Smart Synthesis of a PID Controller

Sallé, Stéphane; Åström, Karl Johan

1990

Document Version:
Publisher's PDF, also known as Version of record

Link to publication

Citation for published version (APA):
Sallé, S., & Åström, K. J. (1990). *Smart Synthesis of a PID Controller*. (Technical Reports TFRT-7462). Department of Automatic Control, Lund Institute of Technology (LTH).

Total number of authors:
2

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

CODEN: LUTFD2/(TFRT-7462)/1–6/(1990)

Smart Synthesis of a PID Controller

Stéphane S. Sallé
Karl Johan Åström

Department of Automatic Control
Lund Institute of Technology
August 1990

Department of Automatic Control
Lund Institute of Technology
P.O. Box 118
S-221 00 Lund Sweden

Document name
INTERNAL REPORT

Date of issue
August 1990

Document Number
CODEN: LUTFD2/(TFRT-7462)/1–6/(1990)

Author(s)
Stéphane E. Sallé
Karl Johan Åström

Supervisor

Sponsoring organisation

Title and subtitle
Smart Synthesis of a PID Controller

Abstract

A new auto-tuner for PID controllers is described. It combines two commonly taken approaches: analysis of process transient responses and estimation of the process critical point through a relay feedback experiment. The system selects the most appropriate controller among a PI, a PID or a PI regulator coupled with a Smith predictor. The controller parameters are based on estimation of the normalised process dead-time and modified Ziegler-Nichols rules. The system is implemented on a Sun workstation and uses a real-time expert system developed with Muse. The paper describes the methods used and gives simulation results.

Key words

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title

ISBN

Language
English

Number of pages
6

Recipient's notes

Security classification

The report may be ordered from the Department of Automatic Control or borrowed through the University Library 2, Box 1010, S-221 03 Lund, Sweden, Telex: 33248 lubbis lund.

1 -

To be presented at the IFAC International Symposium on Intelligent Tuning and Adaptive Control,
15 - 17 January 1991, Singapore.

Smart Synthesis of a PID Controller

Stéphane E. SALLÉ
Laboratoire d'Automatique de Grenoble
B.P. n° 46
F - 38 402 Saint-Martin d'Hères
France

Karl-Johan Åström
Department of Automatic Control
Lund Institute of technology
P.O. Box 118
S - 221 00 Lund
Sweden

Abstract. A new auto-tuner for PID controllers is described. It combines two commonly taken approaches : analysis of process transient responses and estimation of the process critical point through a relay feedback experiment. The system selects the most appropriate controller among a PI, a PID or a PI regulator coupled with a Smith predictor. The controller parameters are based on estimation of the normalised process dead-time and modified Ziegler-Nichols rules. The system is implemented on a Sun workstation and uses a real-time expert system developed with Muse. The paper describes the methods used and gives simulation results.

Key Words. PID control; normalised dead-time; relay feedback; intelligent control, real-time expert system.

1. INTRODUCTION

Despite many advances in control theory, most industrial control loops are still based on PID controllers. Significant efforts have lately been devoted to the automatic tuning of such regulators ; a discussion on this subject can be found e.g. in [*Åström & Hägglund 1988*]. Two ideas are commonly employed for automatic tuning. One approach is based on an open-loop or closed-loop transient response analysis, as in the Foxboro "Exact" controller [*Bristol 1977*]. A second approach is based on analysis of the process response under relay feedback as in the Satt Control controller ECA 40. Both systems use traditional tuning methods of PID controllers in the Ziegler-Nichols spirit [*Ziegler & Nichols 1942*].

The two approaches are combined in this paper. A relay feedback experiment gives an estimation of the process critical point and allows the design of a crude PI controller. Then a pattern recognition is performed for a closed-loop and an open-loop step response. The results of the experiments provide a rough model of the process: dead-time, apparent time-constant, static gain, order (first order or not). Then the controller parameters are selected based on the refined Ziegler-Nichols tuning formula presented in [*Hang & Åström 1988*].

The paper is organised as follows: The statement of the problem is first given. Experiences of many different methods tested, modified and used are described in section 3 and simulations are presented in section 4. Comparison of the proposed controller with two other controllers is presented in section 4 before the conclusions in section 5.

2. PROBLEM DESCRIPTION

A description of a new generation of PID auto-tuners is given in [*Åström et al 1989*]. These systems have reasoning capabilities which help them to "smartly" select the parameters of a controller. Based on knowledge of the normalised process dead-time θ (which is defined as the ratio of the process apparent time-delay L over the process apparent time constant T as shown in figure 2.1, i.e. $\theta=L/T$), the difficulty of controlling the treated process is estimated and some heuristics rules allow to select among a PI, a PID or a PI controller coupled with a Smith predictor. In [*Hang & Åström 1988*], this normalised dead-time is also used in order to refine the Ziegler-Nichols tuning formula. To build such an auto-tuner, the main problem is to estimate the normalised dead-time. Several ways to do this are discussed.

Within this study, the process is assumed to have one input and one output, to be linear and stable (no integrator), to have a global monotone step response except possibly at the beginning and to give stable oscillations under relay-feedback. The class of systems considered is roughly similar to the one considered in the classical works on Ziegler-Nichols tuning. Such systems can be characterised by three parameters : static gain Kp, apparent dead-time L and apparent time constant T (cf figure 2.1).

The approach used in this paper consists of determining the critical point and the parameters Kp, L and T in an autonomous manner, requiring very little a-priori knowledge of the process. These data are then used with a refined Ziegler-Nichols method.

The operator is asked the range of the process rise-time, the magnitude order of the static process gain, and the minimum and maximum values of the process input. The approximate value of the process rise-time is necessary to check the steady state of the process (which is assumed to be stable) before performing the relay experiment. In commercial controllers, the static process gain is often of the order of one. It is one task of the process engineer to set the static gain of the entire process (actuator + process + sensor) in the range of 0.2 and 5. Some a-priori settings of the system assume the same hypothesis. If this hypothesis is not true, a rough estimation is required in order to reschedule the processed data (process input, output and reference). An anti-windup device for the integral part of the developed controller is included. It requires knowledge of the maximal values of the process input.

