UKACC International Conference on Control 2012
Cardiff, UK, 3-5 September 2012

An adaptive PI controller for room temperature control with level-crossing sampling

Burkhard Hensel, Volodymyr Vasyutynskyy, Joern Ploennigs, Klaus Kabitzsch
Dresden University of Technology
Chair for Technical Information Systems, 01062 Dresden, Germany
Email: {burkhard.hensel, volodymyr.vasyutynskyy, joern.ploennigs, klaus.kabitzsch}@tu-dresden.de

***Abstract*—Event-based sampling allows saving energy in the sensor transmitter by avoiding unnecessary messages. One important application is room temperature control with wireless sensors. Optimizing the controller parameters of a PI controller for this application is a difficult task, because usually no process model is available and challenging issues like actuator saturation have to be taken into account. Adaptive controllers offer the possibility to tune themselves automatically. In this paper, an adaptive PI controller based on pattern recognition is proposed, designed for room temperature control, sensor energy efficiency, and level-crossing sampling. The implementation is much easier than that of most other adaptive controllers and robustness to disturbances and noise is high. The focus of this paper lies rather on the basic idea, simulations and practical issues than on theoretical investigations.**

I. Introduction

Room temperature control loops using wireless sensor networks allow high quality control with lower costs than using wired sensors, especially if the building automation system is not installed in the construction phase of the building. Reduction of the energy consumption of the nodes is one of the most investigated research issues in that field. Much energy can be saved by reducing the message count because sending messages requires much more energy than computing [1]. It has been shown that level-crossing sampling (also called send-on-delta, deadband, or Lebesgue sampling) allows a reduction of messages compared to periodic sampling while assuring the same control quality [2]. Therefore, level-crossing sampling can be used in common commercial building automation system technologies, e. g. LonWorks and EnOcean. The main idea of level-crossing sampling is that a new message from the sensor to the controller is only sent if the controlled (measured) signal has changed from the last sent value at least by a threshold Δ_{lc}:

$$|y_m(t_n) - y_m(t_{n-1})| \geq \Delta_{lc} \quad (1)$$

where $y_m(t)$ is the measured signal, and t_n and t_{n-1} are two subsequent time instances at which a message is sent. Usually the sensor wakes up periodically, measures then the current value of the controlled variable (here: room temperature) and decides according to (1) whether a message has to be sent [2].

Adaptive control allows near-optimal control without manual process identification because the controller sets its parameters itself based on available information from past control actions. Additionally, adaptive controllers change their parameters automatically if the process changes. This allows to install an untuned controller in each room of a building, and after start-up each controller optimizes itself according to the room it has to control. This allows cost reductions compared to manual tuning and higher control performance than using always-working, safe, but conservative settings.

Possible reasons for differences between the rooms are the varying room size, wall material, window area, leaking doors or windows, heating and cooling equipment, duct architecture, sensor/actuator location and sensor inertia. Reasons for process changes are variations of the flow temperature from the central heat generator, larger changes of furniture (energy storages), refurbishments (e. g. new windows, new façade insulation), sensor/actuator replacement, variations of the air flow, and changes in the HVAC (heating, ventilation and air conditioning) system.

The main contribution of this paper is an adaptive control algorithm for usage with level-crossing sampling and special emphasis on typical problems of room temperature control, i. e. actuator saturation, typical disturbances, and processes of unknown order. The goal of the adaptation is good control performance in combination with energy efficiency of the sensor. The focus of this paper is the basic idea, practical problems and simulation results; a more theoretical investigation is currently done by the authors.

This article is structured as follows. Section II gives an overview on possibilities for adaptive control based on nonuniformly sampled signals. Section III defines precisely the objective of the adaptive controller. Improvements over an older tuning rule are presented in section IV. The new adaptive controller is explained in section V. The approach is verified using simulations in section VI. While the algorithm is based on reference step responses, in section VII disturbance compensation is briefly discussed. Finally, section VIII draws the conclusions.

II. Overview: Possibilities for adaptive control with level-crossing sampling

To the authors' knowledge, there are only few works towards adaptive control with nonuniform sampling. Pawlowski et al. used a gain-scheduling controller based on outside temperature and outside wind speed together with several event-based sampling schemes for controlling the temperature in a greenhouse [3]. Dormido et al. published an autotuner

978-1-4673-1558-6/12/$31.00 ©2012 IEEE

based on a relay test with level-crossing sampling [4]. But, it cannot be used without interrupting the usual control action. Wang and Hovakimyan presented an $\mathcal{L}_1$ Adaptive Controller for an event-based, but not level-crossing sampling scheme [5].

A usual adaptive controller consists of one component identifying the process and another component for tuning the controller based on this process model [6]. In combination with level-crossing sampling the task can be split in the process identification for nonuniformly sampled signals and a separate PI controller tuning rule based on the identified model. A tuning rule for the latter point has been given by the authors [7].

In literature, there exist three basic approaches for process identification using nonuniformly sampled signals:

1) Resampling/interpolation followed by the application of methods for uniformly sampled signals.
2) Approximate Fourier transform and application of methods based on the frequency spectrum.
3) Identification of continuous-time ARMA (autoregressive moving average) models.

All these possibilities are quite complex and computationally expensive [8]–[10]. Besides, the usual adaptation methods for periodically sampled signals have strong disadvantages [6], [11], e. g. if the time delay or process order is not known in advance or if there are larger disturbances. Unfortunately, these issues are unavoidable in room temperature control. Even critical effects like *bursting* [12] or instability are possible. In addition to the methods above, there are many well known simple graphical approaches based on the open-loop step response, but these cannot be applied in closed loop what is necessary for an adaptive controller.

