15th Triennial World Congress, Barcelona, Spain

DETECTING PROCESS VARIATIONS IN LOW-END PID AUTOTUNERS

Alberto Leva [1], GianAntonio Magnani

Dipartimento di Elettronica e Informazione, Politecnico di Milano
Via Ponzio, 34/5 - 20133 Milano (Italy)

Abstract: This paper presents a technique for detecting process dynamics variations with barely essential computational resources, thus applicable in very low-end autotuning regulators. The method is based on the online computation of convenient indexes, compared to nominal values obtained in the tuning phase. After describing the procedure, the results of its validation on a linear test batch, a test in a nonlinear case and some experimental results are briefly reported.

Keywords: PID control, autotuning, process control.

1. INTRODUCTION

Many statistics report that most industrial regulators are single-loop and that users want them to take only the strictly necessary decisions, so as to maintain as much control on the process as possible (Engineering, 1998). Therefore, any result aspiring to spread in the application domain must take the twofold challenge of yielding a technology meeting the users' operating desires and implementable in low-end products. The first face of the challenge requires applying theoretical results with great attention to the cultural attitude of process engineers and operators, helping them exactly in what they need. The second face calls for bridging the gap between what can be achieved on the basis of theoretical results and what can be actually implemented in low-end products, where computational resource constraints are very strict. One of the major arenæ where this challenge is being taken concerns two of the most desired regulator features, i.e. autotuning and self adaptation (Åström and Wittenmark, 1989; K.J.Åström and Ho, 1992). Looking at the users' requirements, autotuning is highly desired for easing the startup phase of a control system while self adaptation is appreciated for coping with process variations (K.J. Åström and Ho, 1993). However, there is a certain resistance against the use of *continuous* adaptation because it can conceal the symptoms of a process malfunction, adapting the controller over and over until the fault cannot be recovered anymore (Åström and Hägglund, 1995). Moreover, continuous adaptation is not very well suited to the case of sporadic but abrupt variations. As a result, most users desire a good autotuner completed with a mechanism capable of detecting at least significant process variations. When one is revealed the system should above all warn the operator, then correct the regulator parameters if he allows it to. Looking at technological constraints, however, the implementation of quite advanced autotuning methods in low-end products is difficult but nowadays feasible, see e.g. (Leva and Manenti, 2000), while that of any theoretical result on continuous process variation detection is practically almost impossible. For a deeper discussion on the fact here just sketched, see e.g. (Goodwin, 1998) and the references provided herein. This paper presents a method for detecting process variations with the information available in a model-based autotuner and barely essential computational resources, i.e. - to give a crude figure - where a few hundred bytes of

[1] Corresponding author. Voice (39) 02 2399 3410, Fax (39) 02 2399 3412, e-mail leva@elet.polimi.it

memory are a wealth. The aim is not to compete with complex methods, which do not fit in low-end products, rather to find an alternative that is less performing but extremely simple. In fact, if at least the most significant variations are caught, the tuning improvement can be remarkable. The research has been made in two phases. In the first one the only goal was to detect significant process variations so as to invoke the autotuning procedure. In the second, it was also required that the method provided some cues for retuning the regulator. The organisation of the paper reflects this path, in that the second goal (which is at present still being studied) is briefly treated in the section devoted to further developments.