After this preliminary stage, the operator brings the process manually to the desired operating point and three experiments are performed. After the steady state of the process is checked, a relay experiment is performed. During the first two oscillations, the relay parameters can be changed in order to get a significant variation of the process output of reasonable size, typically three times the noise level of the process output. After this phase, the period and magnitude of the oscillations are measured to get an idea of the process critical point. At the same time, estimations are made of the maximum process delay and of first order and second order models of the process, based on analysis of the wave form. A PI controller, based on the Ziegler-Nichols tuning formula, is designed with the obtained information. This regulator will be used later as a safety regulator.

Then a closed loop step response is performed and a precise estimation of the static process gain is obtained. This experiment is followed by an open loop step response which gives a first order model with time delay of the process. In case of too large process errors, the safety PI is switched on. The experiment is stopped when the process has reached 63% of the final value. The PI controller is then used to bring the process rapidly into steady state at the operating point initially fixed by the operator. Different treatments are also performed during the step responses. They are described in the third section.

System Architecture

The system consist of procedural algorithms written in C supervised by a knowledge based system. Such an architecture is described in [*Åström et al 1986*] or in [*Årzén 1987*]. The algorithms are monitored by an expert system developed with Muse (a development tool for real time expert systems which is briefly described in [*Sallé 1989*]). The use of an expert system gives a much cleaner auto-tuner implementation than a procedural language would give, mainly because of the clear separation of algorithms and logic. Another benefit due to the modularity of the expert system is the ease of changing or incrementing the associated logic. The real-time features of Muse are also helpful. The two different computer-processes are linked through a Unix socket. Muse supports this communication.

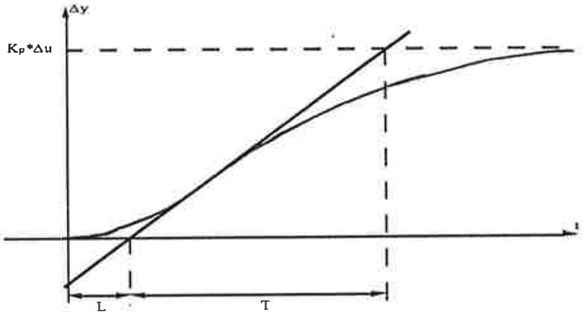

Figure 2.1 Typical step response of considered processes.

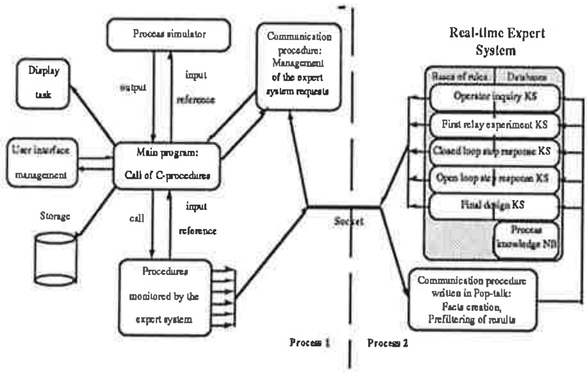

Figure 2.2 System architecture.

The Expert System

Muse from Cambridge Consultants in U.K. is a toolkit for embedded real-time artificial intelligence. It consists of an integrated environment for knowledge representation with an object structure and databases. The Pop-talk language is the central component of this package. Pop-talk is implemented in C and is derived from the Pop series of languages. It has been extended to support object-oriented programming. It also provides a stack-based language that combines list-processing elements with a block-structured syntax as in C or Pascal. On the top of the basic object language, a frame (or schema) system is built which includes multiple inheritance, methods, relations and demons.

Muse applications range from a simple expert system with a single database and rule-set to a complete blackboard system with many knowledge sources (KS) and databases (DB) which co-operate to solve the problem. This possibility is used in the developed system since the set of developed rules is split into five knowledge sources. Each knowledge source is associated with a precise task : Operator interaction, performing the relay experiment, performing the closed-loop and the open-loop step response, computing and testing the final controller. All the knowledge acquired is stored in a notice board.

The present version of the expert system comprises 70 rules. Only one knowledge source is active at a time. Its activation is done in a defined order. This is implemented using objects which have a slot containing the present level of reasoning. The task of each KS consists of bringing the system from one state to another state. To do this, each KS fires the desired procedures, performed tests on the process and elaborates conclusions. Intermediate facts are created in the DB of the active KS, but the final conclusions and results obtained by this KS are stored in the separate notice board. We think that it is a good way to structure the system. For example, to restart a task, the affected KS can be reset without destroying the knowledge already acquired on the process.

Procedure Libraries

The procedures monitored by the expert system are written in C. They include a PID regulator, digital filters, an implementation of a relay with hysteresis with automatic adjustment of its amplitude and hysteresis, relay oscillation analysis, statistic computation routines, pattern recognition routines, etc. These procedures also incorporate a process simulator, a user interface and display tasks. The expert system gives initial parameters to the procedures, it receives and stores the obtained results. The main procedures will be further described in section 3.

Each request made by the expert system is associated with a procedure. A request, sent by the expert system on the socket, is characterised by the first part of the message. The socket is polled and its content is read and analysed through an "if ... else if ..." structure in the communication procedure. The parameters of the request are read and the associated procedure is initialised. The main programme calls the procedure until it is completed or if Muse asks it to be cancelled.

3. METHODS

The different methods used in the system will now be described. Their concepts are shortly explained. It is shown how they are implemented. It is also attempted to point out their advantages and disadvantages based on our experience with the system.