However, there are some approaches for adaptive control which do not need a detailed model and are thus also independent of the sampling scheme: Model free adaptive control and pattern recognition based adaptive control.

In *Model-free adaptive control* a sensitivity function is estimated from measured data and the controller output is computed using a simple learning algorithm [13]–[15] or an artificial neural network [16], [17]. Tuning the parameters of these methods (e. g. special learning parameters of the prevailing method, or security parameters for avoiding division by zero) is difficult, especially without a process model, because the optimal learning parameter settings depend on the controlled process while the final adapted parameters depend on the learning parameters. The meaning and magnitude of these parameters is not comprehensible for non-experts.

Pattern recognition based adaptive control imitates the procedure which a skilled control engineer would perform if there would be the task to improve a running but insufficiently tuned control loop without building a model of the process. The common part of these algorithms is that the first step is to extract special *features* from the measured signals. Pattern recognition based adaptive controllers usually update the controller parameters not permanently but only after significant events like set-point changes or larger disturbances. Some authors avoid therefore the name "adaptive" and use "auto-tuning" instead [18]. However, the problem with the term "auto-tuning" is that most commercially available auto-tuning controllers update their parameters not regularly but only when the user starts a procedure, mostly an open-loop experiment.

Note that the term "pattern recognition" is more known for classification and grouping of patterns [19] what is not done in any of the algorithms itemized below. However, as most authors of such control algorithms use the term "pattern recognition" [18], [20]–[24], that is also done in this paper.

The first one who made pattern based adaptive control popular was Bristol in 1977 [20]. This controller ignores the dynamics of the process and is thus relatively slow.

Seif modeled the transients by a set of elementary patterns without giving details about the adaptation rules [21].

Seem proposed a control algorithm where the changes of the PI controller parameters are described as functions of two features of the closed-loop set-point or disturbance step response [22], [23]. Some details of Seem's algorithm need periodic information of the current process output and are therefore not suitable for nonuniform sampling, but the basic idea can be transferred to level-crossing sampling. Seem's algorithm is optimized to minimize IAE (integral of absolute error) instead of considering also sensor energy efficiency. Seem spent much attention on reaching robustness for many practically important special cases of HVAC control. The algorithm has been used successfully in more than 500,000 controllers [23].

Morilla et al. published a multi-step approach for PID controllers [18]. The results are good, but unfortunately some practically important aspects like disturbances and actuator saturation have not been considered. Some of their used features of the step-responses (e. g. decay ratio and oscillation period) are practically not measurable with level-crossing sampling and sensitive to disturbances and measurement noise.

INTUNE is a commercial adaptive controller using pattern recognition for updating the parameters of a PID controller [25]. To the authors' knowledge, the details of the adaptation rules have not been published.

Segovia et al. proposed a simple iterative pattern based PID controller scheme [24] roughly based on the Ziegler/Nichols tuning rule, but they gave only a tuning rule for oscillatory step responses. Furthermore, they did not write anything about actuator saturation or disturbances.

III. Assumptions and objective

This paper deals with the control loop shown in Fig. 1 where $y_m(t)$ is the signal measured by the sensor (process output $y_p(t)$ with disturbance $d(t)$) and $y_c(t)$ the signal which is sent to the controller. $w(t)$ is the set-point or reference signal, $e_c(t) = w(t) - y_c(t)$ the control error and $u_c(t)$ the control signal (manipulated variable) which is sent to the actuator. The actuator uses a zero-order hold and its output is called $u_p(t)$. For ease of notation, a distinction between continuous-time and discrete-time signals has been avoided here.

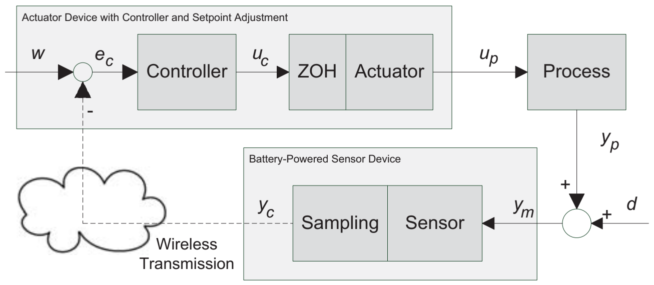


Fig. 1. Control loop with battery-powered sensor device.

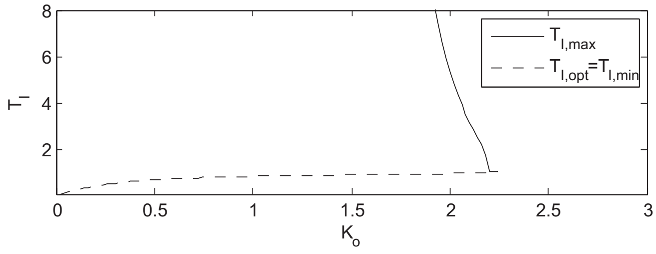


Fig. 2. Minimum (optimal) and maximum value of T_I as a function of K_o for avoiding oscillations for the example of $T_m = 1$ and $\tau = 0.2$, with continuous-time control.

In contrast to this control loop, some other authors integrate sensor and controller in one device and transmit the control signal to the actuator using level-crossing sampling [26]. The advantage of separating the sensor from the controller is that the sensor does not need a receiver or display (for editing the current set-point, weekly schedules or controller parameters) and can therefore save much energy. The actuator usually needs far more energy anyhow as well as it must power a receiver for getting the sensor messages and so the energy consumption of the controller and display is less critical.

