Journal of Thermal Analysis and Calorimetry, Vol. 59 (2000) 999–1015

A NEW SOFTWARE TO SHOW THE BEHAVIOUR OF THE PID CONTROLLER
Influence of the setting parameters of DSCs. Part I

D. Brunel[*] *and L. Elegant*

Laboratoire de Thermodynamique Expérimentale, Faculté des Sciences, BP 71, 06108 Nice Cedex 2 France

(Received May 16, 1999)

Abstract

Some DSC calorimeters such as the model Setaram DSC 111 allow the user to set the parameters of temperature programming. Default values furnished by the constructor are often used, but it is very interesting to study the thermal behaviour of the regulation of the calorimeter under different conditions and kinds of set point temperatures.

For this research we have developed in C a set of softwares in order to show the behaviour of the proportional integral derivative (PID) controller of the DSC. It can help the user to choose correct values for P, I and D parameters according to the kind of experiment conducted. The software allows studies for extra parameters such as the sampling rate of the computerized PID controller or the determination of filtering of the correction.

Keywords: DSC, PID controllers, tuning PID parameters

Introduction

In differential calorimetry a furnace is used in order to study the behaviour of a material in relation to the temperature. This behaviour is characterised by a variation of the enthalpy given by a variation of the thermodynamic state of the material. These thermal studies are conducted for determined profiles of temperatures: isotherms, scanning with positive or negative slope, the features of which can be set by the user. More complex profiles can be used in special cases: scanning with positive slope immediately followed by scanning with negative slope, increasing and decreasing temperatures by steps, etc. The desired profile of temperature is imposed on the material with a programmable furnace. Thus a temperature controller is included in a calorimetric system. At the beginning these regulators were fully made in an analog way, but now for complex systems the regulations are conducted with computational softwares which run the algorithms of regulation from a computer. No doubt that PID is

* Author for correspondence: e-mail: brunel@unice.fr

1418–2874/2000/ $ 5.00

Akadémiai Kiadó, Budapest
Kluwer Academic Publishers, Dordrecht

the best known kind of temperature controller. Basically it is an analog system fully realized with discrete components.

In a DSC calorimeter such as DSC 111 a PID controller is used. In this work we show a computer program which is able to show the behaviour of the controller on a graphic display under different kinds of constraints.

DSC 111 differential calorimeter

Setaram DSC 111 calorimeter was developed in the eighties; the models from this period are very interesting because all the electronic components of their temperature controller are available on the front panel. Modern DSC 111 are fully computerized. The respect of the temperature programmed by the user is obtained with a PID controller. DSC 111 controller is fully electronic because when DSC 111 was designed the computer performances did not allow a digital PID controller in a microcomputer.

A temperature controller can be represented in a schematic diagram which shows its functioning at a time *t*. As for other kinds of thermal controller, we have five important quatities:

• The temperature which must be applied by the furnace T_c. Generally it is called the set point temperature.

• The measured temperature in the furnace T_m.

• The error *E* between the desired temperature and the measured one. This error must be the smallest possible.

• The correction *U* of the furnace power to minimize the value of the error *E* between desired and measured temperatures.

• An eventual perturbation *W* which concerns the process studied.

Regulation and control loop [1]

Theoretically there are two ways to control a system electronically: the use of a servocontrol or a regulator. In a calorimetric system these two ways are combined. The loop of temperature control has the feature of both the servocontrol and the regulator system, because generally the quantity under control (for example the temperature in the furnace of the DSC calorimeter) must follow a desired profile of values in the presence of thermal perturbations brought about by the sample.

So we can represent on a more general block diagram the relationships between the regulator, the process to control and the thermal perturbation (Fig. 1). It is in fact a classical control loop applied in the field of thermal control in differential calorimetry.

In a DSC calorimeter the *W* perturbation can be, for example, the change of the thermodynamic state of the sample which will alter its temperature. This perturbation also represents the change of the control loop with the temperature. Usually the external perturbations are filtered by the walls of the calorimetric block, which can be, i.e. DSC 111. The ‘calorimetric block’ the process and thermal perturbation *W* are presented in Fig. 1. The block ‘Thermal controller’ represents the device which must

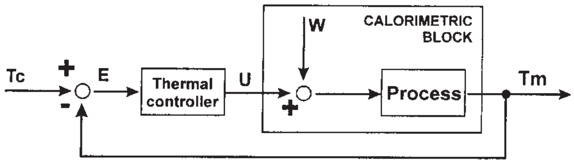

Fig. 1 Organization of the values concerned by a PID analog controller

generate a corrective signal U with the calculated error E between the measured temperature T_m and the desired T_c one. The T_m temperature can be an additional output of the loop in order to give indications to the user.