Testing For Steady State

Before performing an experiment, the system checks if the process is in steady state. This is done by investigating if the process output is close to a constant and if the variations around this value are almost constant. The precondition for the test is knowledge about the time scale of the process. This is based either on the process rise time Tr, as given by the operator, or the relay oscillation period Tu. Three time intervals of length Tr/3 or Tu are used in the test. The mean value and the standard deviation of the measured output and the control error are estimated on the first and on the last interval. Depending on the desired confidence, the variation of the mean (and standard deviation) estimates must be less than 5% or 10% (10% or 20%). The mean value of the error must also be less than the deviation in the measured variable and the process error must be approximately zero. The thresholds are set to allow very small measurement noise (typically less than 0.01%). The permissible variations of the standard deviation estimates are larger than the ones allowed for the mean values due to the estimators' characteristics. The measurement noise is determined as the difference of the maximum and minimum values of the process output. These simple tests have been found, in the authors' experiments, very effective to determine if the process output is in steady state.

Relay Tuning

The idea of relay tuning is to introduce a relay in the feedback loop. For a large class of processes, there will be a limit cycle oscillation. The amplitude and the period of the oscillations give information about the process dynamics that can be used to compute the appropriate controller parameters. The idea of using a relay for tuning purposes is described in details in [*Aström & Hägglund 1988*]. The process information obtained from a relay experiment is essentially knowledge about one point on the open-loop Nyquist curve of the process. In many cases (if the system has a phase lag of at least π at high frequencies), this point is at or close to the intersection of the Nyquist curve with the negative real axis. This point is traditionally described in terms of ultimate gain Ku, and ultimate period Tu. Ziegler and Nichols gave a method to determine PID parameters based on knowledge of this critical point.

When a limit cycle oscillation is established, the output is a periodic signal with period Tue which is close to Tu. If d represents the relay amplitude and ε the relay hysteresis, it follows from a Fourier series expansion that the first harmonic of the relay output has the amplitude 4d/π. If the amplitude (maximum value minus minimum value) of the relay input (which is the control error) is 2a, the ultimate gain is thus approximately given by:

$$K_u \approx \frac{4\,d}{\pi\,a} \text{ since } a \approx \frac{4\,d}{\pi}\left|G(i\omega_u)\right| \text{ and } K_u = \frac{1}{\left|G(i\omega_u)\right|}.$$

The advantages of this method are:

* it requires little a-priori knowledge of the process,

* the estimation method is easy to implement since it is based on counting and comparisons only,

* it is easy to control the oscillation amplitude by an appropriate feedback choice of the relay characteristics d and ε,

* it is robust to measurement noise and the use of a well chosen hysteresis improves its robustness.

Three parameters associated with the relay experiment must be set : the relay amplitude, the relay hysteresis and the desired amplitude of the output oscillations. The value of the relay hysteresis is determined from the measurement noise : more noise gives a larger hysteresis in order to avoid erroneous relay switching. A minimal value of the hysteresis is prescribed to avoid problems with fast relay oscillations as e.g. with first order processes without time delay. The value of the desired oscillation amplitude is also determined from the noise level. As a protection against processes with very high static gain and short time delay, during the first half period of the oscillation, the relay amplitude is increased linearly from zero until either it reaches a default amplitude (which is presently set to 0.2), or the error signal exceeds the desired amplitude.

The amplitude d and the hysteresis ε may be adjusted separately to obtain the desired oscillation amplitude. Notice that an analysis based on harmonic balance gives:

$$\arg\left[\widehat{G(i\omega_{u\omega})}\right] \approx -\pi + \arcsin(\frac{\varepsilon}{a})$$

The ratio ε/a determines how far from the negative real axis the estimated point is. A possible adjustment law consists in fixing the desired phase lag and to adjust ε. It is not presently done.

It typically takes from 4 to 7 half periods to reach a steady state limit cycle oscillation. The actual number depends on whether the relay parameters are adjusted or not. The oscillation is said to be stable when the relative variation between the last two peak-to-peak amplitudes is less than 10%. The oscillation analysis takes two periods. During the first period, the peak amplitudes and the oscillation half periods are measured. During the second period, regularly spaced points are selected on the oscillation curve in order to analyse the wave form (this is explained in the next section).

Wave Form Analysis

The shape of the oscillation under relay feedback may be used to estimate a process model [*Åström & Hägglund 1988*]. The idea of this method is that the process input and output are periodic signals. Regularly spaced samples $y_0, y_1, \ldots, y_{2n-1}$ are chosen as shown in figure 3.2. Based on Z-transform properties, it can be shown that the coefficient of the standard input/output model $A(q)\ y(k) = B(q)\ u(k)$ can be determined from the equation:

$$A(z)\ D(z) + z^r\ B(z)\ E(z) = z^r\ (z^n + 1)\ Q(z)$$

where $r = \deg A(q) - \deg B(q)$, d is the relay amplitude and

$$E(z) = z^n + z^{n-1} + \ldots + z,$$

$$D(z) = (y_r\, z^n + y_{r+1}\, z^{n-1} + \ldots + y_{r+n-1}\, z)\ .\ d$$

The polynomial Q corresponds to initial conditions which give a steady state periodic output. For example, with $n = 3$, a model $G(s) = K_p\, e^{-s\,L} / (1 + s\,T)$ can be determined. The calculations and formula can be found in [*Åström & Hägglund 1988*].

Remarks: This method works very well for a first order process if there is **little process noise**. In discrete time implementations, it is also important that the times associated with the 2n regularly spaced samples correspond exactly to a measured sample, unless the sampling period is very small compared to the process oscillation period. If this is not the case, the errors of the estimated continuous time parameters may be very large (from 10% up to 50%). The method works well for thermal processes where the noise level is small and because such processes are well approximated by first order dynamics with time delays. The method does not work well at high noise levels. To get good estimates in such a case, it is necessary to increase the relay amplitude. The determination of high order models has proven to be difficult. With n equal to five, it was found to be very difficult to determine second order models.