For describing the algorithm, the process is assumed to be a FOLPD (first order lag plus time delay) process

$$G(s) = \frac{Y_p(s)}{U_p(s)} = \frac{K_m}{1 + sT_m} e^{-s\tau} \tag{2}$$

with proportional action coefficient K_m, time constant T_m, and time delay (deadtime) τ. The ratio

$$\eta := \frac{\tau}{T_m} \tag{3}$$

is called *degree of difficulty* [27]. Practical ranges of these parameters for room temperature processes can be found e. g. in [27] where one should keep in mind that variations could be larger, especially for K_m caused by variations of flow (supply) temperature. Nevertheless, the proposed algorithm is not limited to this process type because it does not need to know or identify any time constant directly. The physics of rooms (room airflow) is such complex that it theoretically cannot be modeled adequately by a transfer function (or a differential equation) [28]. The proposed controller has to deal with these issues.

PI and PID controllers are the by far mostly used controllers in industry [29], [30]. The basic continuous-time PI controller is given by

$$R(s) = \frac{U_c(s)}{E_c(s)} = K_P \left(1 + \frac{1}{T_I s}\right) \tag{4}$$

with proportional action coefficient K_P and reset time T_I.

PI(D) controllers have also been used successfully in an event-based fashion [2], [7], [31]. The PI algorithm which is used in this paper is taken from [7], only enhanced by anti-reset windup. Derivative action has not been used. This will be explained later using results of the presented investigations.

A cost function which helps to design the adaptive controller is the product

$$J_{Prod} := N_{sc} \cdot J_{ISE} \tag{5}$$

of the number N_{sc} of messages from the sensor to the controller and the control quality measured as Integral of Squared Error (ISE) J_{ISE}. The adaptation should reach small values of J_{Prod} as this is a hint for a good trade-off between message number and control performance. More detailed reasons for this choice are discussed in [7].

IV. Improved Tuning Rule

In [7] it has been shown that for minimzing J_{Prod} at step responses, large oscillations should be avoided and—in an ideal case (no actuator saturation, FOLPD process)—a suitable tuning rule is

$$K_P = \frac{0.468}{K_m \cdot \eta}, \tag{6a}$$

$$T_I = T_m. \tag{6b}$$

But, typical challenges in room temperature control are actuator saturation, large time-variable (but not step-wise) disturbances, and processes of unknown order. As stated in [7], the tuning rule (6) is not optimal in these cases. Simulations confirm that ignoring these problems leads to poorly tuned control loops. This section thus presents some improvements over tuning rule (6).

A. Oscillating step responses

As the goal is to avoid oscillating step responses, it is interesting to know for which K_P that is possible at all. If the open-loop gain K_o, which is defined as

$$K_o := K_P \cdot K_m, \tag{7}$$

is greater than a limit $K_{o,max}(\eta)$, there are oscillations, independent of the selection of the reset time T_I, see an example in Fig. 2. Fig. 3 shows the maximum $K_{o,max}$ as a function of η, found by simulations. The simulation uses continuous-time control to reach idealized results, what is similar to an infinitesimally small threshold Δ_{lc}. For numerical reasons a deadband around the set-point should be defined in which the control loop is allowed to oscillate for avoiding too conservative results. This has been set to 1 % of the step width.

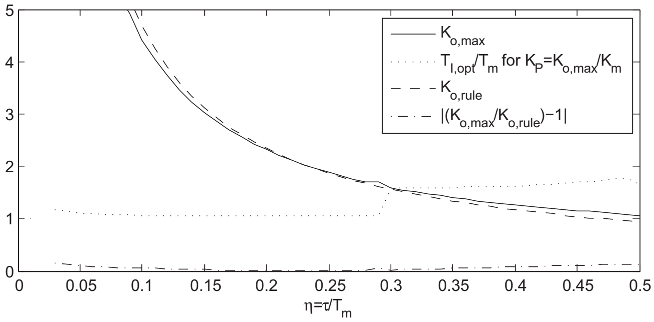


Fig. 3. $K_{o,max}$ as a function of η for a deadband of 1 % and the optimal setting of T_I, normalized to T_m. Also $K_{o,rule}$ according to tuning rule (6) is shown as well as the ratio of $K_{o,max}$ and $K_{o,rule}$.

The conclusion for avoiding oscillating step responses is

$$K_P \overset{!}{\leq} \frac{K_{o,max}(\eta)}{K_m}. \tag{8}$$

K_P according to (6a) lies near to the curve for a deadband of 1 % (see Fig. 3) and gives therefore nearly the fastest response with no more than 1 % overshoot.

B. Actuator saturation

The immediate change of the controlled variable $u_p(t)$ caused by proportional action after a stepwise reference change Δw can be calculated by

$$\Delta u_{prop} = K_P \cdot \Delta w. \tag{9}$$

Since in real applications $u_p(t)$ is bounded between 0 and 1 (valve fully closed to valve fully opened), it is useless to set K_P higher than $1/\Delta w$ because of actuator saturation. That means

$$K_P \overset{!}{\leq} \frac{1}{|\Delta w|}. \tag{10}$$

An illustrative example: Assumed that the reference change for night setback is $\Delta w = -5\,\text{K}$. So, each $K_P > 0.2$ would result in immediate actuator saturation. Of course, increasing K_P does not degrade control performance at step responses (because it has little influence if the stability limit is not exceeded), but high K_P will lead to many messages in "steady state" without improving control performance as will be discussed in section VII.