2. THE PROPOSED TECHNIQUE

2.1 *An overview*

The basic idea of the method can be summarised as follows. After every tuning operation, it is assumed that the regulator can control the process well enough. As such, a simulated copy of the loop is created and ran in parallel with the physical loop. If the behaviour of the two diverges, a "warning" phase is entered until either the divergence ends or a timeout is reached. During the warning, some adimensional indexes are computed that characterise the transients of the signals of interest. By comparing these indexes to convenient nominal values (computed immediately after the tuning) a decision is taken whether the divergence was due to external causes (e.g. a disturbance) or to a modification in the process. If this is the case, or if the warning ended by timeout, the operator is alerted that retuning is needed. The rest of this section is devoted to giving these ideas a quantitative meaning. The autotuner involved is based on the identification of a FOPDT (first-order plus dead time) process model and on the "kappa-tau" (or KT) formulæ (Åström and Hägglund, 1995). As such, it provides the PID parameters (K, T_i, T_d and the set point weight b in the proportional action) and the model parameters (gain μ, time constant T and delay L). This permits to implement the simulated copy of the loop "as the autotuner has seen it in the tuning phase", i.e. composed of the tuned PID and of the FOPDT model. Simulating this loop is the most resource-intensive part of the procedure. However, choosing the sampling interval as a fraction of the measured settling time (Leva and Manenti, 2000) can solve the problem. With reference to figure 1, after the tuning two steady-state values Δy and Δu are computed for the signals δy and δu (Δy is assumed zero if the regulator has integral action, while this is not possible for Δu since it is just a matter of realism that the model represents the

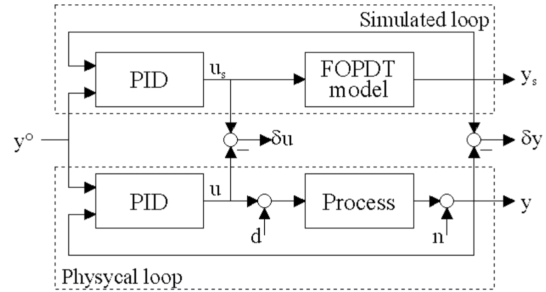

Fig. 1. Physical and simulated loop.

process up to a limited extent). As long as the model is a reasonable copy of the process and disturbances are small $\delta y - \Delta y$ and $\delta u - \Delta u$ should remain close to zero. If they deviate more than a threshold computed on the basis of the measured noise band, the system enters the warning state and monitors the signals y, u, δy and δu. Note, incidentally, that any abrupt set point change like a step starts a warning phase, but this fact is clearly known. The warning timeout is taken as $6T_i$. If by this time $\delta y - \Delta y$ has not returned within the threshold, either the process has changed or a disturbance has arrived that the regulator cannot recover. In any case, the operator is alerted. In the opposite case, the warning phase ends. This means that the regulator has recovered the loop, but the warning might have been caused by a set point variation or a disturbance, which do not necessarily require retuning, or by a process variation, which does. To distinguish, convenient indexes and decision thresholds are employed.

2.2 *The indexes*

From now on we assume that the warning ends, and not by timeout. If no process variation has occurred, y and u can either return to the original value (we omit details on the thresholding mechanism detecting this) or reach a different steady state. Briefly, to qualify the expected transients, we distinguish the two cases of figure 2: if a set point abrupt change has been applied y and u are expected to have the behaviour (a), while if no set point change has been made (thus the doubt is just between a process variation and a disturbance), since the disturbance affects only the physical loop, δy is expected to behave like (b) and δu like (a). Knowing whether the set point has changed or not, the system knows which indexes to compute on which signal. Note that the quantities to be measured -see figure 2 - do not require signal storing. For the transients of type (a) the indexes employed are

$$I_{1a} = \frac{T_1}{T},\ I_{2a} = \frac{p}{f},\ I_{3a} = \frac{T}{T_2} \qquad (1)$$

while for those of type (b) they are

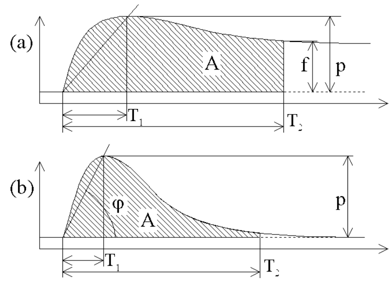

Fig. 2. Quantities used for computing the indexes.