From Fig. 1 we have the two fundamental equations for the temperature control:

$$E = T_c - T_m \text{ and } U = f(E) \tag{1}$$

The function f represents the thermal controller (PID). In the temporal domain the output U of the corrector is the convolution product between the error measured E and the corrective action K of the controller $U(s)=K(s)E(s)$ or with the Laplace transform $U(s)=K(s)E(s)$. The knowledge of the function $K(s)$ of a controller is very important for calculation of the correction U from the error E. This function is independent of the nature of the input E.

Basic components of the PID controllers [2–4]

In Fig. 1 we have shown a control loop which contains a thermal controller. This is the most important part of the system which today can be realized with different kinds of modern controllers: PID (proportional integral derivative), Smith's predictor [5], RST [6, 7], fuzzy controllers [8, 9]. The most modern ones are fully developed and used with computers. PID controllers are industrially used on a very large scale. At the beginning they were designed analogously. It is important to keep in mind that the modern numeric PID controllers are only a sampled release of these electronic systems. The three command laws of a PID controller are:

Proportional command law

The output U is calculated by applying an amplification gain K_p on the error E. The larger E is, the bigger the correction U is. The value of the amplification gain is the parameter K_p and this can be materialized by a potentiometer on the electronic controller. If the value of P is too large the correction U will be too large and the error E will give oscillating values. On the other hand, if the value of P is too small, the correction will run without oscillations but too slowly. In the real systems a security device must be included in order to limit the correction U to avoid the destruction of the power stages of the controller. Above a given value for the error E we will have an overloading for the correction U. Under this value we have a proportionality between E and U; this area is named 'proportional band' so we have in the time space:

$$U(t) = K_p E(t) \tag{2}$$

and in the Laplace domain:

$$U(s) = K_p E(s) \tag{3}$$

This proportional command law has some drawbacks. For example it usually lets a little residual error remain because if the error E=0 the process could no longer run. Moreover if the value of the error E is too large (for example when the process starts) the correction will immediately overload the process. Some studies in the literature [2 in page 85] show that the proportional command law must be kept in order to set the stability of a control loop.

Integral command law

To avoid some drawbacks of the proportional command law it would be desirable to generate the correction U in a progressive way. To realize this goal an integral command law is used and represented by:

$$U(t)=\frac{1}{T_i}\int_0^t E(x)\mathrm{d}x \tag{4}$$

and in the Laplace domain [10]:

$$C(s)=\frac{U(s)}{E(s)}=\frac{1}{sT_i} \tag{5}$$

We see that the correction U is linked to the error E by an integral relation in which a very important constant takes place: T_i, its name is 'integral time'. T_i represents the time constant of the integral command law and is expressed in minutes or seconds. This parameter is usually set by means of its inverted form $1/T_i$ expressed in $\min^{-1}$ or s^{-1} and noted K_i. However K_i is rather given in 'repetitions by minute'. The integral command law gives to the system a progressive behaviour. If the value of T_i is too small, the correction U will be too fast and there is not a progressive start of the command law. Conversely if T_i is too large, the correction U is too slow and the start will be quite 'soft'. The integral command has an advantage over the proportional one. When the signal E is equal to 0, the correction U can be different from 0 and keeps its current value. In fact this progressive law is also persistent. The correction U will increase (or decrease) as long as a positive (or negative) error E exists.

Derivative command law

The above two command laws have a small drawback. When they determine a correction U from the error E, the calculation is conducted on the difference between a desired value and the measured corresponding one. But we must always keep in mind that if we are in a running system, the error E can have very different values owing to the time. Thus we must consider at a time T that the error E can be increasing or decreasing. So for the calculation of the correction U at a time T it will be very good to take the sign of the variation of the error E and the speed of this variation into account. For example, if the error E is currently slowly decreasing the correction will be smaller than if the error is strongly increasing. This is a very important notion which shows the importance of the history of the error. There is a mathematical function which can present these features: the derivative. The idea of the derivative command law is to calculate the correction U from the derivative of the error E. We have:

