IMPLEMENTATION OF ADAPTIVE OPTIMAL CONTROLLER INTO PLC

Vlastimil Lorenc
Doctoral Degree Programme (1), FEEC BUT
E-mail: xloren05@stud.feec.vutbr.cz

Supervised by: Petr Pivoňka
E-mail: pivonka@feec.vutbr.cz

ABSTRACT

This article describes implementation adaptive linear optimal controller (LQ) based on pseudo-space model into the PLC. Identification with artificial neural network is used. In the last part of article is shown comparison between adaptive self-tuning PSD controller and adaptive LQ controller.

1 INTRODUCTION

The idea of the adaptive controllers is to adapt parameters of control law according to changes of the controlled system. Many types of adaptive controllers are known. In this article the adaptive self-tuning LQ controller is described. Scheme of this controller is separated into the two main parts: identification and controller. In this work identification based on neural network approach is used and control algorithm is the linear quadratic optimal controller.

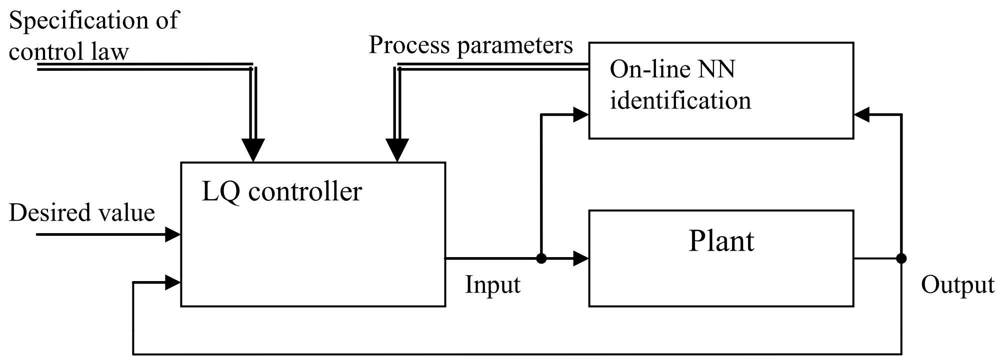

Figure 1: Architecture of self-tuning LQ controller.

2 ON-LINE IDENTIFICATION

For identification of systems is very widely used algorithm recursive least square method (RLS). Instead of RLS, the identification method based on neural network can be used. A very fast algorithm for training neural networks is the Levenberg Marquardt (LM) algorithm.

The main idea of on-line identification is that according to the measured input to the identified system *u(t)* and the corresponding system output *y(t)* we are able to find the vector of system parameters *Θ*.

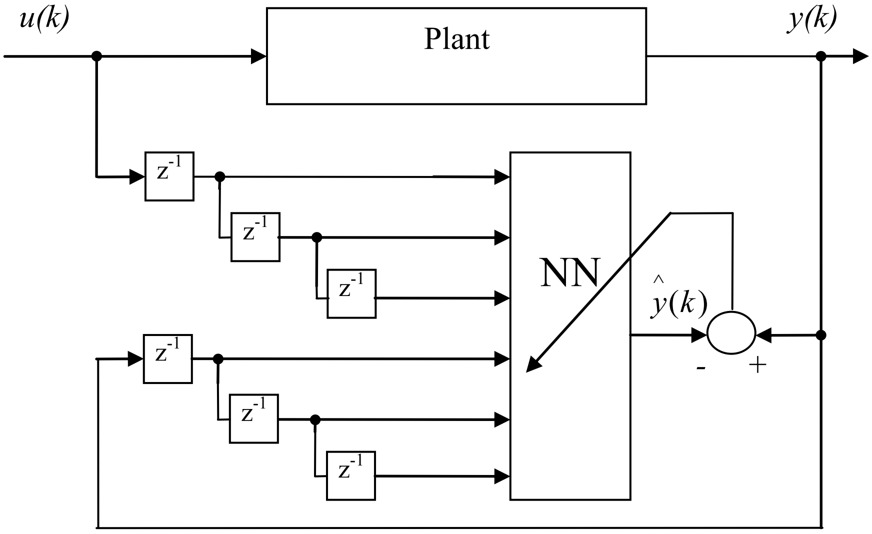

Figure 2: The principle of identification of system using neural network.

For computing the identified system output we can use the linear ARX model

$$\hat{y}(k) = \varphi^T(k)\theta(k) \qquad (1)$$

where

$$\varphi(k) = \left[u(k-1) \ \dots \ u(k-1-m) - y(k-1) \ \dots \ -y(k-1-n)\right]^T \qquad (2)$$

is the vector of measured inputs and outputs and

$$\theta(k) = \left[b_1(k) \ \dots \ b_m(k) \ a_1(k) \ \dots \ a_n(k)\right]^T \qquad (3)$$

is the vector of estimated system parameters.