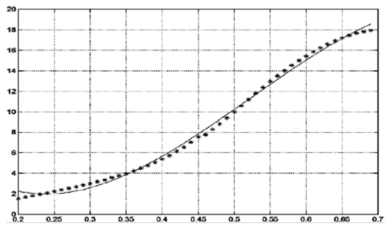

Fig. 3. Interpolating I_{1a} on y versus τ.

$$I_{1b} = \frac{A}{pT},\ I_{2b} = \frac{T}{T_2},\ I_{3b} = \frac{T\tan\varphi}{p}, \qquad (2)$$
$$I_{4b} = \frac{T_1}{T},\ I_{5b} = \frac{A}{pT_2}$$

T being the time constant of the FOPDT process model used for the tuning. In synthesis, for warnings produced by a set point variation six indexes are computed (I_{1a-3a} on y and on u), while for warnings occurring without set point variations eight are used (I_{1b-5b} on δy and I_{1a-3a} on δu). Despite their simplicity, these indexes have proven to be *extremely* insensitive to the gain and time scale of the process, to the amplitude of set point variations or load disturbances and to reasonable noises. More precisely it has been observed that, taking virtually any FOPDT model and tuning for it a PID with the KT method, these indexes either do not vary significantly or depend almost exclusively on the normalised delay $\tau = L/(L+T)$. This suggests them as good indicators of variations in the most important quantity employed by the tuning method, i.e. of the necessity to retune. Coming back to the technique rationale, to decide if the process has changed it is necessary to have a value to compare every index with, i.e. the value that each index would take if computed immediately after the tuning. These values (we shall call them "nominal") can be easily obtained by means of convenient functions of τ interpolating the nominal values of each index. Results obtained in this way have been definitely good: for example, figure 3 shows how index I_{1a} computed on y for a set point variation depends on τ. In this particular case a quite good interpolating function is

$$\bar{I}_{1a}(\tau) = 14.71 - 118.36\tau + 321.81\tau^2 - 206.03\tau^3,$$

and similar curves can be used for interpolating all the indexes. Hence, after the tuning phase, a nominal value $\bar{I}_{jk}$ can be computed as a function of the model normalised delay for each index. The so obtained normalised indexes $i_{jk} = I_{jk}/\bar{I}_{jk}$ can then be expected to be close to one if the process remains similar to that described by the model used for the tuning and to diverge from one if it changes. This assumption has been confirmed by a number of tests with different processes, which we omit for brevity. The normalised indexes are the quantities used for detecting a change at the end of a warning phase.

2.3 *The decision mechanism*

It can be stated that the normalised indexes contain the information required for detecting a process change in the cases of interest. This is very important because it separates the problem of gathering information from that of using it to decide whether a retune operation is required. The former is solved by the proposed procedure satisfactorily, as witnessed by a number of tests. The latter can be tackled with several and more or less complex methods. At present, a very simple (and preliminary) solution is employed. Each normalised index i_{jk} is compared to two bounds $i_{jk}^{min} < 1$ and $i_{jk}^{max} > 1$, then a voting mechanism is used. A process change is detected if at least two indexes are outside the bounds, or if one exceeds one bound by at least the difference of its two bounds. Of course, more indexes outside the bounds denote a "larger" variation. Values for i_{jk}^{min} and i_{jk}^{max} have been determined experimentally, and in so doing it is worth noting that there is anyway a human contribution in deciding which process variations are relevant. This would need a deeper discussion, but space does not permit it. Anyway, it has been found that $i_{jk}^{min} = 0.5$ and $i_{jk}^{max} = 2$ are suitable values for all indexes except i_{2a}, which is connected to the relative overshoot: for it, 0.9 and 1.4 have been found to give good results. Of course this part of the procedure needs further research and more systematic approaches can be taken, e.g. training a neural network with the normalised indexes as input and the user's decision whether the process has changed or not as training signal. This would permit to tailor the decision mechanism to specific needs and, together with other solutions such as the use of fuzzy logic, is at present being studied. Nevertheless, the preliminary solution adopted produces fairly good results and is extremely simple, thus advisable in low-end products. The overall procedure is shown in the flow chart of figure 4, where the confirmation requests to the operator have been omitted for simplicity.