$$U(t)=T_{\mathrm{d}}\frac{\mathrm{d}E(t)}{\mathrm{d}t} \tag{6}$$

and in the Laplace domain [10]:

$$C(s)=\frac{U(s)}{E(s)}=sT_{\mathrm{d}} \tag{7}$$

The value T_{d} is named differential time. It is expressed in seconds or minutes. Usually this parameter is noted K_{d} and named 'derivative action coefficient'. The use of a derivative corrector brings about a fast correction of the error E without the risk of an overrunning of the desired value. The derivative command is in charge of the dampening or elimination of this overrunning brought about by the integral command law. More generally, this command law improves the stability of a system and is necessary in the systems which have a high order.

Remark concerning the derivative command law

In the analog domain the derivative controller is filtered by a lowpass filter. This operation is conducted in order to avoid the amplification of the noise present in the signal, because the derivative operator increases this phenomenon for noise, which is a kind of signal having fast variations. The transfer function of such a derivative is given by the relation below that we will include in our further calculations [11]:

$$C(s)=\frac{T_{\mathrm{d}}s}{1+\frac{T_{\mathrm{d}}}{N}s} \tag{8}$$

These three fundamental command laws can be combined to build a controller. If we want only the PI command laws, the contribution D will be set to zero. With a PID controller the three contributions complement each other. For a large error E it is the proportional factor which predominates with the help of the derivative factor at the start of the correction (time T=O$^+$). During the return at the equilibrium it is the integral command which finish the correction, being slowed down by the derivative correction.

Structure of the PID corrector in the DSC 111 calorimeter

We noted above that a PID controller is a combination of these three basic components. It exists mainly three kinds of structures: the parallel, mixed and serial structures. The analog releases of the DSC 111 began to be manufactured in the seventies. Their PID controller is fully analog and realized with discrete electronic components. The three stages P, I and D are made with operational amplifiers. We determined that this structure belongs to the mixed family. The transfer function is given in the relation:

$$C(s)=K_p\left(1+\frac{1}{T_i s}+T_d s\right) \tag{9}$$

This is the most widely used structure in the small controllers. We can see in Fig. 2 that the factor K_p concerns the integral and derivative command laws.

The transfer function is then described by the relation (9), which is the well-known form of the PID presented in the literature.

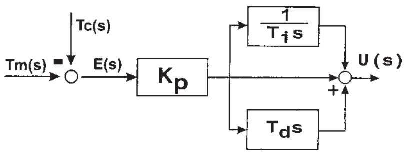

Fig. 2 Mixed structure of a PID controller

Digitalization of the analog PID controllers [1]

The above analog relations are established in the analog domain. Thus it is necessary to write them in the digital domain in order to develop the corresponding algorithms for computers. An interesting method consists in using an analog transfer function which will give an analog controller. Theoretically the determination of this controller is conducted from experiments. Then the controller is digitalized in order to obtain the corresponding numeric transfer function which will be implemented on a desktop computer. We decided to modify this method in order to allow the user to monitor the choice of the tuning parameters of a digitalized PID controller. The software developed graphically displays in real time the evolution of the correction calculated according to desired values and owing to the set points chosen. All the parameters can be modified at any time and their new values are immediately taken into account.

Numerization of the PID simulator [7]

We will use the mathematical notation generally used in the numeric domain with:

h – the value of the sampling rate (called ‘step’). This has a time dimension.
k – number of steps (other values can be used: 1, m, n...)

Exactly as for the analog domain, a discrete convolution product is defined, because the output of a numerized system will be obtained from the convolution of the numerized input and the numerized transfer function. In our study the numerized system is the PID controller, the output is the correction *U* calculated, and the numerized input is the error *E* between the measured and desired temperatures. For our whole work we will consider that the system is linear, causal and stationary. Then it is possible to write:

$$u(kh)=\sum_{l=0}^{k} e(lh)k(kh-lh) \tag{10}$$

This is the classical form of a product of convolution. This relation means that the output $u(kh)$ is expressed as the convolution product of the input $e(lh)$ and the impulse response $k(kh-lh)$ of the system. The knowledge of the impulse response allows the calculation of the output $u(kh)$ according to any input $e(lh)$. We can determine the discrete transfer function with the use of the *Z* transform.

