Proceedings of the 8th IEEE Mediterranean Conference
on Control and Automation (MED 2000)
Rio Patras, GREECE 17-19 July 2000

MODIFICATION AND APPLICATION OF AUTOTUNING PID CONTROLLER *

NEDJELJKO PERIĆ, IVAN BRANICA, IVAN PETROVIĆ

Department of Automatic Control and Computer Engineering in Automation, Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia, nedjeljko.peric@fer.hr, ivan.branica@fer.hr, ivan.petrovic@fer.hr

Abstract. This contribution presents a modified autotuning algorithm of the PID controller. The motivation for the modification of the basic autotuning algorithm is to enlarge the class of processes to which it can be applied. The basic autotuning algorithm introduced by Åstrom and Hägglund is extended by the preliminary identification procedure and through the usage of the dead time compensating controller. These modifications are detailed through the description of the algorithms' functioning. The proposed algorithm has been implemented in the programmable logic controller (PLC) Siemens SIMATIC S7-300. The experimental results confirm the good robustness properties of the proposed algorithm, which were demonstrated in a simulation study.

Key Words. Autotuning, PID control, relay experiment, PLC.

1. INTRODUCTION

Control systems are designed to minimise the effects of process variations and environmental influences on the quality of process control. Sometimes these variations and influences are significant, so that the conventional linear controllers with constant parameters are unable to control the processes successfully. The main tasks of the controller in such situations are retaining the stability and maintaining the desired performance of the control system. Two control engineering approaches have been devised to achieve these demanding goals: robust control and adaptive control. In the robust control approach a fixed controller is designed, which satisfies the requirements for all models belonging to a certain class. If this class of models encompasses all expectable process and environment variations, the designed controller solves the problem. On the other hand, an adaptive controller adjusts itself to the changes during operation. It recognises variations in the process and in the environment, and adapts the structure and the parameters of the controller accordingly. The adaptation mechanism further automates the control of the process by performing tasks usually made by the control engineer, and thus, extends the idea of the feedback loop. It acts in the adaptation loop of an adaptive control system.

Research into control paradigms resulted in the development of the following adaptive control techniques [1]: gain scheduling, model reference adaptive systems (MRAS), self-tuning regulator (STR) and autotuning.

As known from the literature [2], the autotuning procedure is performed on the demand of the user, or colloquially after a 'button push'. Thus, it is not performed continuously in the adaptation loop, but rather when the need for tuning or re-tuning arises. This technique reiterates the design steps which the control engineer performs during the design of the controller. Firstly, a simple experiment is performed which determines some characteristics of the process. After that, using the data obtained, the controller parameters are calculated, and the designed controller

* Croatian Ministry of Science has supported this work under grant 036-006. Laboratory was equipped in collaboration with Siemens–Croatia and Pliva–Croatia.

is started. Such a feature of modern controllers is particularly useful during commissioning of control systems. Besides, autotuning can also be used for the build-up of the table of controller parameters for gain scheduling [3].

This contribution proposes a modification of the basic autotuning approach. The application of the modified algorithm to a wide class of industrial and power plant processes has been described. The modified algorithm has been implemented as a set of program blocks in a programmable logic controller (PLC).

The paper is organised in several sections. The introduction describes the autotuning approach in the adaptive framework. The basic autotuning algorithm and the proposed modifications are detailed in the second section. In the third section, the structure of the modified algorithm is presented. The fourth section gives some experimental results obtained with the modified autotuning PID controller. Finally, the potential uses of the proposed algorithm are pointed out in the conclusion.

2. MODIFICATIONS OF THE AUTOTUNING PID CONTROLLER

The autotuning algorithm, which was originally proposed in the well-known paper [2], is an automated version of the second Ziegler-Nichols experiment. This algorithm follows the idea of performing a simple identification experiment and designing the PID controller on the basis of the information obtained. The difference is that the autotuning algorithm performs these tasks automatically. Instead of using a proportional controller to control the process and reach limit cycle oscillations, the process is controlled by a relay. Such a usage of the relay is possible since most industrial processes having a phase lag of at least -180^0 at high frequencies will enter limit cycle oscillations under relay control. The amplitude and frequency of the limit cycle are measured, which constitute the information used for the design of the PID controller. In [2] a PID controller design based on amplitude and phase margin specifications is proposed. However, many different PID design techniques can be used in the autotuning framework if some modifications are introduced into the algorithm [3].