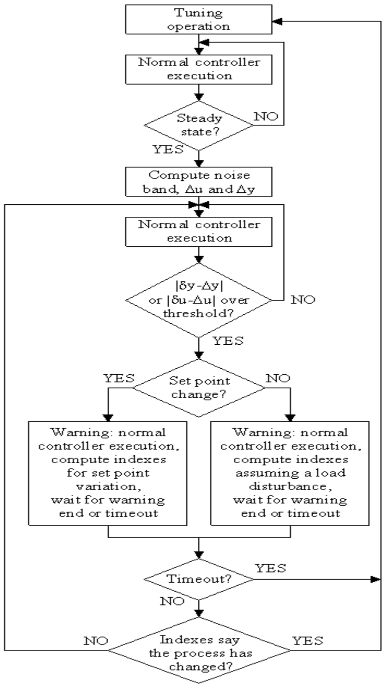

Fig. 4. Flow chart of the procedure.

3. VALIDATION IN LINEAR CASES

Test 1. The process considered is

$$P_1(s) = \frac{e^{-s}}{(1 + sT_p)^2}$$

with $1 \leq T_p \leq 5$. The PID has been tuned with $T_p = 1$. The results of the presented procedure caused by a set point change are depicted in table 1. In all the tables, indexes that never exceed the bounds are not reported for compactness, values out of the bounds are in bold and values near the bounds in italic. Note that in the various situations *all* the indexes participate in the decision.

	Indexes on y		Indexes on u	
T_p	i_{2a}	i_{3a}	i_{2a}	i_{3a}
1	1.00	0.72	0.99	1.26
1.5	1.06	**0.42**	1.08	0.71
2	1.15	**0.30**	1.22	**0.40**
2.5	1.23	**0.18**	*1.35*	**0.30**
3	1.30	**0.14**	**1.48**	**0.23**
3.5	*1.37*	**0.11**	**1.60**	**0.19**
4	**1.42**	**0.09**	**1.72**	**0.13**
4.5	**1.48**	**0.08**	**1.83**	**0.11**
5	**1.53**	**0.06**	**1.93**	**0.09**

Table 1. Set point change with P_1.

Test 2. The process considered is

$$P_2(s) = \frac{1}{(1 + 20s)^2(1 + sT_p)}$$

with $20 \leq T_p \leq 300$. The PID has been tuned with $T_p = 20$. Results caused by a load disturbance are given in table 2.

Test 3. The process considered is

$$P_3(s) = \frac{\mu_p}{(1 + 20s)(1 + 14s)(1 + 10s)(1 + 7s)}$$

	Indexes on δy				Indexes on δu		
T_p	i_{1b}	i_{2b}	i_{3b}	i_{4b}	i_{1a}	i_{2a}	i_{3a}
20	1.07	0.94	0.62	1.84	0.96	1.01	1.20
60	1.66	**0.44**	**0.44**	**2.66**	1.11	1.18	*0.61*
100	**2.31**	**0.29**	**0.36**	**3.25**	1.35	1.29	**0.38**
140	**2.95**	**0.25**	**0.31**	**3.73**	1.56	*1.37*	**0.32**
180	**3.57**	**0.22**	**0.28**	**4.15**	1.75	*1.39*	**0.23**
220	**4.20**	**0.17**	**0.25**	**4.53**	*1.92*	**1.42**	**0.21**
260	**4.79**	**0.16**	**0.24**	**4.88**	**2.07**	**1.44**	**0.19**
300	**5.38**	**0.15**	**0.22**	**5.21**	**2.22**	**1.46**	**0.18**

Table 2. Load disturbance with P_2.

with $1 \leq \mu \leq 10$; the PID has been tuned with $\mu = 1$. Results with a load disturbance are in table 3.