Note that the fulfillment of (10) does not guarantee that there is no actuator saturation, because that depends on the value of u_p before the set-point change as well as on integral action.

Besides, derivative action (a PID controller) increases actuator action and hence also actuator saturation. Thus, it is at step responses reasonable to do without derivative action.

C. New tuning rule

The proposed tuning rule for K_P is the combination of (8) and (10)

$$K_P = \min\left(\frac{1}{|\Delta w|}, \frac{K_{o,max}(\eta)}{K_m}\right) \tag{11}$$

As (6a) approximates $K_{o,max}(\eta)/K_m$ quite well, the rule can be approximated by

$$K_P \approx \min\left(\frac{1}{|\Delta w|}, \frac{0.468}{K_m \cdot \eta}\right). \tag{12}$$

No rule for setting T_I is given here because iterative optimization is used in the proposed adaptive controller.

V. Proposed adaptation strategy

This section presents the new adaptation strategy.

A. Used patterns

Only reference step responses are taken to analyze the process behavior. In a typical office building, reference changes occur twice a day: Once in the morning and once in the evening because of night setback. In residential buildings, there are often four reference changes because the set-point is higher only in the morning and in the evening. So, there are at least two step responses a day at which the controller parameters can be optimized.

The precondition for benefiting from an adaptive controller is that the process does not change faster than the adaptation can follow. Faster changes are regarded as belonging to the disturbances. However, the sources of process changes which have been itemized in section I do not change significantly during one day (or they change fast but only seldom, like on refurbishment). Only considering the variation of the flow temperature may be advantageous, what could be done via gain scheduling [3], [32].

Some other authors used also step-wise disturbances for updating the controller parameters [22], [24]. However, since significant step-wise load changes do not often occur in practical room temperature control, only set-point changes are considered here.

B. Proportional action coefficient K_P

As

$$K_m = G(0) = \frac{Y_p(0)}{U_p(0)}, \tag{13}$$

and assuming a constant disturbance d, K_m can be calculated from two pairs ($u_p[k]$, $y_p[k] + d$) measured in "steady state" with different values of $u_p[k]$. So, after each closed-loop step response, when the process is again in "steady state", K_m is estimated by

$$K_m = \frac{(y_p(t_s) + d) - (y_p(t_0) + d)}{u_p(t_s) - u_p(t_0)} \approx \frac{y_c(t_s) - y_c(t_0)}{u_c(t_s) - u_c(t_0)} \tag{14}$$

where t_0 is the step time (or shortly before) and t_s is the time when steady state is (assumed to be) reached after the step response. This method is relatively robust to measurement noise but in the case of slowly varying loads $d(t)$ the estimation gets inaccurate. If $(t_s - t_0)$ is chosen too small, the "steady state" may not yet have been reached, leading to more inaccurate estimation of K_m. Contrary, if $(t_s - t_0)$ is chosen too large, load changes can get more influence on the results. In the simulations of section VI $(t_s - t_0)$ has been set to one hour.

After estimating K_m, (12) is applied to compute K_P. Unfortunately, η depends on τ and T_m which are hard to estimate in closed loop under noisy conditions with time-variable load and significant threshold Δ_{lc}. Instead, an upper limit η_{max} can be used which is the greatest degree of difficulty which is expected to occur. According to [27], for room temperature control η_{max} is 0.3. If η of the real process is lower than η_{max}, K_P is lower than necessary, but the reset time T_I will be adjusted (reduced) to improve the control loop performance, see section V-C. The authors are working on more sophisticated solutions, but simulations show that this simple solution works well, too.

C. Reset time T_I

As announced in section IV-C, an iterative method is used for updating T_I, i. e. the reset time is updated after each set-point change based on the measured overshoot. The overshoot h_r is a monotonically decreasing function of T_I, see some examples in Fig. 4(a). This is intuitively clear because the higher the reset time T_I, the slower is the response and the less the set-point is exceeded before the controller can react on the overshoot. The basic idea for adaptation is:

- If there is (too much) overshoot, increase the reset time.
- If there is no (or too little) overshoot, decrease the reset time.

Because of (12) it is guaranteed that there is a setting of T_I without oscillations (in the case of an ideal FOLPD process, continuous-time control and no disturbances).

The step width for updating T_I must be defined. Simple learning algorithms (similar to first order low-passes) could be used, comparable to [13], [22], [24], but it is possible to apply more specific algorithms. An initial attempt is given in the following; more sophisticated solutions as well as theoretical analysis are part of current research of the authors.

Let $T_{I,opt}$ define the smallest possible reset time without overshoot. Fig. 4(a) shows some examples for the overshoot h_r as a function of $T_I/T_{I,opt}$ using continuous-time control. The qualitative curves of Fig. 4(b) substantiate the assumption that the overshoot can be approximated by

$$h_r \approx \begin{cases} \gamma \cdot \left(\frac{T_{I,opt}}{T_I} - 1\right), & \text{if } T_I < T_{I,opt} \\ 0, & \text{otherwise,} \end{cases} \tag{15}$$

The proportionality constant γ is discussed later.