Minimum phase systems with a relative degree of one is an interesting class of systems that can be successfully approximated by a first order system. A controller can be designed using special techniques such as a constant high gain proportional feedback. In theory, such processes can be detected by examining the value of the process output derivative at the relay oscillations extreme point. Such an analysis is performed in the system. The slopes before and after the extreme point are estimated based on 3, 5 or 7 points of the curve (depending on the noise level). The mean values of the slopes computed on each extreme are sent to the expert system. Based on the amplitude of the variation of these values and on the comparisons with the other results, the expert system deduces if the system belongs to this class. If it is the case, special rules for synthesising the controller are used.

The mean value of the times required to reach the extreme is also sent to the expert system. It is used to estimate the maximum process time delay.

Method of Moments

This method is described in [*Åström & Hägglund 1988*]. It consists of estimating a process model from the values of the transfer function and its derivatives at $\omega = 0$. If the process model is:

$$G(s) = \frac{K_p e^{-sL} Q(s)}{P(s)} = \frac{K_p e^{-sL} \Pi_{i=1}^{q} (1 + q_i\, s)^{m_i}}{\Pi_{i=1}^{p} (1 + p_i\, s)^{n_i}},$$

an expression of $G^{(n)}(0)$ can be established [*Sallé 1990*] and:

$$G(0) = K_p,\ \frac{d\,G}{ds}(0) = -\,G(0) * \left[L + \sum_{i=1}^{p} n_i\, p_i - \sum_{i=1}^{q} m_i\, q_i \right], \text{etc} \ldots$$

The value of the transfer function and its derivate at $s = 0$ can be computed from the following equations:

$$Y(0) = G(0)*U(0),\ \frac{d\,Y}{ds}(0) = \frac{d\,G}{ds}(0)*U(0) + G(0)*\frac{d\,U}{ds}(0), \text{etc} \ldots$$

and from the following integrals:

$$U(0) = \int_0^\infty u(t)\,dt,\ U^{(n)}(0) = \int_0^\infty t^n u(t)\,dt \text{ and } Y(0) = \int_0^\infty y(t)\,dt,\ Y^{(n)}(0) = \int_0^\infty t^n y(t)\,dt$$

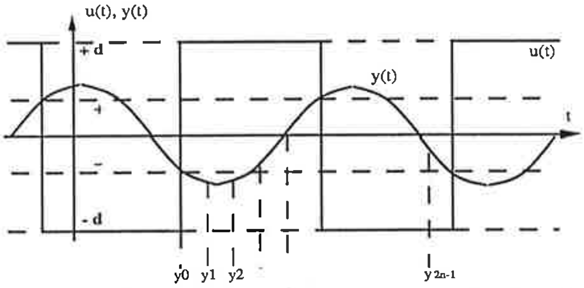

Figure 3.1 Input and output given by a process under relay feedback.

This method can be used with arbitrary input signals. However, for the computations of the integrals to be truncated in time, the input and the output level must be the same at the beginning and at the end of the experiment.

This method works perfectly in theory. It should be robust to noise since it is based on computations of integrals. However, the estimation of a simple first order model with time delay requires the computation of G"(0) and Y"(0). Due to the t^2 term, these computations led to numerical problems for long transients. Due to errors in estimation of G"(0), the results are poor even at low noise levels. We have had good experience in estimating process gain Kp and the sum T + L for first order systems. Combined with the estimates of Ku and Tu from the relay experiment, this gives a good method for determining the parameters Kp, T and L of a first order system. This method is used during the relay experiment and the two step responses. The results are sent to the expert system and their accuracies are estimated based on the estimated measurement noise and on the value of the input integral.

Pattern Recognition

Information about the process can also be deduced from pattern recognition of transient responses. In this way, it is possible to determine the static process gain and the dead-time from a closed loop step response. It is also possible to detect a non-minimum phase system. This is done by checking that the closed loop system is in steady state. A change in the set point is then performed. When the process is in steady state, the static gain is evaluated as the ratio of the variation of the process output mean values and the variation of the process input mean values. The difference between the first significant variation of the input and the first significant variation of the output gives a rough estimate of the process dead-time.

When the estimation of the static process gain is performed, the PI is turned off and an open loop step response is performed. The amplitude of the process input step is computed in order to get the same final value of the process output as before the closed loop step response. A maximum error limit is set based on estimation of measurement noise. As soon as the error is greater than this value, the PI controller is turned on and the experiment is stopped. It is restarted when the process is considered again to be in steady state. A rough estimate of the process dead-time is computed and a non-minimum phase characteristic is detected. The inflection point of the step response is determined as the point with maximum slope. This slope is estimated based on 3, 5 or 7 points of the process output curve, depending on the noise level. This experiment is finished when the output has reached 63% of its final value and the maximum slope has been reached several points before. The PI controller is then switched on to bring the system to steady state at the desired level. The apparent dead-time (cf figure 2.1) is estimated and two estimates of the apparent time constant are computed. One is based on the slope computation (cf figure 2.1) and the other on the time it takes to reach 63% of the final value. Notice that this operation is quite safe since the process output is confined to a predetermined band when the loop is opened.

Special calculations are done to estimate a precise discrete time first order model of the process. The step response of such a process has no inflection point since the slope is maximal at the beginning of the response. Since we are working in discrete time and using more than two points for the estimation of the slope, a correction must therefore be done [*Sallé 1990*].