	Indexes on δy		Indexes on δu		
μ_p	i_{2b}	i_{5b}	i_{1a}	i_{2a}	i_{3a}
1	0.70	0.70	0.98	1.00	1.07
2	0.70	*0.50*	**0.41**	1.11	1.16
3	0.66	**0.41**	**0.30**	1.22	0.97
4	0.59	**0.34**	**0.25**	*1.33*	0.93
5	*0.54*	**0.30**	**0.22**	**1.42**	0.79
6	**0.44**	**0.25**	**0.20**	**1.50**	0.63
7	**0.37**	**0.22**	**0.18**	**1.57**	*0.53*
8	**0.31**	**0.20**	**0.17**	**1.64**	**0.42**
9	**0.25**	**0.18**	**0.16**	**1.70**	**0.33**
10	**0.18**	**0.16**	**0.15**	**1.76**	**0.25**

Table 3. Load disturbance with P_3.

4. TESTING ON A NONLINEAR PROCESS

In this section we present the (simulated) application of the proposed technique to the control of a heat exchanger with impressed thermal flux. The controlled variable y is the outlet temperature and the control variable u is the fluid flow. Omitting model details for brevity, the process is apparently nonlinear because the transport delay and the thermal resistances depend on the fluid flow rate. The disturbances considered are variations in the thermal flux Q. Two examples are reported. In the first one a set point step is applied starting from the operating point where the regulator has been tuned. Due to the flow modifications impressed by the regulator the process nonlinearity comes into effect, so that when a significantly different operating condition is reached the process dynamics changes. The method detects this, as illustrated in figure 5 and table 4. In the second example, the process undergoes a heat flux reduction. Here too the regulator modifies the flow rate to keep the output temperature at the desired value $y°$, thus changing the system dynamics. Also these variations are caught by the method, as shown in figure 6 and table 5. It can be seen that in both cases the biggest perturbation is definitely caught as a process variation. In fact transients show that the regulator can still control the process but a retuning is advisable. The second case is either caught as a "moderate" variation or not caught at all. This denotes that the decision logic could be refined, but on the other side it can be seen that deciding to retune or not is really a question. As such, this (and many other) cases lead to conclude

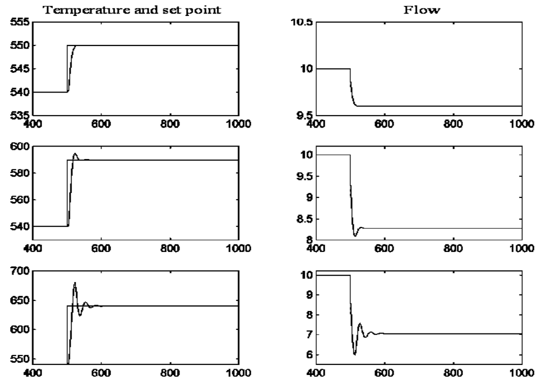

Fig. 5. Heat exchanger: set point variation.

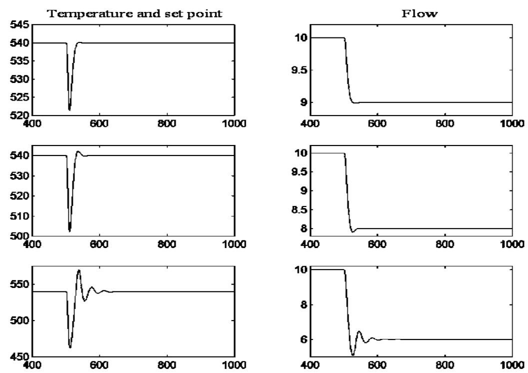

Fig. 6. Heat exchanger: heat flux variation.

that the proposed method can actually be helpful in practice.

	Indexes on y			Indexes on u		
$\delta y°$	i_{1a}	i_{2a}	i_{3a}	i_{1a}	i_{2a}	i_{3a}
10C	1.55	1.00	0.64	*0.59*	1.00	1.68
50C	*0.59*	1.10	*0.50*	**0.31**	1.10	0.89
100C	*0.58*	**1.40**	**0.25**	**0.29**	*1.36*	**0.40**

Table 4. Heat exchanger: set point variation.