$$K(z)E(z)=\left(\sum_{k=0}^{\infty} k(kh)z^{-k}\right)\left(\sum_{k=0}^{\infty} e(kh)z^{-k}\right)=$$

$$=[k(0)+k(h)z^{-1}+k(2h)z^{-2}+\ldots][e(0)+e(h)z^{-1}+e(2h)z^{-2}+\ldots] \tag{11}$$

If we develop and factorize:

$$=e(0)k(0)+[e(h)k(0)+e(0)k(h)]z^{-1}+[e(2h)k(0)+e(h)k(h)+e(0)k(2h)]z^{-2}+\ldots$$

$$\ldots+\left[\sum_{l=0}^{k} e(lh)k(kh-lh)\right]z^{-k}+\ldots$$

$$=\left[\sum_{k=0}^{\infty} e(lh)k(kh-lh)\right]z^{-k}$$

So we obtain:

$$K(z)E(z)=U(z) \text{ or } K(z)=\frac{U(z)}{E(z)} \tag{12}$$

Thus the *Z* transform allows, with this important relation in the digital domain, the same property that was brought about by the Laplace transform in the analog domain. We name the value 'discrete transfer function', then we obtain a ratio which is independent of the kind of input.

Methodologies for the numerization

A first possibility for designing a numerized converter is to include a numerized PID controller with cascade converters in the loop of regulation.

A second method consists in directly writing numerical approximations of the analog PID regulator.

We are now going to describe this second approach. First, we have to apply numeric approximations on the analog PID controller where the transfer function is given in relation (9) in which we include the lowpass filter given in relation (8). That will give the following transfer function:

$$K(s)=K_p\left(1+\frac{1}{T_i s}+\frac{T_d s}{1+\frac{T_d}{N}s}\right) \tag{13}$$

As different numeric approximations exist, we will choose the one which will bring about good stability in the numerized system.

Stability of a system [1, 3–5]

In our development we must be sure that for all bounded input of the numerized PID we will obtain a bounded output. This is the very important BIBO stability (bounded input bounded output) [1 in page 152]. When we use Z transforms, there is a theorem which says that: 'A discrete awaiting system, linear, causal and stationary which is described with a rational proper function will have a BIBO stability if and only if all its poles are in the unit circle'. This notion of unit circle in discrete domain is equivalent to the notion of the imaginary axis in the analog domain. With this theorem we know that if we have a pole in this circle we will obtain in the time domain a bounded and deadened signal. If the pole is exactly on the circle the signal will be bounded but not deadened.

Numerical approximations [1, 12]

To develop the discrete algorithm we are going to use numerical approximations. More generally a system such as our controller can be described by the well-known differential analog equation:

$$\begin{aligned} u^n(t)+a_{n-2}\frac{d^2u(t)}{dt^2}+a_{n-1}\frac{du(t)}{dt}+a_n u(t)= \\ = e^n(t)+b_{n-2}\frac{d^2e(t)}{dt^2}+b_{n-1}\frac{de(t)}{dt}+b_n e(t) \end{aligned} \tag{14}$$

The functions $u(t)$ and $e(t)$ respectively represent the output and the input of the analog PID controller in the time domain. We can now write the relation in a corresponding discrete form

$$u^{n}(kh)+a_{n-2}\frac{d^{2}u(kh)}{dkh^{2}}+a_{n-1}\frac{du(kh)}{dkh}+a_{n}u(kh)=...$$
$$...=e^{n}(kh)+...+b_{n-2}\frac{d^{2}e(kh)}{dkh^{2}}+b_{n-1}\frac{de(kh)}{d(kh)}+b_{n}e(kh) \tag{15}$$

Now we are going to show some different ways in order to replace the discrete quantities by numeric equivalent approximations. If we use the 'first Euler's method' we have the following approximations which are easy to demonstrate:

$$\frac{du(kh)}{dkh}=\frac{u(kh+h)-u(kh)}{h}$$

and

$$\frac{d^{2}u(kh)}{dkh^{2}}=\frac{u(kh+2h)-2u(kh+h)+u(kh)}{h^{2}} \tag{16}$$

This first Euler's method is often also named 'Forward approximation' because it is used for the calculations the present time 'kh' and future steps such as '$u(kh+h)$'. It implies that we have a future knowledge of the steps that can sometimes be a drawback. If we put the results of (16) and the equivalent forms for $e(t)$ in the Eq. (15) we will obtain, with the general properties of time shift of the Z transform and a final identical with (15), the equality:

