Proceedings of the 7th WSEAS International Conference on Automation & Information, Cavtat, Croatia, June 13-15, 2006 (pp156-160)

A new Adaptive PID Control Approach Based on Closed-Loop Response Recognition

MARTIN FOLTIN, JÁN MURGAŠ, IVAN SEKAJ
Department of Electrical Engineering and Information Technology
Slovak University of Technology
Ilkovičova 3, 812 19 Bratislava
SLOVAKIA

Abstract: - A new type of adaptive control, based on the closed-loop response recognition is proposed. This approach tries to mimic the behaviour of an expert, who has experience with the particular controlled system. The adaptive system updates the controller parameters according to the changes of system behaviour. The algorithm uses a knowledge-based system, which evaluates the closed-loop response shape after the decay of the transient response or after only a part of it, and computes corrections of controller parameters. Because the main problem of such a control structure consists in designing of the knowledge-base, we focused our interest on the design of an automatic procedure for the knowledge-base generating.

Key-Words: - adaptive control, control response shape, recognition, knowledge base generating, PID controller

1 Introduction

To control dynamic processes, with time-varying or non-linear behaviour, the adaptive control systems use to be used. In such cases usually exact mathematical models, stability and performance conditions are employed for control algorithm design. The results obtained by such an approach are often very good, however sometimes same problems may emerge e.g. when the complexity or non-linearity degree of the model increases or in case of inexact measurements or insufficient a priori information about the process (unmodeled dynamics etc.). Thus, sometimes the "exact" mathematical approach and the resulting algorithms cannot be used to control the given process or some phases of it. In that moment it is the task for a human-operator to carry on and to control the process to the required state. The human works with his own model, based on his experience and estimations. He is able to deal with inexact, uncertain, nonnumeric and complex information (he exploits his „natural“ intelligence), which in our case, he is able to apply for the process control. Using the operator „manual adaptation“ of a complex or non-exactly described processes the results are sometimes better than using the „automatic adaptation“.

The goal of the proposed approach is to mimic the human behaviour and to update the controller parameters with respect to the required closed-loop response shape. The underlying idea consists in that after the disturbance occurs or after the reference signal change respectively we let the transient process decay and then evaluate the close-loop response (Fig. 1.). After evaluating of the control process characteristics, corrections of actual controller parameters are computed using a knowledge-based mechanism.

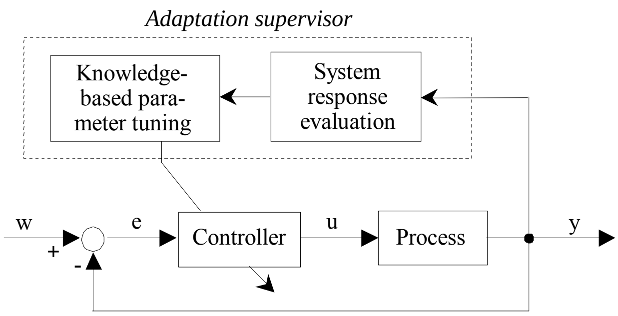

Fig. 1: Block scheme of the knowledge-based adaptation system of a PID controller

The aim is to design a self-tuning algorithm, which will provide a good controller adaptation without exact a-priori information about the process. Similar methods can be found in [1], [2], [3] [5], [6]. These methods are using various characteristics for the transient mode based on overshoot, settling time, damping, etc., which are inputs into the knowledge-based adaptation mechanism. On the other hand, our method is based on a time-response shape recognition using the least-square-error method. The main problem of all knowledge-based strategies consists in the design of the knowledge-base. This requires experience, knowledge and time for tuning. From that reason we focused our interest on design of an automatic algorithm for the knowledge-base creation. This algorithm is able to generate the adaptation knowledge-base for the user defined process (or a class of processes).