In spite of many advantages of the basic autotuning algorithm, various issues have to be resolved during its implementation:

- A relay experiment must be performed at the operating point. Consequently, for a successful application of an autotuning algorithm, it is necessary to devise a scheme for reaching the operating point. It can be done as part of the initialisation procedure of the algorithm's supervision shell, through manual control or by an automatic procedure [4].
- Since an autotuning algorithm should not require an operator with expert knowledge, only simple parameters should be used as its input parameters. Furthermore, there should be as few input parameters as possible. These parameters should include an approximate time scale and the desired amplitude of limit cycle oscillations [5].
- Problems may arise due to the usage of a relay element during the identification experiment. The analysis of the relay experiment assumes the existence and uniqueness of limit cycle oscillations, which cannot be guaranteed for all processes. The precise characterisation of the class of processes that converge to a limit cycle oscillation with unique parameters under relay control is still an open problem of control engineering [6].
- In the basic autotuning algorithm the PID controller is designed by using a simple rule for moving the ultimate point of the process to the specified position. Such a procedure gives acceptable results for many processes, but it is possible to achieve better control performance if the employed design rule is optimised for the controlled process. Many modifications of the basic autotuning algorithm implementing the ideas of optimisation of autotuning according to the type of the controlled process have been proposed in the literature [3]. However, these modifications of the basic autotuning algorithm require a-priori knowledge of the type of the controlled process.

Modifications of the basic autotuning algorithm were introduced in order to resolve the above-mentioned issues in the implementation of the autotuning algorithm, to widen the area of possible application, and to make it more robust and accurate. Many proposed modifications of the basic autotuning algorithm use the dynamic element *W(s)* connected in series with the relay when performing the relay experiment. Such an autotuning setup is depicted in Figure 1.

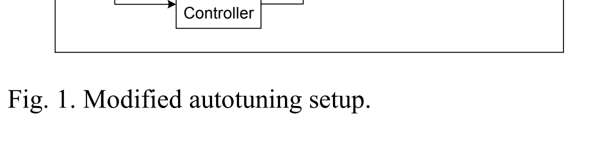


Fig. 1. Modified autotuning setup.

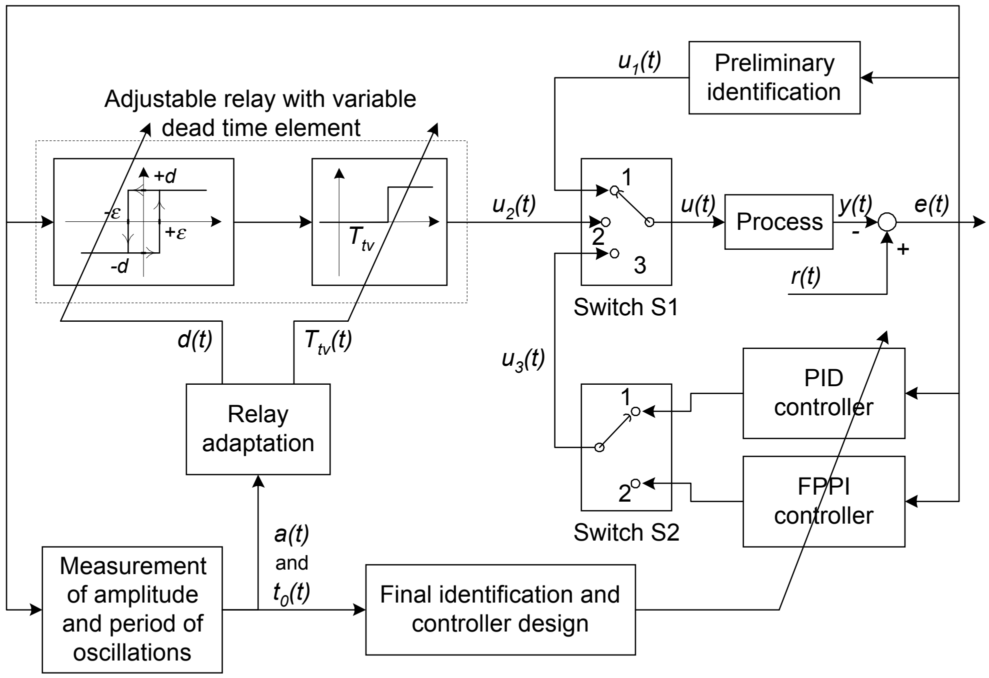


Fig. 2. Structure of the modified autotuning PID controller.