$$s=\frac{z-1}{h}\text{ and }z=sh+1 \tag{17}$$

We remark that for a real negative value of s (half left plane) we will have a real z value lower than 1. Thus, if we have a BIBO stable analog PID controller, we cannot obtain a BIBO equivalent discrete PID controller. In the second Euler's method we will use the following approximations:

$$\frac{du(kh)}{dkh}=\frac{u(kh)-u(kh-h)}{h}$$

and

$$\frac{d^{2}u(kh)}{dkh^{2}}=\frac{u(kh)-2u(kh-h)+u(kh-2h)}{h^{2}} \tag{18}$$

This second method is also named 'Backward approximation' because it is used for the calculations the present time 'kh' and past steps such as '$u(kh-h)$' which are known. With the same kind of calculations as for the former method we obtain:

$$s=\frac{1-1/z}{h}=\frac{z-1}{zh}\quad\text{then}\quad z=\frac{1}{1-sh} \tag{19}$$

If we study the BIBO stability criterion (by using the complex notation: $s=a+jb$) for this method we find that a stable analog PID controller will give a BIBO discrete

PID controller. This second method is often preferred for these qualities. Usually the forward method is performed to discretize the integral term and the backward method is rather used to discretize the differential term. For the digitization of the integral term we replace s by the numeric approximation from the relation (17) and obtain for this term:

$$\frac{1}{T_i s}=\frac{h}{T_i(z-1)} \tag{20}$$

For the digitization of the differential term we replace s by the expression from relation (19) and obtain:

$$\frac{T_d s}{1+\frac{T_d}{N}s}=\frac{NT_d\frac{z-1}{zh}}{N+T_d\frac{z-1}{zh}}=\frac{N(z-1)}{\frac{Nhz}{T_d}+z-1}=\frac{N(z-1)}{\left(\frac{Nh}{T_d}+1\right)z-1} \tag{21}$$

If we use the relations (20) and (21) in the equation of the analog PID controller we will have the final numeric PID controller:

$$K(z)=K_p\left(1+\frac{h}{T_i(z-1)}+\frac{N(z-1)}{\left(\frac{Nh}{T_d}+1\right)z-1}\right) \tag{22}$$

It is this method of digitization which is used in our software; thus we must adapt the numeric expression (22) in order to implement it on a computer. To realize this implementation we separate the calculations for each command law (P, I and D) as shown in Fig. 2. As we have $U(z)=K(z)E(z)$ we will obtain:
For the integral command law:

$$U_i(z)=\frac{h}{T_i(z-1)}E(z)=\frac{hz^{-1}}{T_i(1-z^{-1})}E(z) \tag{23}$$

It is important to write negative powers of z because they represent past discrete steps; it is one of the most interesting properties of the Z transform.

For the differential command law:

$$U_d(z)=\frac{N(z-1)}{\left(\frac{Nh}{T_d}+1\right)z-1}E(z)=\frac{N(1-z^{-1})}{\frac{Nh}{T_d}+1-z^{-1}}E(z) \tag{24}$$

Relations (23) and (24) are very important because they show now the writing of current signals $U(z)$ and $E(z)$ owing to preceding steps represented by z^{-1}. In the time domain they will become (*kh–h*). Knowing that we can finally write the time discrete equivalent equations:

1 – Writing of the discrete error (input signal of the PID controller):

$$e(kh)=T_c(kh)-T_m(kh) \tag{25}$$

At a time discrete *kh* we have for the error *e*(*kh*) a simple difference between the desired temperature T_c and the effective measured temperature T_m.

2 – Writing of the differential discrete command law:

From relation (24) we can write:

$$U_d(z)\left(\frac{Nh}{T_d}+1-z^{-1}\right)=E(z)N(1-z^{-1}) \tag{26}$$

We develop and rearrange owing to the powers of *z*:

$$U_d(z)(T_d+Nh)-T_dU_d(z)z^{-1}=NT_dE(z)-NT_dE(z)z^{-1}$$

Now we write the equivalent relation in the time domain:

$$u_d(kh)(T_d+Nh)-T_du_d(kh-h)=NT_de(kh)-NT_de(kh-h)$$

And finally we obtain the directly codable discrete relation:

$$u_d(kh)=\frac{T_d}{(T_d+Nh)}\Big(u_d(kh-h)+N\big(e(kh)-e(kh-h)\big)\Big) \tag{27}$$