2 The Self-Tuning Principle

As mentioned, the first part of the adaptation mechanism is the transient process evaluation. When some disturbance in the control loop occurs or when the reference signal is changing, we let decay the transient process, or at least some part of it, and then we evaluate the closed-loop response. The recognition of the transient response shape is based on its comparison with a set of representative shapes, which are stored in a database, using the least-square-error method. After retrieving the "most similar shape" from the database, corrections of the PID controller are calculated using the knowledge-base. Without loss of generality consider the controller be of the type PID, which is described in time domain by the equation (1)

$$u(t) = Pe(t) + I\int e(t)dt + D\frac{de(t)}{dt} \tag{1}$$

where P, I, D are the controller parameters, which will be adapted. As mentioned, the basic part of the self-tuning system is the knowledge-base in form of a database. The database is designed (automatically) by an off-line algorithm in advance, before the adaptation system is used in the on-line operation. The method can be used under one of the following two considerations. The first is, that it is possible to identify the operating points or the system parameter space boundaries respectively of the controlled system, which can be reached during the operation. For such a space it is possible to generate the adaptation knowledge-base. The second alternative is that we are able to recognize the "type" of the controlled system according to its time-response shapes ("type of its dynamic behaviour"), no matter what is the time scale. According this we can search for an appropriate (in advance prepared) adaptation knowledge-base. A description of how such database is designed, what is its structure and how it is used is in the following section.

2.1 Self-tuning database creation

Consider that the controlled system is operating in a working space, with known boundaries, or some physical operating points are known respectively. Let us consider a “nominal” model of this system, which parameters are in the "centre" of this working space (average of the identified parameter values) or the nominal model is the most important working point. For this nominal system let us design a PID controller (with the parameters P^*, I^*, D^*), which will guarantee desired closed-loop behaviour (e.g. no overshoot, desired settling time, oscillation damping, etc. i. e. (Sekaj 99)). The idea of the adaptation knowledge-base (or database) creation is based on an "inverse" procedure. Here we consider instead of a closed-loop model, consisting of the PID controller and the controlled system with its moving parameters, a "quasi-equivalent" closed-loop model with the nominal system and the PID with moving parameters

$$P = P^* + \Delta P, I = I^* + \Delta I, D = D^* + \Delta D$$

such that

$$\begin{aligned} P &\in \langle P_{\min}, P_{\max} \rangle = I_P \\ I &\in \langle I_{\min}, I_{\max} \rangle = I_I \\ D &\in \langle D_{\min}, D_{\max} \rangle = I_D \end{aligned} \tag{2}$$

The size of the controller parameter intervals can be (roughly) estimated with the aim to cause similar changes of the closed-loop dynamic behaviour (from point of view response shapes) as in the real process.

Remark: In general it is not possible to transform the closed loop with fix PID controller and moving system parameters into an equivalent closed loop with 3 moving PID parameters and the fix nominal system. But it can be experimentally shown that a rough estimation of (2) is normally sufficient for the knowledge-base creation.

The space of the controller parameters is 3-dimensional

$$S_{PID}=I_P \times I_I \times I_D, \; S_{PID} \subset R^3 \tag{3}$$

Let us generate a 3-D orthogonal grid with defined step sizes s_P , s_I and s_D for each interval. The number of all nodes of this grid is $n=n_P.n_I.n_D$, where $n_P = \frac{I_P}{s_P}, n_I = \frac{I_I}{s_I}, n_D = \frac{I_D}{s_D}$. Let us remark, that the commonly used size of the database was between 5x5x5 and 9x9x9. Each node of the grid represents a set of 3 PID parameters.

The algorithm for generating of the knowledge-base is as follows. For each node of the grid with the corresponding controller parameters P_g, I_g, D_g (where g denote the coordinates in the grid) a simulation is performed of the closed-loop system, which result is the time-response of the selected disturbance (set-point step, external disturbance, etc.). The obtained time-response is stored into the database together with the current controller parameters.

In case of a PI controller the database has a form of a 2-D table, where each item of the table contains a record of the closed-loop time-response and the set of the corresponding parameters P and I. An illustrative example for a particular closed-loop is in the Fig.2, where the highlighted response represents the desired closed-loop response shape. For the PID case the table is by analogy 3-dimensional.