	Indexes on δy			Index on δu
δQ	i_{2b}	i_{4b}	i_{5b}	i_{3a}
-10%	0.76	1.84	0.77	1.24
-20%	0.61	*1.91*	0.66	*0.59*
-40%	**0.26**	**2.12**	**0.46**	**0.27**

Table 5. Heat exchanger: heat flux variation.

5. EXPERIMENTAL RESULTS AND FURTHER DEVELOPMENTS

The method as presented here has one major limitation. It only signals that a retune is advisable, and for doing it relies on the KT autotuner it is coupled to. This can be accepted, because if the warning has recovered the system has reached a steady state, thus it can undergo a tuning operation (which causes a moderate upset). Conversely, if the tuning has ended by timeout, the system may have gone into an oscillatory or even unstable situation. In this case it may be ready for another automatic tuning or not, but in real-life cases this can only be judged by the operator. Anyway, though helpful also *as is*, the method should be given the ability of providing at least a "first-guess" correction of the regulator parameters if a process change is detected. This is being studied, and we now present briefly the (preliminary) results reached so far. First, as can be guessed from figure 3, some of the interpolating curves of the normalised indexes can provide a reasonably good estimate of τ after the change. This is still being investigated but has already been extensively verified, so that the details of the estimation will be further improved but its reliability can be already taken as a fact. Then, as indicated by the tests performed, different process changes (e.g. a gain rather than a delay or time constant(s) modification) tend to affect some normalised indexes more than others. Unfortunately, though this fact is qualitatively apparent, extracting quantitative information is not as easy and reliable as obtaining the estimate of the new τ. However, it is always possible to obtain a *very rough* estimate of gain variations from the (steady state) values of δy and δu after a recovered warning, and one of the loop time scale from the duration of the warning itself. Though this matter is still being studied, it is the authors' opinion that, even if the estimate of the new gain and settling time are extremely rough, provided that of τ is good the KT method can achieve a satisfactory (albeit approximate) correction of the PID parameters. As a brief example, consider the case referring to the first and last row of table 1. The model used for the tuning (when $T_p = 1$) has $\mu = 1$, $T = 1.55$ and $L = 1.5$, thus $\tau = 0.49$. When $T_p = 5$ a change is detected and the new τ as interpolated by the indexes is 0.4. Moreover, no significant gain changes are noticed and the duration of the warning is approximately 3.5 times the nominal one, so that the process after the change can be approximately described with $\mu = 1$, $T = 5.9$ and $L = 3.93$. Figure 7 reports the set point and load disturbance responses of the original loop ($T_p = 1$ and the regulator tuned for it) and those of the loops with $T_p = 5$ and no retune, $T_p = 5$ and the approximate PID retune provided by the KT method with the estimates above, $T_p = 5$ and a PID retuned completely (i.e. by re-identifying the FOPDT model). Clearly the complete retune produces better results, as shown e.g. in figure 8 where (from the same situation) the process delay has been modified from 1 to 10. In any case, the performance of the proposed method as an "approximated retuner" are quite encouraging, so this research path will be exploited.
To further back up this statement, we also report some experimental results. In the laboratory setup considered two electric heaters and a cooling fan act on a metal board, whose temperature is

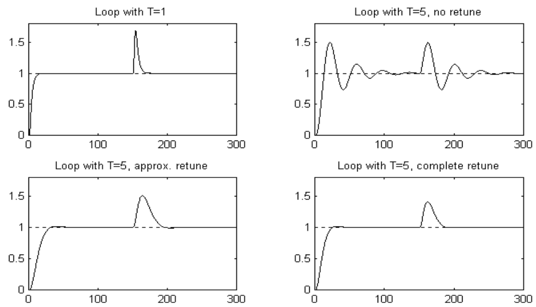

Fig. 7. Retuning example 1.

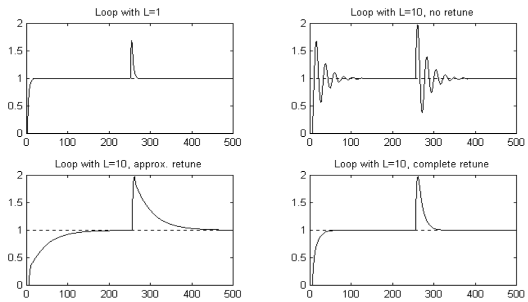

Fig. 8. Retuning example 2.