One possible choice of the dynamic element *W(s)* is the choice of an element with variable dead time. Such an autotuning procedure has been proposed in [7], and described in detail in [3]. It allows identification of different points on the process Nyquist curve. For example, when the identified point has a phase angle equal to -135^0, it can be used for the design of a PID controller which is appropriate for the control of processes with a large dominant time constant or for processes with integrator and time lags.

Another modification consists of prefixing a procedure of preliminary identification to the autotuning algorithm, as described in [4]. It is performed in order to determine the type of the controlled process and to make crude estimates of its parameters. The type of the relay experiment, which is used in the ensuing stage of the algorithm, is determined. The values obtained are used for setting the initial parameters of the relay (hysteresis and relay output amplitude). This procedure shortens the time needed for the relay experiment because the relay parameters are already adjusted to approximate the parameters of the process at the beginning of the experiment. Furthermore, the results of preliminary identification can be used for the determination of the process dead time. It is important to note that preliminary identification is made during the phase of reaching the operating point, and it can be viewed as a normal start-up procedure of an autotuning algorithm.

3. STRUCTURE OF THE MODIFIED AUTOTUNING PID CONTROLLER

The structure of the autotuning PID controller with modifications given in the second section is shown in Figure 2. It can be seen that the process control is performed in three phases:

- Preliminary identification;
- Adaptive relay control;
- Control phase.

The switch S1 controls the transition from one phase to another, selecting the appropriate control signal. The first phase of the autotuning controller consists of preliminary identification. In the second phase the relay experiment is performed, and in the third phase a suitably designed controller controls the process. A suitable controller is selected by proper setting of the switch S2.

The relay experiment is the most important and at the same time a very sensitive part of the autotuning algorithm; consequently, it has to be designed carefully. The relay experiment is performed according to the type of the process which is determined during preliminary identification. If a first order with dead time (FODT) process is detected, a non-modified relay experiment is performed [5]. Otherwise, if the process can be modelled with an integrator and a time lag, a modified relay experiment with adjustable dead time is executed [7]. During the experiment the relay amplitude d and the dead time T_{tv} are adjusted simultaneously. Consequently, the relay element and the dead time element have to be implemented in such a manner as to allow simple reconfiguration.

A very important precondition for successful relay experiment is accurate measurement of the amplitude and the period of limit cycle oscillations. It is achieved by employing measures for noise removal and by collecting the oscillation data at appropriate time instants. These measures prevent noise fluctuations being interpreted as oscillations of the process output. For this purpose, a median filter can be used [8].

Another problem associated with the relay experiment is that the adjustment of relay parameters may cause changes in the parameters of limit cycle oscillations. Since limit cycle oscillations gradually change to a new steady state, several oscillation periods after the change of relay parameters should not be included into the measurement. This significantly improves the accuracy of the measurement, but prolongs the relay experiment.

The convergence of the relay adjustment procedure and the permissible deviation of oscillation parameters from the desired values determine the duration of the relay experiment. If the tolerance were too small, the relay adjustment procedure would last too long. Simulations demonstrated that a tolerance of 10% - 20% is a reasonable trade-off between the required accuracy and the duration of the relay experiment.

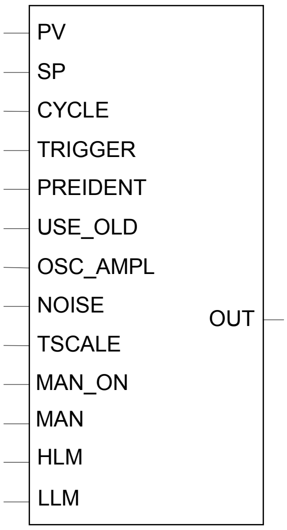
After the completion of the relay experiment, an appropriate control strategy is selected (switch S2) and the respective controller is tuned. If the preliminary identification procedure detected a type of process which can be modelled with an integrator and a time lag, a PID controller is employed and it is tuned according to the tuning method given by [7]. When the process can be modelled with a FODT model, the estimated time constant is compared to the estimate of the dead time. If the comparison shows that the estimated dead time is larger than the estimated time constant, a filtered predictive PI (FPPI) controller is tuned and started [9]. Otherwise, if the estimated time constant of the process is larger than the dead time estimate, a PID controller is used. It is tuned according to the ISTE criterion [10].

4. IMPLEMENTATION OF THE AUTOTUNING PID CONTROLLER

In order to evaluate the features of different autotuning algorithms, and to check the proposed extensions of the basic algorithm, a computer simulation was performed. It allows simple and quick testing of the algorithm behavior for a wide class of model processes. The MATLAB/SIMULINK package was chosen as the programming environment for the computer simulation.