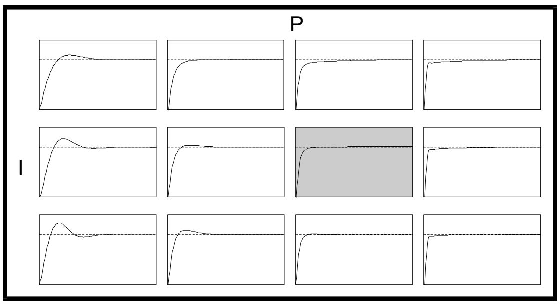

Fig. 2 : Database of closed-loop time-responses

2.2 The on-line use of the self-tuning system

After an off-line generating of such a database all data are ready for the application in the on-line adaptive system. The adaptation mechanism is as follows. After the detection of some disturbance in the closed-loop we let decay the transient period (or only a part of it). Next, the acquired time response is compared with all records in the database and the "most similar" time-response is found from all records in the database, i.e. such, which minimize the criterion (4), (Fig.3)

$$J = \sum_{k=1}^{N} (y_{g,k} - y_k)^2 \rightarrow \min \qquad (4)$$

where y_g is the closed-loop response from the database under the controller parameters P_g, I_g, D_g , g is the position in the database of the "most similar" time-response, y is the actual time response of the controlled system and N is the number of time-response samples.

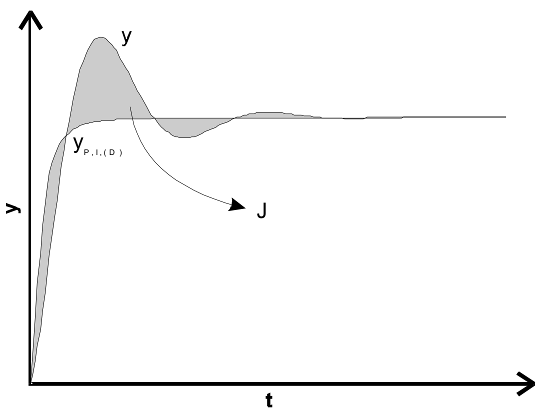

Fig. 3: Recognition of the time-response using the least-square error method

After retrieving of the selected time-response the corresponding parameter set (P_g, I_g, D_g) is used for the calculation (or it is interpolated between neighbour items in the grid). The last step is the calculation of controller parameter corrections (5)

$$\Delta P = P^* - P_g \ , \ \Delta I = I^* - I_g \ , \ \Delta D = D^* - D_g \qquad (5)$$

Finally, these corrections are used for the actual PID controller updating (6)

$$\begin{aligned} P_{new} &= P_{actual} + \Delta P \\ I_{new} &= I_{actual} + \Delta I \\ D_{new} &= D_{actual} + \Delta D \end{aligned} \qquad (6)$$

After the adaptation, the algorithm is waiting for new disturbances and the adaptation cycle is repeating.

The main steps of the algorithm are as follows:

1. Detection of a disturbance and waiting for the closed-loop response decay
2. Searching for the most similar time response from the database using (4)
3. Interpolation between neighbour PID parameter sets in the database
4. Correction of the actual PID parameter set using (5) and (6)

Remark 1: To ensure that the time responses in the algorithm are independent on the time constants, all time responses (in off-line database generating and in on-line processing) can be normed by suitable time constants before their use. This can ensure that the same knowledge-base can be used for adaptation of systems with different time constants with similar time-response shapes (i.e. thermal, hydraulic, electrical systems).

Remark 2: A modified version of the adaptation mechanism is possible, where the adaptation is not applied after the complete step response, but only after a part of it (after the first reaching of the setpoint, after reaching of 66% of the setpoint, after defined time etc.). In some cases only 5% of the step response time is enough for the correct recognition and adaptation. This can considerably speed-up the adaptation process.

3 Case Study

In the first simulation experiment (Fig. 4 and Fig. 5) a typical adaptation process is depicted (line 1), where the setpoint is changing each 40s between values 0 and 1. It is evident, that after an oscillating response, when the controller parameters were not well tuned the self-tuning procedure have corrected the parameters. In the next setpoint step the control process has the required behavior. Line 2 (bold) represents the modified version of the adaptation mechanism, where the adaptation is not applied after the complete transient mode decay, but after 5 % of it.

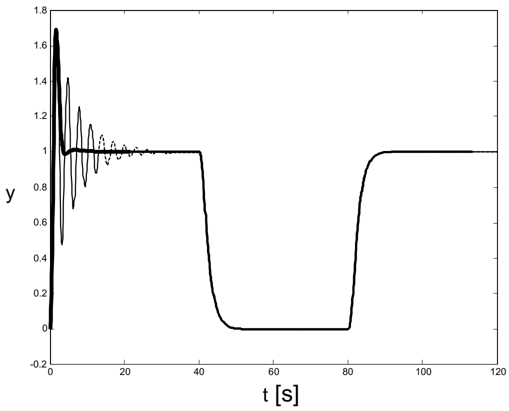

Fig. 4. Adaptation process using the complete transient mode - 1, adaptation process using only a part of the transient mode - 2 (bold)