As it was written for training of the neural network can be used the Levenberg–Marquardt method. New vector of parameters is in each step given by next equation.

$$\theta(k+1) = \theta(k) - \left(JJ^T + \lambda I\right)^{-1} J^T \varepsilon(k) \qquad (4)$$

where J is Jacobian matrix in form

$$J = \begin{pmatrix} \frac{\partial \varepsilon_1}{\partial w_1} & \cdots & \frac{\partial \varepsilon_1}{\partial w_2} \\ \vdots & \ddots & \vdots \\ \frac{\partial \varepsilon_p}{\partial w_1} & \cdots & \frac{\partial \varepsilon_p}{\partial w_n} \end{pmatrix} \qquad (5)$$

and ε is the vector of errors between the output of the system and the output of the model for all training patterns. The parameter p is the number of training patterns and n is the number of estimated parameters.

3 LINEAR LQ CONTROLLER

Quadratic performance can be defined by

$$J = x^T(k)Qx(k) + \sum_{k=k_0+1}^{k_0+T} q_y\left(w(k) - y(k)\right)^2 + q_u\left(u(k) - u0(k)\right)^2 \qquad (6)$$

where *w(k)* denotes desired value, *y(k)* system output and *u(k)* is action value. Parameter *u0(k)* is signal equal to desired value, which is used for elimination of offset. Next parameters q_y (q_u) denotes weights for output (action) value, k_0 denotes the first step while the minimization is used and $x^T(k)Qx(k)$ denotes the minimum at the last step k_0+T.

When we will work with the pseudo state matrix $S = [S_u, S_x, S_w, S_{u0}]$ defined by

$$S_u = \begin{bmatrix} 1 \\ 0 \\ 0 \\ b_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix},\ S_x = \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & \ddots & 0 & 0 & \cdots & 0 \\ b_1 & \ldots & b_m & a_1 & \cdots & a_n \\ 0 & \cdots & 0 & 1 & 0 & 0 \\ \vdots & \ddots & \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 \end{bmatrix},\ S_w = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \end{bmatrix},\ Su0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 1 \end{bmatrix} \qquad (7)$$

and when we use vector $z^T(k) = [x(k), w(k), u0(k)]$ and $x^T = [u(k), x(k\text{-}1), w(k), u0(k)] = S(k)z(k\text{-}1)$ we can rewrite quadratic performance to the more suitable form [1, 2]

$$J = \sum_{k=k_0+1}^{k_0+T} z^T(k)Qz(k)\,. \qquad (8)$$

Using nonstandard state-vector we can build universal quadratic performance. For example for standard penalization according to equation (6) is matrix *Q* defined as follows

$$Q = \begin{bmatrix} q_u & 0 & 0 & 0 & 0 & 0 & 0 & -q_u \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_y & 0 & 0 & -q_y & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -q_y & 0 & 0 & q_y & 0 \\ -q_u & 0 & 0 & 0 & 0 & 0 & 0 & q_u \end{bmatrix}. \qquad (9)$$

The method for minimization quadratic performance (8) is known [1, 2]. In each step is solved next algorithm where $H = S^TQS$:

Step	Equation	Notes
1.	$H^* = H_{xx} - H^T_{ux} - H^{1}_{uu}H_{ux}$	recursively solves lost function
2.	GDG^T	LD-FIL decomposition
3.	$u(k) = -G_{uu}^{-1}G_{ux}x(k-1)$	solves action value

Table1: Iteration algorithm for solving LQ controller.

4 REAL PROCESS CONTROL RESULTS

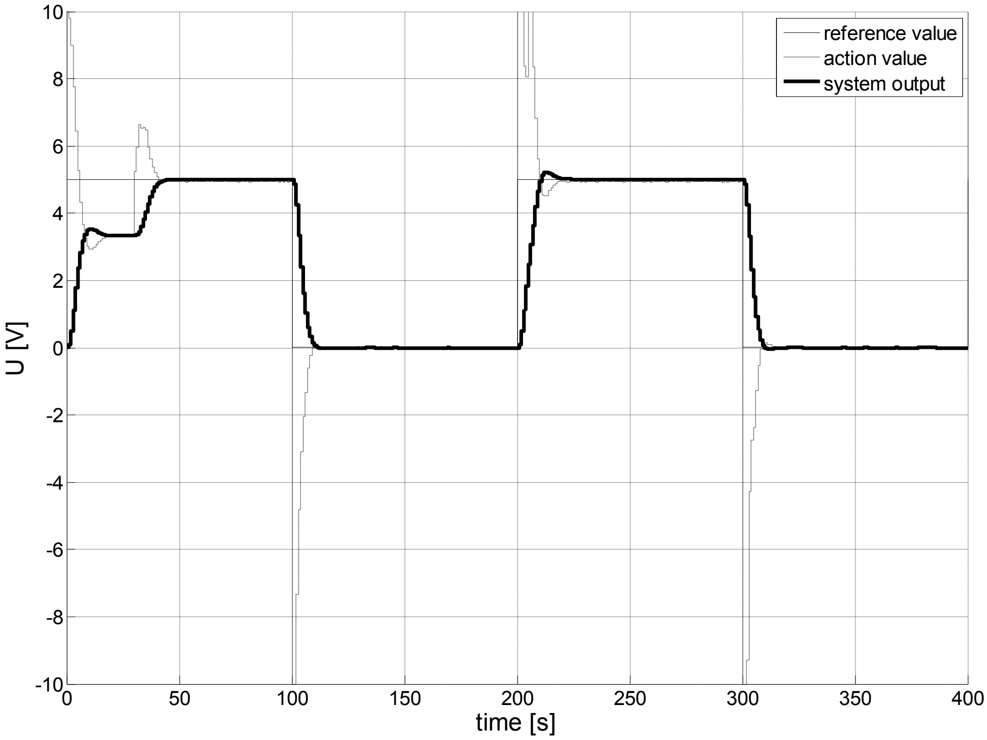

Figure 3: Real process control using LQ controller (q_u =1, q_y = 10).