3 – Writing of the integral discrete command law:

From relation (23) we can write:

$$U_i(z)(1-z^{-1})=\frac{h}{T_i}E(z)z^{-1}$$

We develop and rearrange with the same approach as for the relation (27) and obtain:

$$u_i(kh)=\frac{h}{T_i}e(kh-h)+u_i(kh-h) \tag{28}$$

4 – Writing of the final equation of the discrete PID controller:

The last operation consists in combining the above two command laws ((27) and (28)) with the proportional law K_p according to the relation (9) which describes the mixed structures of our discrete numeric PID controller. Then we obtain:

$$u(kh)=K_p\big(e(kh)+u_i(kh)+u_d(kh)\big) \tag{29}$$

Relation (29) means that the output (correction) of the discrete PID controller at a time *kh* is given by the 3 PID command laws at the same time *kh*. We note that the I and D corrections are built with the knowledge of past steps (at time *kh–h*). The choice of the sampling rate *h* is very important for the behaviour of the controller.

Usually we consider that the value of the sampling rate is smaller than the tenth of the response time of the thermal process.

For us the response time of the process is represented by the response time of the DSC 111 itself. According to the kind of thermal experiments for which we develop this software, the response times go from about 10 up to about 40 s. Thus we will choose sampling rates *h* smaller than one second. Under this condition the discrete behaviour of our PID controller will be equivalent to the same analog one. The correction $u(kh)$ in relation (29) is fully calculated at each time *kh* from the error $e(kh)$. These kinds of controllers are named 'absolute correctors', 'standard correctors' or 'correctors under position form'.

Overloading of the PID controller [5, 12]

Due to the presence of integral calculations there is a well-known overloading problem in controllers which use integral command law. If we have a large error $e(kh)$ with the same sign for a long time, the integral correction calculated with relation (28) will have a very large value which will overload the output of the controller. This phenomenon must be avoided in order to protect the power stages of a thermal regulation and perhaps the process studied itself. From a mathematical point of view we must wait for a long time for the error $e(kh)$ with an opposite sign to go under the overloading value. This phenomenon brings about instability and oscillations into the system. In the domain of regulation it is called 'reset windup' or 'integral windup'. Methods exist in order to avoid this drawback. Usually these methods are called 'antireset windup methods'. We will call them 'ARW'. Fundamentally all these calculations seek to limit or suppress the overload of the integral command law.

We present here only two ARW methods which are included in our software. They concern the calculation of the integral term in order to maintain it under the overloading conditions:

Limitation of the calculated integral term

The quantity $u_i(kh)$ will normally be calculated only if the global quantity calculated $u(kh)$ in relation (29) is smaller than a maximum value $u(kh)_{max}$. Otherwise we have:

$$u_i(kh)=u_i(kh-h) \text{ if } |u(kh)|>u(kh)_{max} \tag{30}$$

Thus in this case the choice of the determination of $u_i(kh)$ depends on all of the three quantities $e(kh)$, $u_i(kh)$ and $u_d(kh)$. With this method we note that the quantity $u(kh)$ can be larger than $u(kh)_{max}$, so we can sometimes have overloading of this output.

The standard ARW method

This method is chosen when we want in every case to avoid the overloading of the output. Thus if the calculated value of the general output $u(kh)$ from relation (29) must be smaller than a given maximum $u(kh)_{max}$ we will have the following calculations for the integral term $u_i(kh)$:

$$u_i(kh)=\frac{u(kh)_{max}}{K_p}-e(kh)-u_d(kh) \text{ if } u(kh)>u(kh)_{max}$$

$$u_i(kh)=-\frac{u(kh)_{max}}{K_p}-e(kh)-u_d(kh) \text{ if } u(kh)<-u(kh)_{max} \quad (31)$$

This method certainly gives an output $u(kh)$ under the overloading conditions of the controller, but we can note that in the determination of the integral term the error and the derivative terms are involved. This method is the default one used by the software.

Elaboration of the measured temperature T_m

In our software we have not included a real model of the process, the temperature of which is controlled by the discrete PID controller. We can see in Fig. 3 that the functional box called 'Calorimetric block' has $G(s)$ for transfer function such as $T_m(s)=G(s)U(s)$.

