This support is very gratefully acknowledged.

9. References

- Astrom, K. J. & Haggglund, T. (1995). *PID controllers: theory, design and tuning*, International Society for Measurement and Control, ISBN 1-55617-516-7, Durham, North Carolina, USA
- Bennett, S. (1993). *A history of control engineering, 1930-1955*, IET, ISBN 0-86341-280-8, Stevenage, UK
- Doyle, J., Francis, B. & Tannenbaum, A. (1990). *Feedback control theory*, Macmillan Publishing, ISBN 0-02330-011-0, New York, USA
- Dwarapudi, S.; Gupta, P. K. & Rao, S. M. (2007). Prediction of iron ore pellet strength using artificial neural network model, *ISIJ International*, Vol. 47, No 1., pp. 67-72, ISSN 0915-1559
- Doležel, P.; Taufer, I. & Mareš, J. (2011). Piecewise-Linear Neural Models for Process Control, *Proceedings of the 18th International Conference on Process Control '11*, pp. 296-300, ISBN 978-80-227-3517-9, Tatranská Lomnica, Slovakia, June 14-17, 2011
- Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation*, Prentice Hall, ISBN 0-02352-761-7, New Jersey, USA
- Hecht-Nielsen, R. (1987). Kolmogorov's mapping neural network existence theorem, *Proc 1987 IEEE International Conference on Neural Networks*, Vol. 3, pp. 11-13, IEEE Press
- Hunt, K. J., Ed. (1993). *Polynomial methods in optimal control and filtering.*, IET, ISBN 0-86341-295-5, Stevenage, UK
- Isermann, R. (1991). *Digital Control Systems*, Springer-Verlag, ISBN 3-54010-728-2, Heidelberg, Germany
- Montague, G. & Morris, J. (1994). Neural network contributions in biotechnology, *Trends in biotechnology*, Vol. 12, No 8., pp. 312-324, ISSN 0167-7799
- Nguyen, H.; Prasad, N.; Walker, C. (2003). *A First Course in Fuzzy and Neural Control*, Chapman & Hall/CRC, ISBN 1-58488-244-1, Boca Raton, USA



Introduction to PID Controllers - Theory, Tuning and Application to Frontier Areas

Edited by Prof. Ramesh C. Panda

ISBN 978-953-307-927-1

Hard cover, 258 pages

Publisher InTech

Published online 29, February, 2012

Published in print edition February, 2012

This book discusses the theory, application, and practice of PID control technology. It is designed for engineers, researchers, students of process control, and industry professionals. It will also be of interest for those seeking an overview of the subject of green automation who need to procure single loop and multi-loop PID controllers and who aim for an exceptional, stable, and robust closed-loop performance through process automation. Process modeling, controller design, and analyses using conventional and heuristic schemes are explained through different applications here. The readers should have primary knowledge of transfer functions, poles, zeros, regulation concepts, and background. The following sections are covered: The Theory of PID Controllers and their Design Methods, Tuning Criteria, Multivariable Systems: Automatic Tuning and Adaptation, Intelligent PID Control, Discrete, Intelligent PID Controller, Fractional Order PID Controllers, Extended Applications of PID, and Practical Applications. A wide variety of researchers and engineers seeking methods of designing and analyzing controllers will create a heavy demand for this book: interdisciplinary researchers, real time process developers, control engineers, instrument technicians, and many more entities that are recognizing the value of shifting to PID controller procurement.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Petr Doležel, Ivan Taufer and Jan Mareš (2012). Discrete PID Controller Tuning Using Piecewise-Linear Neural Network, Introduction to PID Controllers - Theory, Tuning and Application to Frontier Areas, Prof. Ramesh C. Panda (Ed.), ISBN: 978-953-307-927-1, InTech, Available from:
<http://www.intechopen.com/books/introduction-to-pid-controllers-theory-tuning-and-application-to-frontier-areas/discrete-pid-controller-tuning-using-piecewise-linear-neural-network>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2012 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the [Creative Commons Attribution 3.0 License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.