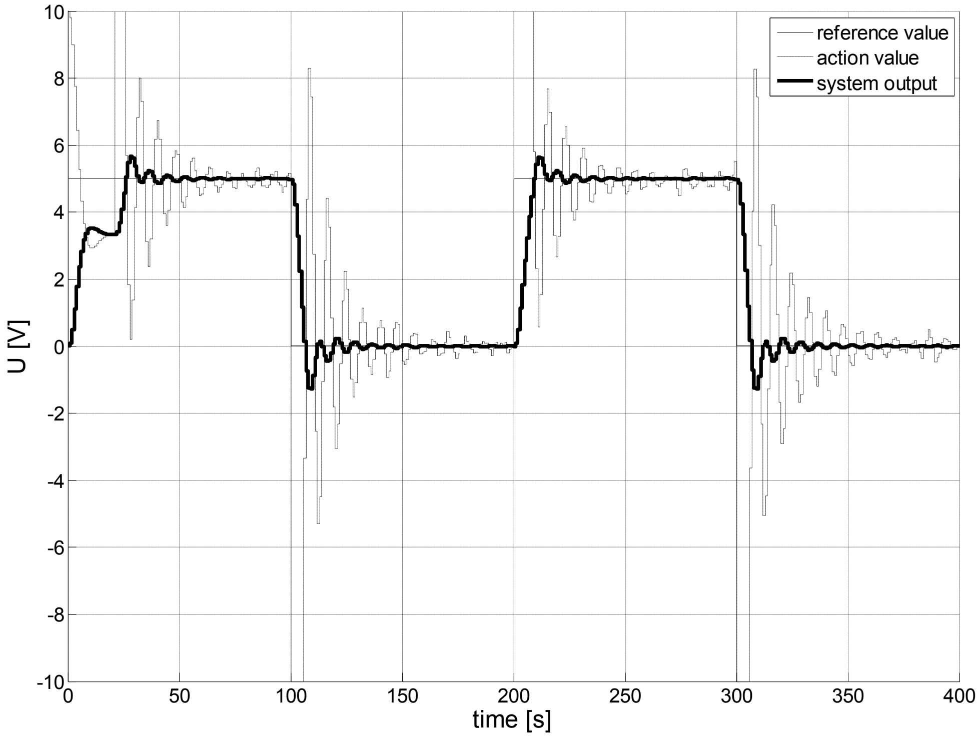

Figure 4: Real process control using self-tuning PSD controller.

Described LQ controller was implemented into the PLC B&R. As controlled system was used the laboratory model with the transfer function $F(s) = \frac{1}{(10s+1)(s+1)^2}$. In the figure 3 we can see real process response with described LQ control algorithm. Figure 4 shows real process response with the self-tuning PSD controller based on modified Ziegler Nichols method [2, 4]. In both cases the identification based on neural network approach with the same setup and initial conditions was used. Sampling period was set to $T_s = 1s$.

5 CONCLUSION

It was described design of adaptive LQ controller that uses universal weight matrix. The universal weight matrix Q can be written in many forms to designer's expectation. This algorithm solves one iteration in each step. It means that time for computing is short and this algorithm can be used for implementation in industrial controller. In the last two figures we can see comparison between LQ adaptive controller and self-tuning PSD controller. One can see that in term of changes of action value LQ controller produces the better results.

Acknowledgement: The paper has been prepared as a part of the solution of Czech Science Foundation GAČR project No. 102/06/1132 Soft Computing in Control and by the Czech Ministry of Education in the frame of MSM MSM0021630529 Intelligent Systems in Automation.

REFERENCES

[1] Švancara, K.: Adaptive Optimal Controller with Identification Based on Neural Networks, PhD thesis, FEEC BUT, Brno 2004.

[2] Bobál V., Böhm J., Prokop R. and Fessl J.: Practical Aspects of Self-Tuning Controllers, Algorithms and Implementation (Praktické aspekty samocinne se nastavujících regulátoru: algoritmy a implementace), VUTIUM, 1999, in Czech, ISBN 80-214-1299-2.

[3] Švancara K.: and Pivonka P.: Closed Loop On-line Identification Based on Neural Networks in Adaptive Optimal Controller, In Proceedings of 10th Zittau Fuzzy Colloquium, Zittau (Germany), 2002.

[4] Lorenc V.: Self-Tuning Controllers in MATLAB - B&R Environment, Diploma thesis in Czech, FEEC BUT, Brno 2006.