An Asymmetrical Relay

The relay experiment essentially gives information about one point on the Nyquist curve. This is equivalent to two parameters Ku and Tu. It is possible to determine a reasonable PID controller based on this implementation. To obtain a fine tuned PID controller, it is however useful to know a third parameter e.g. the static process gain. This can be determined from open or closed loop step tests as it has been discussed. It would, however, be highly desirable to determine all parameters from a single experiment. This can be

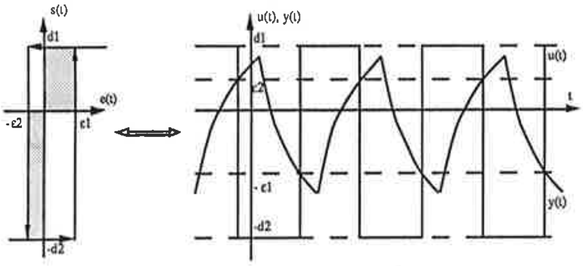

Figure 3.2 An asymmetrical relay characteristic function and an associated time response.

obtained by using an asymmetrical relay as the one shown in figure 3.2. Depending on the placement of the origin in the rectangle of the relay characteristic function, three main types of asymmetrical relays can be defined. The first kind of asymmetrical relay consist of keeping the vertical axis in the middle of the rectangle and shifting the horizontal axis towards the negative or positive values. This kind of relay is equivalent to take a symmetric relay and to add a bias to the relay input. Another type of asymmetrical relay consist of shifting both vertical and horizontal axes in the same direction. The last type of asymmetrical relay (which is displayed in figure 3.2) consist of shifting both vertical and horizontal axes in the two opposite directions. This last type has been chosen since, when tested with several processes, it gave the most symmetric (in time) oscillations and the biggest absolute value of the input integral (this is a necessary condition to rely on the results given by the method of moments). The first type gives a small value of the input integral and the second type gives the most asymmetrical (in time) oscillations. The usefulness of using two different hysteresis is shown in [*Sallé 1990*] in the case of a first order process.

The static gain may be obtained either as the ratio of the mean values of input and output or with the method of moments. An analysis based on harmonic balance gives an estimation of the ultimate gain. The equations used in the developed system are fully described in [*Sallé 1990*]. Experiments with many different processes showed that the estimates of the critical point obtained with an asymmetrical relay were quite similar to those obtained with a symmetric relay. Moreover, when the measurement noise is not too large, the estimate of the static gain is accurate. A drawback of this method is that it requires symmetric oscillations if a normal relay is used. Should it not be the case due to, for instance, a nonlinear process, the previous reasoning is erroneous. In [*Hang & Åström 1987*], it is shown how the standard experiment may be modified to obtain symmetrical oscillations.

The method presented here may be employed when a process identification is performed by using a pseudo random binary sequence. Usually, this sequence is centred around zero. If its mean value differs from zero and if the final output is close to the initial output, a precise estimation of the static process gain may be obtained by the method of moments.

Detection of Outliers

Outliers are particularly detrimental for controllers with derivative action. A simple scheme to detect outliers is therefore introduced. It just consist of detecting measurements that are incompatible with the next and previous measurement values. If at time t a process output is larger than expected (e.g. the output at time t-1 plus two or three times the measurement noise level), the output at time t+1 is compared to the output at time t-1. If these data are almost equal (e.g. their difference is smaller than the measurement noise level), a false measure is considered to be detected and the process input at time t+1 is set to the opposite of the input at time t. This simple device is very effective as it will be shown in section 4. It is especially true if the process has higher order dynamics and a rather long dead-time.

Combination of the Different Approaches

The expert system gathers all the results obtained from the C procedures.It analyzes them to find a consistent process model.

The two models obtained by the method of moments are evaluated. The static process gain is first estimated by comparing the available estimations. The estimate given by the closed loop step response normally gives the best accuracy and emphasis is put on it. Depending on the experimental conditions, an estimation of $G'(0)$ and $G''(0)$ may be also obtained. These data give an idea about the process time delay and the sum of the process time constants.

The process time delay is then estimated as the time when the output starts to change after a variation in the process input or as the apparent time delay (cf figure 2.1). The latter may be larger particularly if the process is non-minimum phase. Such a feature may be detected at this stage.

Based on the slope analysis of the relay oscillations, on the comparison of the different models, and on their estimated time delays, the process is classified as being well approximated by a first order process (type 1) or not well approximated (type 2). The process time constant is then estimated, depending on the type of the process. The estimate of the ultimate gain Kue is recomputed in the case of type 1 processes in order to reduce the committed error on this data (which may be bigger than 50%).

An estimate of the normalised process dead-time θ and the normalised process gain κ (which is defined as Kp*Ku) is then obtained. These two parameters are dimensionless and experiments have shown that processes with a small θ or a large κ are easy to control and processes with a large θ or a small κ are difficult to control [*Åström et al 1989*].

The final controller follows the recommendations given in [*Åström et al 1989*]. Set point weighting is introduced to reduce the overshoot. A derivative part is added to the controller when the noise level is said to be low and when θ is smaller than one.The parameters of the controller are based on the refined Ziegler-Nichols tuning formula stated in [*Hang & Åström 1988*].

Computation of the Regulator Parameters

Depending on the value of the normalised process dead-time θ, four cases are considered for the synthesis of the controller. The used formula are given without any explanations. The interested readers must refer to the relevant references in order to get further details. T_{ue} represents the relay oscillation period, K_{ue} the estimate of the ultimate process gain. The factor N introduced in order to filter the derivative action is arbitrarily set to 8. The set point weighting consists of a ß factor introduced before the reference signal in the proportional term of the controller.

$\theta \leq 0.15$:

In this case, the process is declared to be easy to control and rather well approximated by a first order process. Since Ziegler-Nichols tuning may not give the best results in this case, a new way of designing a PI controller is tested. This design tries to take profit of the available power in the following way. In the case of a first order process (with the transfer function Kp/(1 + T*p)) coupled with a PI controller (with the transfer function Kr + 1/(Ti*p)), the poles of the closed loop transfer function can be fixed such that they correspond to a relative damping of 0.707 and a natural frequency ω_n equal to n/T rd/s. The factor n measures how fast the closed loop is. Some calculations leads to:

$$Kr = \frac{n\sqrt{2} - 1}{Kp} \text{ and } Ti = T\,\frac{n\sqrt{2} - 1}{n^2}.$$

The settings we chose, are that a set point change of an amplitude equal to half the peak-to-peak amplitude of the relay oscillations, creates an immediate, and often maximum change, in the process input equal to 70% of the available power. This last quantity is estimated as the difference between the present process input and the closest input extreme value. These requirements give a value of Kr since it may be assumed that, in case of a set point change, the initial variation of the control signal is mainly due to the proportional part of the controller. From knowledge of Kp and Kr, the factor n is deduced. Its value is then restricted to the interval [0.5 ; 3]. Eventually, Ti is computed since both T and n are known. The factor ß is set to one.