Equation (15) can be used to calculate $T_{I,opt}$ based on the measured overshoot $h_{r,old} > 0$ and the appropriate reset time $T_{I,old}$:

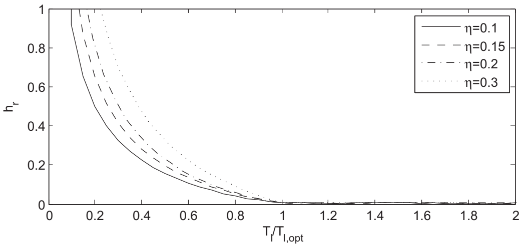
$$T_{I,new} = T_{I,opt} = \left(1 + \frac{h_{r,old}}{\gamma}\right) \cdot T_{I,old}. \tag{16}$$

If there is no overshoot, it is only known that the reset time should be reduced, but not how much. However, from (15) a maximum step width can be calculated with the requirement that the overshoot after the update must not exceed a given limit $h_{r,max}$. The important case is $T_{I,old} = T_{I,opt}$, because a reduction of T_I will then result in the highest overshoot. The rule for updating is

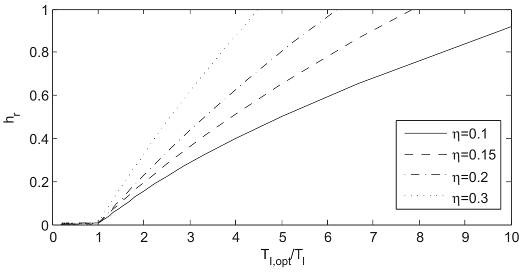
$$T_{I,new} = \frac{1}{1 + \frac{h_{r,max}}{\gamma}} \cdot T_{I,old}. \tag{17}$$

The greater $h_{r,max}$ is chosen, the faster the algorithm converges to less conservative settings. $h_{r,max}$ can be reduced after finding a rough estimation of $T_{I,opt}$, e. g. after the first step-response with overshoot happened.

The remaining task is to find the appropriate value of γ. This parameter depends on η, K_o, Δ_{lc}, the real process order, and actuator limits. Additionally, the measured overshoot $h_{r,old}$ can be adulterated with disturbances and other effects, in particular when using level-crossing sampling where the exact overshoot cannot be measured. Similarly to [22] a low-pass filter could be used for reducing influences of disturbances, but filter design is difficult if the process parameters can vary over a wide range, especially with nonuniform sampling. Fortunately, it is not very important to know an exact value



(a) h_r as a function of $T_I/T_{I,opt}$



(b) h_r as a function of $T_{I,opt}/T_I$

Fig. 4. Overshoot h_r as a function of reset time T_I, normalized to $T_{I,opt}$ (and inverse) for several η and $K_o = 1.5$.

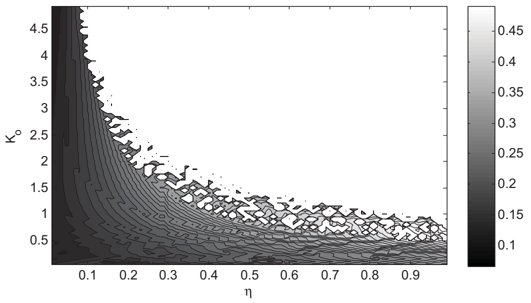
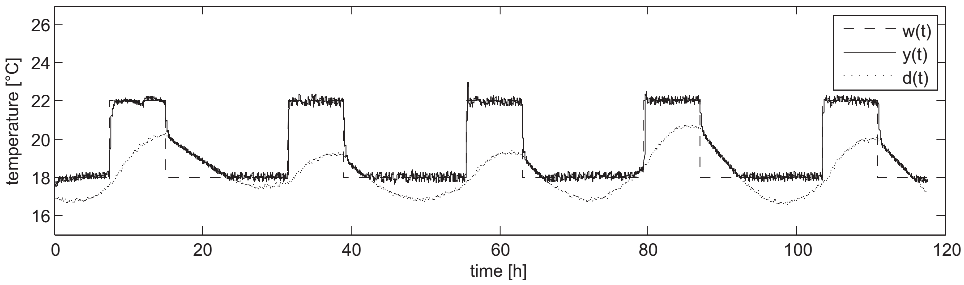
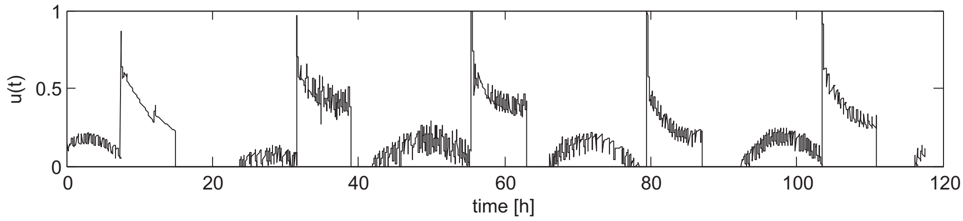


Fig. 5. γ as a function of η and K_o for idealized conditions: Continuous-time control, FOLPD process, no disturbances, no actuator saturation.



(a) $y_m(t)$, $w(t)$ and $d(t)$



(b) $u_p(t)$

Fig. 6. Simulation over five days with level-crossing sampling and measurement noise.

of γ because of the iterative method—if γ is chosen too high, only the convergence rate is degraded. Thus, a worst case estimation is enough. Fig. 5 shows γ as a function of K_o and η under the idealized assumptions of continuous-time control, a FOLPD process, no disturbances and no actuator saturation. According to this graphic and assuming that in room temperature control $\eta < 0.3$ holds [27], the worst case to be expected is $\gamma \approx 0.3$. This value has also been found to work well in simulations.

Step responses with too little overshoot *and* actuator saturation must not be used for updating T_I, because the too little overshoot may result from actuator saturation and not from sluggish tuning. This is no problem, since the controller knows whether the actuator limits (0 or 1) have been reached in the last step response.