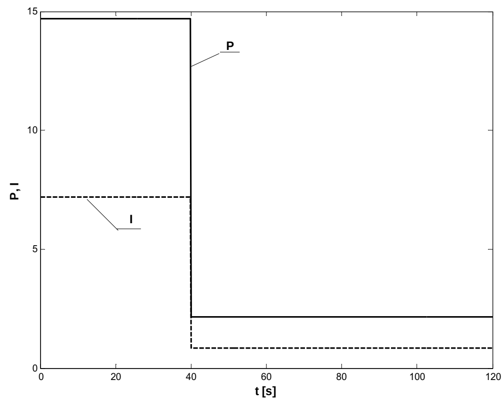

Fig.5 Time behaviour of PID controller parameters for the line 1

In Fig. 6 an example of the adaptation process of a DC-drive speed control (in real-time) is depicted. In time t=0 the controller is not well tuned. After each setpoint change some corrections of the parameters are performed. The adaptation process is completely finished after the 4-th adaptation cycle.

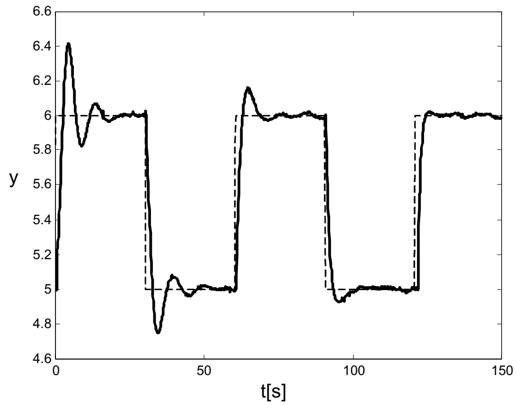

Fig. 6 : Adaptation process of the PID controller of a DC-drive speed

Finally, let us make a short remark about the computational requirements of the proposed approach. The off-line generating of the database, which depends on the size, is taking about 2 - 5 minutes of computation time and a size of 30 MB of memory space (P4/2.0GHz processor). The on-line adaptation algorithm is working with the downloaded database and is fast and computationally simple. It can be applied also on conventional process-level control systems (industrial PC, PLC, monolithic microcomputers etc.)

4 Conclusion

The described knowledge-based adaptation of a PID controller mimics the tuning of the controller by an experienced human-operator. The advantage of this approach is in, that it does not require a mathematical model of the controlled process. It is able to adapt the PID controller parameters only after the closed-loop behaviour recognition, after the decay of the transient mode or after decay of a part of it. Another advantage is the possibility to use the designed knowledge-base for a class of systems with similar dynamic behaviour, no matter what are their time constants. Many experiments in simulation and real-time with various controlled systems (linear/non-linear, non-oscillatory/oscillatory, minimum phase/non-minimum phase, SISO, MIMO, etc.) have proved, that the proposed adaptation approach is robust, it is working reliable and really satisfactory.

Acknowledgement. This work has been supported from the grants of the Slovak grant agency VEGA 1/3841/06.

References:

[1] DeSilva C.W. (1991). An Analytical Framework for Knowledge-Based Tuning of Servo Controllers, *Eng. Applic.Artif.Intellig.*, Vol. 4, No.3, p. p. 177-189.

[2] Oliveira, P., P.Lima, J.Sentineiro (1991). Fuzzy Supervision on Intelligent Control Systems, *ECC 91*, Grenoble, France

[3] Pfeiffer, B.M., R.Isermann (1993). Selftuning of classical controllers with fuzzy-logic, *Proc. of IMACS Symp. "Mathematical and Intelligent models in system Simulation"*, Brussels

[4] Sekaj, I. (1999). Genetic Algorithm-Based Control System Design and System Identification, *Conferrence Mendel'99*, Brno, Czech Republic, june 9th –12th , p. p. 139-144

[5] Sekaj, I. (2000). Knowledge-Based Adaptive Control, *Doctoral thesis*, FEI STU Bratislava

[6] Sekaj, I., M. Foltin (2000). Adaptive Control Based on the Closed-Loop Response Recognition. *VDI Berichte*, Vol. No. 1761-1805, p. p. 82-88.