A derivative part is added if the estimated noise level is said to be low (smaller than 0.005). The time derivative follows the recommendation given in [*Åström et al 1989*]: Td = Ti/8.

$0.15 < \theta \leq 0.6$:

This is the prime application area for PID controllers with Ziegler-Nichols tuning.

If the noise level is smaller than 0.01, a PID controller is used and its parameters are given by the well known relations:

$Kr = 0.6*Kue$, $Ti = 0.5*Tue$ and $Td = Ti/4$.

If the noise level is greater than 0.01, a PI controller with the following parameters is used:

If $\theta \leq 0.4$ then $\mu = -1.2*\theta*\theta + 1.3*\theta - 0.11$
else $\mu = -0.15*\theta*\theta + 0.33*\theta + 0.11$.
If $\theta \leq 0.4$ then $\Omega = 0.16*\theta^{-0.87}$ else $\Omega = 0.25*\theta^{-0.37}$.
$Kr = \mu*Kue$, $Ti = \Omega*Tue$ and $Td = 0$.

ß is set to $0.1+5*\theta/3$ if $\theta \leq 0.3$ and to $0.3+\theta$ otherwise.

$0.6 < \theta \leq 1.0$:

Ziegler-Nichols tuning becomes less useful. It is recommended to introduce some dead-time and possibly feedforward compensation devices.

If the noise level is smaller than 0.01, a PID controller is used:

$Kr = 0.6*Kue$, $Ti = 0.5*(1.5 - 0.8*\theta)*Tue$ and $Td = Tue/8$.

If the noise level is greater than 0.01, a PI controller with the following parameters is used:

$\mu = -0.15*\theta*\theta + 0.33*\theta + 0.11$, $\Omega = 0.25*\theta^{-0.37}$.
$Kr = \mu*Kue$
$Ti = \Omega*Tue$ and $Td = 0$

The ß factor is set to $1.6 - \theta$ if $\theta \leq 0.8$ and to 0.8 otherwise.

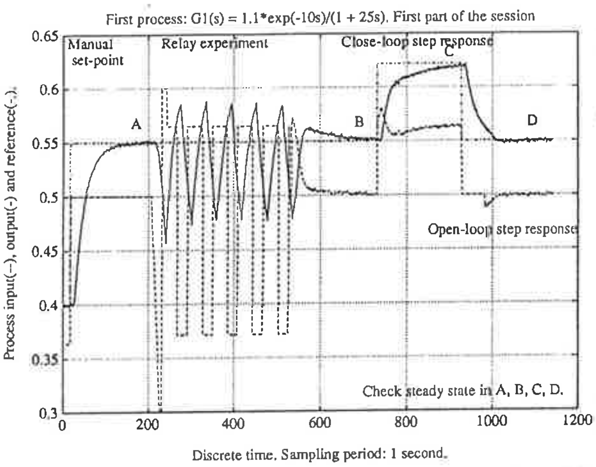

Figure 4.1 Process input, output and reference obtained during an entire session with the developed system.

$1.0 < \theta$:

The process is said to have a long dead-time. It is then essential to introduce some dead-time and possibly feedforward compensation devices. This can be done for instance with a pole placement or a Smith predictor. The latter structure has been chosen. The parameters of the PI controller used with this structure are given by the relations obtained with a very small θ. However, due to the process characteristic, the factor n, related to the closed loop's speed, is restricted to the interval [0.5 ; 1.5]. Notice that no extensive test of this way of setting the PI parameters has been performed.

4. SIMULATION RESULTS

Figure 4.1 shows a typical experiment performed with the developed system. The four different phases previously described can easily be identified. A first order process with Kp = 1.1, L = 10 s and T = 25 s is used. The estimated value of θ is 0.395 which must be compared with the theoretical value 0.4. The estimated value of κ is 4.63 which must be compared with the theoretical value 4.59. During all the showed simulations, a Gaussian centred white noise with a standard deviation of 0.001 is added to the process output.

The next two tables display the results given by the developed system obtained with two different processes. Their transfer functions are equal to:

$$G_1(s) = \frac{K_{p1} e^{-Ls}}{1 + T_1 s} = \frac{1.1\, e^{-Ls}}{1 + 25\, s} \text{ and } G_2(s) = \frac{K_{p2} e^{-Ls}}{(1 + T_2 s)^2} = \frac{0.85\, e^{-Ls}}{(1 + 20\, s)^2}$$

The tables report the results obtained for different values of L. The last column states if the tested process is considered as a first order process or not. The errors committed on the estimation of the process normalised delay are bigger with the second process than with the first process. This may be explained by the fact that this data has no theoretical value in the latter case. Indeed, the dominant time constant may be defined in several ways: based on the largest slope, on the time when the output reaches 63% or 95% of its final value, etc... The theoretical values of θ reported in the table and in the next curves use the time when the output reaches 63% of its final value.

TABLE 1: Estimates obtained with the first process.

L	θ	$\hat{\theta}$	K_u	$\widehat{K_u}$	ω_u	$\widehat{\omega_u}$	$\widehat{K_p}$	$\hat{T}$	$\hat{L}$	$\overset{\circ}{d}$
1	0.04	0.05	36.3	35.3	1.6	1.365	1.12	25.85	1.17	1
3	0.12	0.21	12.5	3.92	0.55	0.185	1.075	23.05	4.77	>1
5	0.2	0.19	7.72	8.29	0.338	0.343	1.08	26.0	4.86	1
7	0.28	0.25	5.70	6.66	0.25	0.249	1.053	27.95	6.89	1
10	0.4	0.395	4.17	4.24	0.179	0.176	1.093	25.75	10.15	1
15	0.6	0.62	2.99	2.915	0.125	0.1225	1.093	24.7	15.35	1
20	0.8	0.78	2.40	2.44	0.098	0.0985	1.10	25.25	19.75	1
25	1	0.94	2.06	2.15	0.0812	0.081	1.10	26.4	24.85	1
30	1.2	1.12	1.83	1.90	0.0697	0.071	1.10	25.91	29.1	1
40	1.6	1.15	1.545	1.88	0.055	0.0595	1.10	30.3	34.8	1
50	2	1.46	1.38	1.63	0.0458	0.0495	1.10	29.87	43.70	1

TABLE 2: Estimates obtained with the second process.