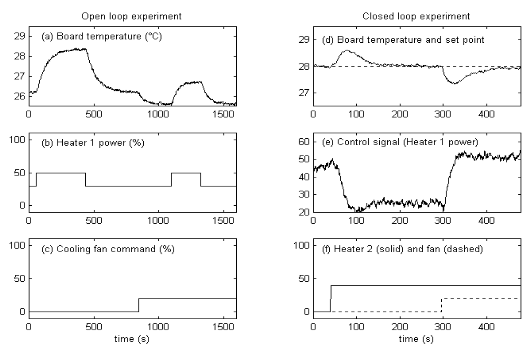

Fig. 9. Experiment results (open loop test and process variations).

the controlled variable. The control signal is the power of heater 1. Clearly heater 2 provides a load disturbance while the fan actually modifies the process dynamics reducing both gain and settling time, as witnessed by the open loop tests of figure 9(a-c). After tuning a PID with no disturbances acting, two warnings have been generated by a 40% step of heater 2 power and a 20% step of the fan command, see figure 9(d-f). The method recognises the first warning as due to a load disturbance, while the second is taken as a variation. Indexes and transients' analysis claim for a gain reduction of about 30% a time scale reduction of about 35% and, above all, a normalised delay reduction of 20% approximately. These are very rough estimation, but retuning the PID on their basis produces the results of figure 9. Apparently the new tuning is not stunning (the control is a bit oscillatory and probably too sensitive to noise), but the improvement with respect to the original one is evident.

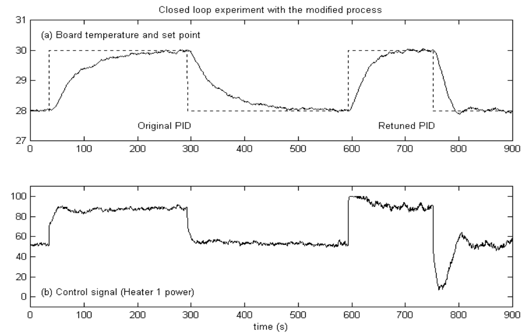

Fig. 10. Experiment results (retuning).

6. CONCLUSIONS

We have presented a method for detecting process variations (and possibly perform an approximate retune) suitable for extremely low-end autotuners. The goal was not to compete with more powerful methods, rather to make a combined use of theoretical reasoning and heuristics to obtain a tool that can be used where computational resources are minimal. The preliminary results obtained so far are encouraging. The change detection mechanism needs just some refinements in the decision phase, while the retuning phase needs further research. Nevertheless, the idea of completing a given autotuner with extremely simple indicators of the expected responses and to relate these indicators to the most relevant characteristics of the process model appears to be very promising.

7. REFERENCES

Åström, K.J. and B. Wittenmark (1989). *Adaptive control*. Addison-Wesley. White Plains, NY.

Åström, K.J. and T. Hägglund (1995). *PID controllers: theory, design and tuning - second edition*. Instrument Society of America. Research Triangle Park, NY.

Engineering, Control (1998). Single-loop controllers dominate marketplace.

Goodwin, G.C. (1998). Industrial control: setpoints for 2010. In: *Proc. UKACC '98*.

K.J. Åström, C.C. Hang, P. Persson and W.K. Ho (1992). Towards intelligent PID control. *Automatica* **28**(1), 1–9.

K.J. Åström, T. Hägglund, C.C. Hang and W.K. Ho (1993). Automatic tuning and adaptation for PID controllers - a survey. *Control Eng. Practice* **1**(4), 699–714.

Leva, A. and R. Manenti (2000). Employing a microcontroller architecture for the implementation of an advanced PID autotuner. In: *Proc. AARTC 2000*. pp. 173–178. Palma de Mallorca.