This iterative method allows adaptation without estimating τ and T_m what would be difficult with level-crossing sampling and disturbances. It also improves the tuning results in the case of unknown process order where (6) leads to suboptimal results [7].

D. Initial settings

The initial settings of K_P and T_I should be based on the most critical process to be expected. This is the maximum of each parameter η, K_m, and τ, i. e. for room temperature control according to [27] $\eta = 0.3$, $\tau = 0.05\,\text{h}$, and $K_m = 10\,\text{K}$. K_P can then be set using (12) and (with regard to [7]) T_I to T_m, i. e. $T_I = 0.5\,\text{h}$.

In result, the first step response is stable. In most cases (all but the most critical case) the response will be too sluggish. After each step response, K_P is updated using (12) and (14) as well as T_I is reduced according to (16). The responses get faster until overshoots occur. Then T_I is increased using (17).

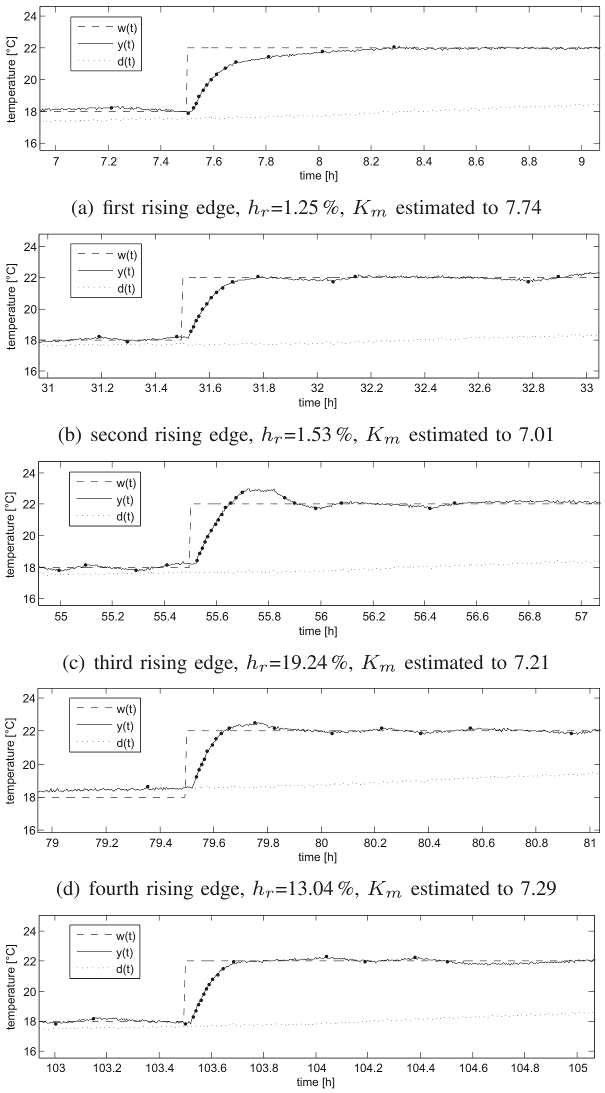
VI. Simulations

Fig. 6 presents a simulation over five days using a process with $T_m = 0.15\,\text{h}$, $K_m = 7\,\text{K}$, and $\tau = 0.01\,\text{h}$. The disturbances (this is the room temperature without heating, i. e. $u_p(t) \equiv 0$) are based on measurements taken from an office building which was built in 2005, having a heat energy consumption of 34 kWh/m^{2}a. By setting Δ_{lc} to 0.3 K (see section VII) and applying night setback of 4 K, an overshoot of 7.5 % (0.3K/4K) must be accepted because of level-crossing sampling. The internal sampling period of the sensor is 0.005 h.

As this is not the "most critical" of expectable processes, the first step response is slower than necessary, shown detailed in Fig. 7(a). Fighting against that, the adaptation algorithm estimates K_m, updates K_P and reduces T_I. The negative set-point changes cannot be used for pattern recognition, because actuator saturation without overshoot occurs (the valve is fully closed). The second rising edge is significantly faster than the first, Fig. 7(b). The third even has (too much) overshoot, Fig. 7(c). So, the adaptation algorithm increases the reset time. The fourth step response has less overshoot, Fig. 7(d), but due to the inexact adaptation method, also this step response has too much overshoot and the reset time is further increased. The fifth set-point change is as desired.

VII. Disturbance compensation

The tuning rules (6) and (12) are exclusively based on step responses. But, it is interesting how the parameters should be chosen in "steady state" when only disturbances have to be compensated. To the authors knowledge, until now only limit cycles have been studied in several publications (e. g. [2], [33]). Limit cycles are periodic movements of the controlled variable between two or more sampling levels which often occur in "steady state", especially because of integral action,



(a) first rising edge, h_r=1.25 %, K_m estimated to 7.74

(b) second rising edge, h_r=1.53 %, K_m estimated to 7.01

(c) third rising edge, h_r=19.24 %, K_m estimated to 7.21

(d) fourth rising edge, h_r=13.04 %, K_m estimated to 7.29

(e) fifth rising edge, h_r=7.12 %, K_m estimated to 9.53

Fig. 7. The five first rising edge step responses with level-crossing sampling and measurement noise. The dots are sampling instants, i. e. level crossings.

and that results in an unpleasant high message rate without improving control performance. But, if disturbances lead to level crossings (and therefore messages) anyhow, the importance of avoiding limit cycles is reduced.