L	θ	$\hat{\theta}$	K_u	$\widehat{K_u}$	ω_u	$\widehat{\omega_u}$	$\widehat{K_p}$	$\hat{T}$	$\hat{L}$	$\overset{\circ}{d}$
1	0.2	0.35	47.84	4.94	0.314	0.089	0.844	36.1	12.7	>1
3	0.25	0.41	16.5	4.09	0.18	0.082	0.844	32.7	13.5	>1
5	0.3	0.35	10.2	3.7	0.139	0.076	0.84	35.2	12.4	>1
7	0.36	0.55	7.52	3.32	0.116	0.071	0.84	30.9	17.1	>1
10	0.44	0.38	5.51	5.72	0.096	0.122	0.838	38.4	14.6	1
15	0.58	0.7	3.96	2.33	0.077	0.054	0.848	33.7	23.6	>1
20	0.72	1.1	3.185	2.09	0.065	0.047	0.844	30.3	33.25	>1
25	0.86	0.99	2.725	1.89	0.057	0.042	0.848	34.1	33.7	>1
30	1.0	1.2	2.42	1.76	0.051	0.039	0.843	32.2	29.5	>1
40	1.28	1.03	2.05	2.6	0.043	0.05	0.844	39.4	40.9	1
50	1.56	1.42	1.83	1.55	0.037	0.03	0.84	38.9	55.4	>1

Comparison With Two Other Controllers

Two controllers are designed at the end of the tuning procedure. The first regulator is based on a pure application of Ziegler-Nichols tuning formula. The theoretical values of the process ultimate gain and ultimate period are given by the operator. These values can be computed or estimated since the true model of the process is known. A PI structure is selected if the measurement noise is greater than 0.01 or if the estimate normalised process dead-time is greater than one. If this is not the case, a PID controller is selected.

The second regulator is based on the rules used in the Satt control auto-tuner ECA 40 or ECA 400 as it is exposed in [*Hägglund & Åström 1989*]. If, based on the relay experiment, the process is classified as a first order system without dead-time or if the measurement noise is larger than 0.05 and the normalised process dead-time less than one, a PI controller based on the relations Kr = 0.5*Kue and Ti = 4*Tue/(2π) is used. If the normalised process dead-time is bigger than one, a PI controller satisfying the relations Kr = 0.25*Kue and Ti = 1.6*Tue/(2π) is used. Otherwise, the Satt controller has a PID structure. Its parameters are such that, by introducing this regulator in the control loop, the point $G(i\omega_{ue})$ estimated on the process Nyquist curve is moved to the point

$$G(i\omega_{ue})*G_{PID}(i\omega_{ue}) = 0.5*e^{-i\frac{3\pi}{4}}$$

This design method can be viewed as a combination of phase and amplitude margin specification. The value of the ratio of Ti over Td is fixed to 6.25. The factor ß is set to zero.

The behaviour of the closed loop system is compared during four experiments : a set point change, a set of false measures, a linear variation of the process reference and a load disturbance. Figure 4.2 displays a typical test-phase performed with the three different controllers. The four experiments have the same length. The set of false measures is equal for the three tested controllers. A false measure has a probability equal to 5% to occur ; the disturbance added to the output is uniformly distributed in the interval [-0.2 ; 0.2]. During these experiments, the sums of the absolute values of the process error are computed.

The three controllers performances tested with three different processes are displayed in the curves in figure 4.3 to 4.5. The theoretical normalised dead time is reported on the x-axis and a weighted sum of the absolute value of the control error on the

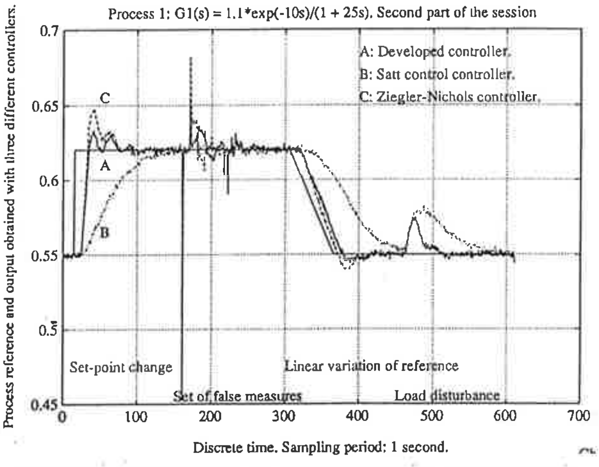

Figure 4.2 Process output and reference obtained during the test phase of the developed controllers.

0/6

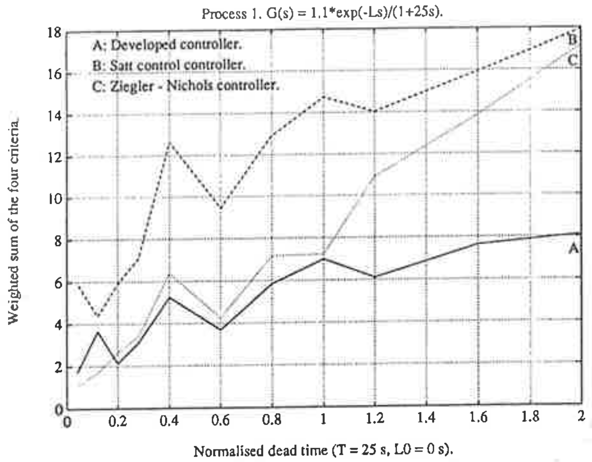

Figure 4.3: Test of the three controllers with the process 1.