After the completion of the simulation study, a stand-alone software module implementing the extended autotuning algorithm was realized. The modified autotuning PID controller was implemented in a SIEMENS PLC, product family SIMATIC S7-300. It was programmed in the STEP 7 programming environment [11] as a project in structured control language (SCL) [12] and statement list language (STL). The SCL resembles Pascal, while the STL has many characteristics of an assembly language. These languages differ considerably from the simulation environment which was used for the algorithm development. This difference has necessitated some changes in the design of the algorithm.

Different building blocks of the algorithm denoted in the structural scheme (Fig. .2.) are united in one block, the stand-alone software module. The core of the block is a switch which dispatches calls to appropriate subroutines according to the state of the algorithm. These subroutines implement basic parts of the algorithm: preliminary identification, relay experiment, relay adjustment, calculation of controller parameters and controllers (PID and FPPI). The stand-alone software module is depicted in Figure 3.



Block inputs:
PV – process variable
SP – set-point
CYCLE – sampling time
TRIGGER – trigger signal for starting algorithm
PREIDENT – flag for using preliminary identification
USE_OLD – flag for using values from previous run of the algorithm
OSC_AMPL – desired amplitude of relay oscillations
NOISE – noise band
TSCALE – approximate time scale
MAN_ON – flag for manual input
MAN – value of manual input
HLM – high limit of the control signal
LLM – low limit of the control signal

Fig. 3. Stand-alone autotuning software module in STEP7 programming package.

The autotuning algorithm is implemented in the Cycle-interrupt organisation block which is called at regular intervals by the operating system of the PLC. Since the algorithm requires the implementation of the dead time element for the FPPI controller, it has to be adaptable to a large scale of possible values of the process dead time. The limited memory resources of the PLC dictate the usage of variable sampling time to achieve this goal. Variable sampling time is implemented by an internal counter for jumping over the required number of sampling steps.

5. EXPERIMENTAL RESULTS

The first laboratory process on which the autotuning controller was tested is the laboratory blower process (Fig. 4.). The main parts of this process are the heater, fan, air influx valve, temperature sensor, and measurement electronics.

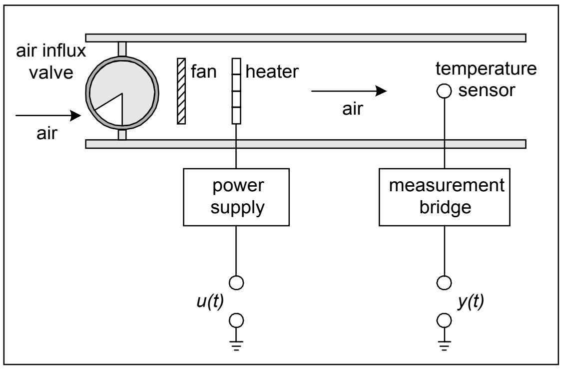


Fig. 4. Laboratory blower process.

The results of the experiment are shown in Figure 5. In this experiment the autotuning controller successfully performed the tasks of preliminary identification, relay experiment and PID controller tuning. The adjustment of the relay during the experiment lasted only four iterations, thus confirming the quick convergence of the procedure.

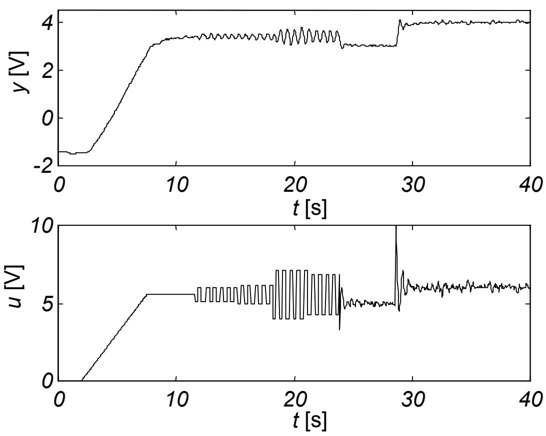


Fig. 5. Autotuning experiment on the laboratory blower process.

The controller parameters during the autotuning procedure showed that the controller adapted its behavior according to the process parameters. In addition, this experiment demonstrates that the controller was able to complete the procedure in spite of the presence of a strong noise signal, exhibiting good robustness properties.