The disturbance is equal the room temperature without heating or cooling, i. e. $u_p(t) \equiv 0$. The main reasons for disturbances are time-varying heat flows to or from the outside which depend on the outside temperature, sunlight, and room utilization (how many persons and machines are creating heat). These influences do not lead to any step-wise changes of the room temperature. It is well known from experience and measured in the office building considered in section VI that without heating the room temperature is a slowly varying signal, usually periodic with the minimum after midnight and the maximum after noon. The amplitude depends on the outside temperature, sunlight, and room utilization and does thus change from day to day. In this section, for theoretical investigations disturbances are assumed to be of the form

$$d(t) = A \cdot \sin\left(\frac{t}{24\,\mathrm{h}}\right), \tag{18}$$

that means a sinus curve with a period of one day and an amplitude A.

Simulations show that the PI controller parameter settings minimizing (5) depend on the ratio a of the disturbance amplitude A and the threshold Δ_{lc} which is defined as

$$a := \frac{A}{\Delta_{lc}}. \tag{19}$$

The practical bounds of a are of interest. Since humans usually do not feel temperature changes smaller than 0.3 K there is no need to set Δ_{lc} significantly smaller than this value—even if it would allow a smaller value for the cost function (5) because of smaller ISE, the occupants would not notice it while the message rate (and therefore energy consumption) would be unnecessarily high, and also the costs for a sensor measuring such exactly would be high. Besides, the temperature variations inside one room because of stratification and bordering spaces are significantly greater than 0.3 K [28]. The amplitude A can be expected to be lower than 5 K (very old buildings with poor insulation), for modern, well insulated buildings lower than 1.5 K. So, a can be expected to be lower than 17 (5 K/0.3 K), often lower than 5 (1.5 K/0.3 K).

Simulations show that for such small values of a the cost function (5) can be optimized by considerably reducing K_P compared to the value got by (6a) because reducing K_P increases the period of limit cycles and hence decreases the message number and energy consumption. As the temperature set-point trajectory correlates roughly with the disturbance (at noon higher than at midnight) the necessary control action is further reduced.

This fact can be used to improve the tuning rule in "steady state" by simply adding a factor κ, getting

$$K_P = \frac{0.468 \cdot \kappa}{K_m \cdot \eta}. \tag{20}$$

Optimal values of κ as a function of a and η found by simulations can be seen in Fig. 8. The optimal κ rises with a because for higher a a higher K_P can improve the ISE more significantly than for lower values of a.

κ could be realized in the controller by using a second degree of freedom (set-point weighting).

Note that using derivative action of a PID controller would lead to stronger actuator action and faster oscillations (more messages) without improving control performance.

Unfortunately, since a and η are neither known a priori nor estimated on-line, the formula cannot be applied. Automatically finding values of κ suitable for each controlled room is another part of current research of the authors. Additional sensors, e. g. for outdoor temperature, room occupancy, and illumination, can help estimating a. Weather forecasts promise even better results. However, these solutions are much more

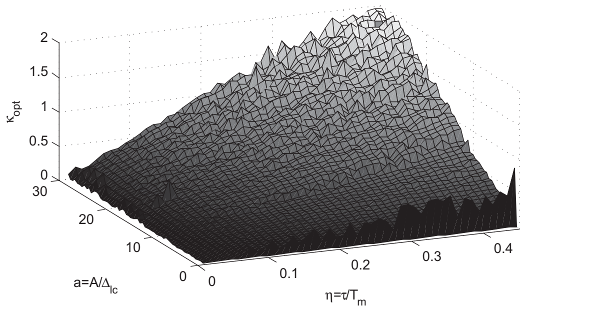


Fig. 8. Optimal setting of κ in (20) as a function of degree of difficulty η and disturbance to threshold ratio a. Simulation parameters are $T_m = 0.3\,\text{h}$, $K_m = 10$, $T_I = T_m$ and $\Delta_{lc} = 0.3\,\text{K}$. Minimum inter sample intervals [7] and actuator saturation have not been taken into account.

expensive than a simple single-loop controller like the proposed one.

VIII. Conclusion

A pattern-based adaptive controller designed for level-crossing sampling and room temperature control has been presented. Simulations show that the algorithm can deal with typical problems of this kind of control loop. Also several points for future research have been pointed out, including practical improvements, theoretical analysis and less conservative assumptions.

Acknowledgment

This work was funded by the European Commission, the European Social Funds and the Sächsische Aufbaubank in the junior research group ZESSY.

References

[1] R. Zheng, J. C. Hou, and N. Li, "Power management and power control in wireless networks," in *Ad Hoc and Sensor Networks*. Nova Science Publishers, 2004, pp. 1–25.

[2] J. Ploennigs, V. Vasyutynskyy, and K. Kabitzsch, "Comparative study of energy-efficient sampling approaches for wireless control networks," *IEEE Transactions on Industrial Informatics*, vol. 6, no. 3, pp. 416–424, August 2010.

[3] A. Pawlowski, J. L. Guzmán, F. Rodríguez, M. Berenguel, J. Sánchez, and S. Dormido, *Factory Automation*. InTech, March 2010, ch. Study of event-based sampling techniques and their influence on greenhouse climate control with Wireless Sensor Networks, pp. 289–312.

[4] S. Dormido, L. Grau, J. L. Fdez-Marron, M. A. Canto, and J. M. de la Cruz, "A PID autotuner based on the Åström and Hägglund method's using non-uniform sampling," in *Proceedings of the 6th Mediterranean Electrotechnical Conference*, vol. 2, Ljubljana, Slovenia, May 1991, pp. 860–863.