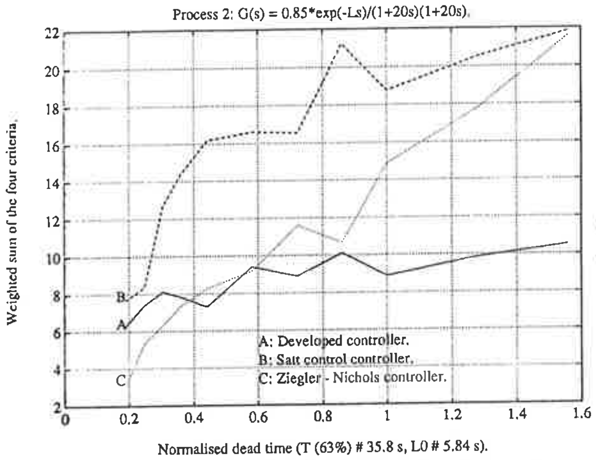

Figure 4.4: Test of the three controllers with the process 2.

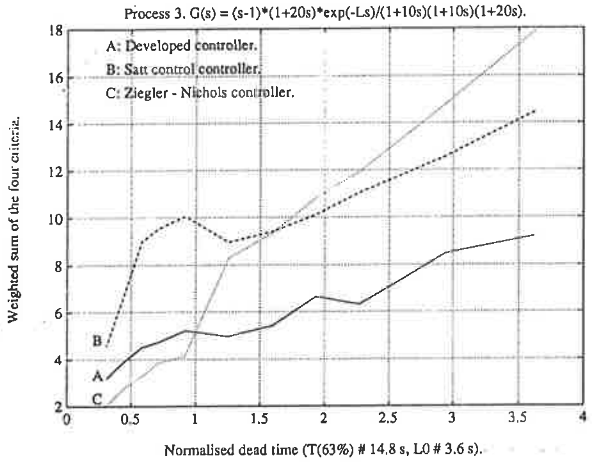

Figure 4.5: Test of the three controllers with the process 3.

y-axis: $y(\theta) = (1+\mu) * \sum_{\text{exp. 1}} |e_k| + \sum_{\text{exp. 2}} |e_k| + \sum_{\text{exp. 3}} |e_k| + \sum_{\text{exp. 4}} |e_k|$

where μ is the overshoot obtained during the set point change experiment. The contribution of each experiment to the final value of $y(\theta)$ is roughly 40%, 10%, 30% and 20%.

For small values of θ, it can be seen that the "ideal" Ziegler-Nichols controller is better than the developed controller. However, it can be argued that:

1. This is an ideal controller since it uses the true ultimate point.
2. In this case, the value of μ is much bigger (typically 0.6 compared to 0.2 or 0) than for the two other controllers.
3. The process input variations given to obtain a constant process output are much larger than those given by the two other controllers.

5. CONCLUSIONS

This paper has described a possible architecture of a system for expert control and its use to obtain an improved PID tuner. The expert system structure is similar to the one described in [*Årzén 1987*]. Our system is, however, built up of commercial components as the expert system shell Muse, procedures in C and Unix sockets.

Muse has some very nice features such as the blackboard facilities, the object-oriented programming, the downloading facilities. However, it is closer to a general programming environment than an expert system shell with all the advantages and disadvantages this implies. Two of these drawbacks are that it takes some time to learn the system and the built-in real-time facilities are less developed than in other similar tools.

The expert system has the following ingredients: relay oscillations analysis, open and closed loop step responses. The advantages of characterising process dynamics by three parameters suggested in [*Åström et al 1989*] have clearly been demonstrated. A novel feature introduced is the asymmetrical relay. By using such a device, it is possible to obtain three parameters from one single experiment. This is worth further works.

REFERENCES

Årzén, K.E. (1987): "Realization of Expert System Based Feedback Control", PhD dissertation, Coden: LUFTD2/TFRT-1029, Dpt of Automatic Control, Lund Inst. of Technology.

Åström, K.J., Anton, J.J., Årzén, K.E. (1986): "Expert Control", Automatica, 22, nº 3, pp. 277-286.

Åström, K.J., & Hägglund, T. (1988): "A new autotuning method", IFAC Workshop on Adpative Control of Chemical processes, 17-19 August, Copenhagen, Denmark

Åström, K.J., & Hägglund, T. (1988): "Automatic Tuning of PID Controllers", ISA, Research Triangle Park, NC, USA.

Åström, K.J., Hang, C.C., Persson, P. (1989): "Towards Intelligent PID Control", IFAC Workshop on Artificial Intelligence in Real-Time Control, 19-21 Sept., Shenyang, PRC.

Bristol (1977): "Pattern Recognition: An alternative to Parameter Identification in Adaptive Control", Automatica 13., pp. 197 - 202.

Hägglund, T. & Åström, K.J. (1989): "An Industrial Adaptive PID Controller", Int. Symposium on Adaptive Systems in Control and Signal Processing, 19-21 April, Glasgow, UK.

Hang, C.C. and Åström, K.J. (1988): "Pratical aspects of PID auto-tuners based on relay feedback", IFAC Workshop on Adpative Control of Chemical processes, 17-19 Aug., Copenhagen.

Hang, C.C. and Åström, K.J. (1988): "Refinements of the Ziegler-Nichols Tuning Formula for PID Auto-tuners", Proc. ISA Annual Conference, Houston, USA, pp. 1021-1030.

Sallé, S.E. and Årzén, K.E. (1989): "A Comparison Between Three Development Tools for Real-Time Expert-Systems: Chronos, G2 and Muse", Proc. of CACSD, Tampa, Florida-USA.

Sallé, S.E. (1990): "Sur l'utilisation des systèmes experts dans l'automatique", PhD dissertation. Laboratoire d'Automatique de Grenoble, France.

Ziegler, J.G. & Nichols, N.B. (1942): "Optimum Settings for Automatic Controllers", Trans. of the ASME, 64, pp. 759-768.