Another experiment involved the following process:

$$G_P(s) = \frac{e^{-sT_t}}{(3s+1)(s+1)(0.5s+1)},$$

where T_t = 1s, and which was set on the laboratory process simulator. The response of the control system is shown in Figure 6. This process was classified as a FODT process, and its approximate parameters were estimated by the preliminary identification procedure. In the control phase, a PID controller was employed. The overall performance of the autotuning controller was acceptable: relay adjustment was short and the PID controller obtained displayed satisfactory behaviour, although a somewhat too oscillatory. Therefore, this controller can be employed for similar stable, non-oscillatory processes.

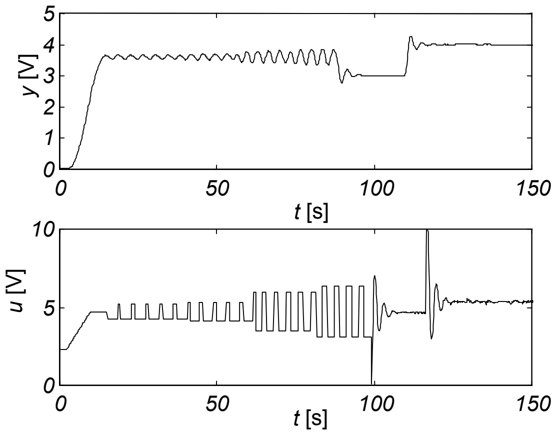


Fig. 6. Autotuning experiment on the process simulator.

The experiments presented show that all tasks of the autotuning algorithm can be performed successfully by PLC implementation. The preliminary identification procedure properly detects the type of the process, choosing the type of relay experiment accordingly. Relay experiments converge to limit cycle oscillations with the desired parameters after only a few iterations of the adjustment procedure. The PID and FPPI controllers obtained exhibit good control behviour.

6. CONCLUSION

The proposed modification of the autotuning PID controller enlarges the class of processes to which an autotuning controller can be applied. It adjusts the control algorithm of the controller to the process, and thus, improves the behaviour and the robustness of the control system. These properties of the controller are especially noticeable when the controlled process has a large dead time, and a FPPI control algorithm is employed. The experimental results obtained with PLC implementation of the autotuning PID controller confirm the good properties of the algorithm.

7. REFERENCES

[1] Åstrom, K.J. (1996). Tuning and Adaptation, IFAC 13th Triennial World Congress, San Francisco, USA, pp. 1-18.

[2] Åstrom, K.J., Hägglund, T. (1984). Automatic Tuning of Simple Regulators with Specifications on Phase and Amplitude Margins, Automatica, Vol. 20(5), pp. 645-651.

[3] Branica, I. (1999). Development and Applications of Autotuning PID Controller, Master thesis, Faculty of Electrical Engineering and Computing, Univ. of Zagreb, Croatia, URL: http://www.rasip.fer.hr/act/

[4] Branica, I., Petrović, I., Perić, N. (1999). Towards Industrial Autotuning Controller, Proc. of 3rd Int. Conf. on Systems - IMACS/IEEE CSCC'99: Progress in Simulation, Modeling, Analysis and Synthesis of Modern Electrical and Electronic Devices and Systems, Athens, Greece, pp.136-141.

[5] Perić, N., Petrović, I., Branica, I. (1997). A method of PID controller autotuning, Proc. of IFAC-IFIP-IMACS Conf. on Control of Ind. Sys., Belfort, France, Vol. 2, pp. 43-48.

[6] Johansson, K.H. (1997). Relay Feedback and Multivariable Control, PhD thesis, Lund Institute of Technology, Lund, Sweden.

[7] Voda, A.A., Landau, I.D., (1995). A Method for the Auto-calibration of PID Controllers, Automatica, Vol. 31(1), pp. 41-53.

[8] Petrović, I., Perić, N., Branica, I. (1998). Autotuning PID Controller with Supervision Shell, Proc. of the 17th IASTED Int. Conf., Grindelwald, Switzerland, pp. 342-344.

[9] Normey-Rico, J.E., Bordons, C., Camacho, E.F., (1997). Improving the Robustness of Dead-time Compensating PI Controllers, Control Eng. Practice, Vol. 5(6), pp. 801-810.

[10] Zhuang, M., Atherton, D.P. (1993). Automatic tuning of optimum PID controllers, IEE PROC.-D, Vol. 140(3), pp. 216-224.

[11] SIEMENS AG (1997). STEP 7 Program Design (Programming Manual), Nürnberg, Germany.

[12] SIEMENS AG (1996). Structured Control Language (SCL) for S7-300/S7-400 Programming, Nürnberg, Germany.