[5] X. Wang and N. Hovakimyan, "$\mathcal{L}_1$ adaptive control of event-triggered networked systems," in *Proceedings of the 2010 American Control Conference*, Marriott Waterfront, Baltimore, MD, USA, June–July 2010, pp. 2458–2463.

[6] K. J. Åström and B. Wittenmark, *Adaptive Control*. Reading: Addison-Wesley Publishing Company, 1989.

[7] B. Hensel, J. Ploennigs, V. Vasyutynskyy, and K. Kabitzsch, "A simple PI controller tuning rule for sensor energy efficiency with level-crossing sampling," in *Proceedings of the 9th International Multi-Conference on Systems, Signals & Devices*, Chemnitz, Germany, March 2012.

[8] J. Gillberg and L. Ljung, "Frequency domain identification of continuous-time ARMA models: Interpolation and non-uniform sampling," Linköpings universitet, Linköping, Sweden, Tech. Rep., September 2004.

[9] E. Müller, H. Nobach, and C. Tropea, "Model parameter estimation from non-equidistant sampled data sets at low data rates," *The measurement of scientific and technological activities*, vol. 9, pp. 435–441, 1998.

[10] S. Ahmed, B. Huang, and S. L. Shah, *Identification of continuous-time models from Sampled Data*. Springer London, March 2008, ch. Process Parameter and Delay Estimation from Non-uniformly Sampled Data, pp. 313–337.

[11] V. J. VanDoren, *Techniques for Adaptive Control*. Amsterdam: Butterworth-Heinemann, 2003, ch. Introduction, pp. 1–21.

[12] B. D. O. Anderson, "Adaptive systems, lack of persistency of excitation and bursting phenomena," *Automatica*, vol. 21, no. 3, pp. 247–258, 1985.

[13] Z.-G. Han and X. Yu, "An adaptive model free control design and its applications," in *Proceedings of the 2nd International Conference on Industrial Informatics*, Berlin, Germany, June 2004, pp. 243–248.

[14] Y. Ma, X. Chen, and X. xiaohua, "A novel model free adaptive controller with tracking differentiator," in *Proceedings of the 2009 IEEE International Conference on Mechatronics and Automation*, Changchun, China, August 2009, pp. 4191–4196.

[15] Z. Hou and S. Jin, "Data-driven model-free adaptive control for a class of MIMO nonlinear discrete-time systems," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 2173–2188, December 2011.

[16] G. S. Cheng, *Techniques for Adaptive Control*. Amsterdam: Butterworth-Heinemann, 2003, ch. Model-Free Adaptive Control with CyboCon, pp. 145–202.

[17] X. Aidong, Z. Yangbo, S. yan, and L. Mingzhe, "An improved model free adaptive control algorithm," in *Proceedings of the Fifth International Conference on Natural Computation*, Tianjian, China, August 2009, pp. 70–74.

[18] F. Morilla, A. González, and N. Duro, "Auto-tuning PID controllers in terms of relative damping," in *Proceedings of the IFAC Workshp on Digital Control*, Terrassa, Spain, April 2000.

[19] A. K. Jain, R. P. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 1, pp. 4–37, January 2000.

[20] E. H. Bristol, "Pattern recognition: An alternative to parameter identification in adaptive control," *Automatica*, vol. 13, pp. 197–202, 1977.

[21] A. A. Seif, "On the adaptive pattern recognition control," in *Proceedings of the IEEE Conference on Computer Systems and Software Engineering*, May 1992, pp. 706–709.

[22] J. E. Seem, "A new pattern recognition adaptive controller with application to HVAC systems," *Automatica*, vol. 34, no. 8, pp. 969–982, 1998.

[23] ——, "An improved pattern recognition adaptive controller," in *Proceedings of IFAC Conference on Energy Saving Control in Plants and Buildings*, Bansko, Bulgaria, October 2006, pp. 117–122.

[24] J. P. Segovia, D. Sbarbaro, and E. Ceballos, "An adaptive pattern based nonlinear PID controller," *ISE Transactions*, vol. 43, pp. 271–281, 2004.

[25] T.-L. Chia and I. Lefkowitz, *Techniques for Adaptive Control*. Amsterdam: Butterworth-Heinemann, 2003, ch. Expert-Based Adaptive Control: ControlSoft's INTUNE Adaptive and Diagnostic Software, pp. 203–232.

[26] M. Beschi, S. Dormido, J. Sánchez, and A. Visioli, "On the stability of an event-based PI controller for FOPDT processes," in *Proceedings of the IFAC Conference on Advances in PID Control*, Brescia, Italy, March 2012.

[27] Control technology. Siemens Switzerland Ltd, Building Technology Group. Technical brochure, Reference number 0-91913-en.

[28] P. Riederer, *Thermal room modelling adapted to the test of HVAC control systems*, 2002, PhD thesis.

[29] K. J. Åström and T. Hägglund, "The future of PID control," *Control Engineering Practice*, vol. 9, pp. 1163–1175, 2001.

[30] A. O'Dwyer, *Handbook of PI and PID Controller Tuning Rules*, 2nd ed. London: Imperial College Press, 2006.

[31] K. E. Årzén, "A simple event-based PID controller," in *Preprints 14th World Congress of IFAC*, January 1999.

[32] F. Rodríguez, *Modeling and hierarchical control of greenhouse crop production (in Spanish)*. University of Almería, Spain, 2002, PhD thesis.

[33] M. Beschi, A. Visioli, S. Dormido, and J. Sánchez, "On the presence of equilibrium points in PI control systems with send-on-delta sampling," in *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, Orlando, FL, USA, December 2011, pp. 7843–7848.