From a physical point of view this means that the measured temperature T_m depends directly on the power brought by the correction U of the controller. In the real DSC 111 this power U is built by Joule effect. This power is analogous to a thermal

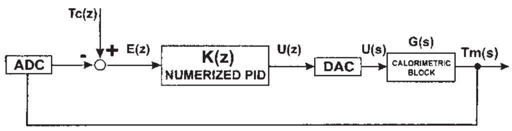

Fig. 3 The numerized PID loop in Z transform

flow, so we can write:

$$\vec{\Phi}=-\lambda\vec{\nabla}T \quad (32)$$

We note that the thermal flow is proportional to the gradient of temperatures created by the flow inside the thermal block. The calorimetric blocks of calorimeters have a very simple configuration, so for our model we will write that the difference of temperatures is proportional to the correction output U of the controller. The difference of temperature (ΔT) is the algebraic difference between the temperature for a present step and the temperature for the previous one. If we combine this difference in the discrete domain, we will have:

$$\Delta t_m=t_m(kh)-t_m(kh-h)=\lambda.u(kh)$$
$$\Rightarrow t_m(kh)=t_m(kh-h)+\lambda.u(kh) \quad (33)$$

Important remark

In reality the calorimetric block acts as a filter between the correction U and the temperature measured T_m, so if we would determine exactly the relation between these quantities we must conduct complex experiments to finally obtain the real transfer function which best represents the behaviour of the block for given environments. But for the moment we are only interested in the behaviour of our numeric PID controller confronted with the evolution of the error between the desired temperature $t_c(kh)$ and the measured temperature $t_m(kh)$. So we will only use the measured temperature $t_m(kh)$ combined with the correction $u(kh)$ only to generate a new measured temperature $t_m(kh)$. The result of this combination will be supposed to be the effective temperature measured.

From a calorimetric point of view relations (33) take into account the fact that the temperature measured in a calorimetric block follows the correction sent by the controller. According to the former remark the value of the parameter λ will be chosen in order to give a correct behaviour to the numeric controller. We performed experiments with different values for λ and finally fixed this parameter at 0.1 which seems to represent a good compromise. So we have:

$$t_m(kh)=t_m(kh-h)+\frac{u(kh)}{10} \tag{34}$$

Development of the software

In the software developed we used the equations presented in relations (25), (27), (28) and (29). As these equations did not include ARW algorithms we added the ability of the software to take the ARW Eqs (30) and (31) into account. We have provided the ability for the user to change the value of the most important parameters of the calculations at any time. A delay parameter is included to allow the modification of the speed of the calculations in order to better simulate the behaviour of the calorimeter. So this delay can create an artificial time constant close to that of the calorimeter. Thus we have the following six parameters:

K_p – proportional term
T_i – integral term
T_d – derivative term
N – parameter of the lowpass filter in the derivative term
h – sampling rate in the numerization of the analog PID controller
T – time delay that we have added in order to slow down the speed of the calculations. This allows adjustment of the performances of our software to those of a real calorimeter, in order to better simulate its dynamic behaviour.

As they are less often modified, the choice of the kind of ARW algorithms and the quantity λ from relation (34) are performed directly at the C code level. The name of the software developed is PIDX. On a computational point of view the output is fully graphic and the program does not need a mouse. All the current values for the above parameters are always displayed. Several kinds of signal are proposed for the temperature desired t_c. The user makes his choice among the following signals proposed: sine, rectangular, triangular and logarithmic signal. At any time the user can change the kind of signal by pressing the corresponding numeric key shown in the menu bar. Then the current parameters are immediately applied to the new signal chosen.

In addition, the user can select two special signals: a manual signal so that the user (with the keyboard) can adjust in real time the shape of the desired temperature, and a file signal which can be a real or a simulated temperature profile which is stocked beforehand on the hard disk of the computer.

When the software is started the default signal is rectangular because it shows the performance of the numeric PID with very fast changes in temperature. The following default parameters are taken into account prior to a different choice of the user: K_p=2, T_i=2, T_d=1, N=10, h=0.8 and T=13. Figure 4 shows the default display when the program is started. On the same screen are displayed in real time the evolution of the desired temperature (rectangular one in Fig. 4) and the temperature measured (decreasing since response). When the signals come at the right end of the screen, the display is cleared and begins again from the left of the screen.

At the bottom there is a menu bar which summarizes the commands available. Figure 5 shows more precisely the options proposed.

All the parameters are set with the keyboard. Figure 5 specifies the following settings:

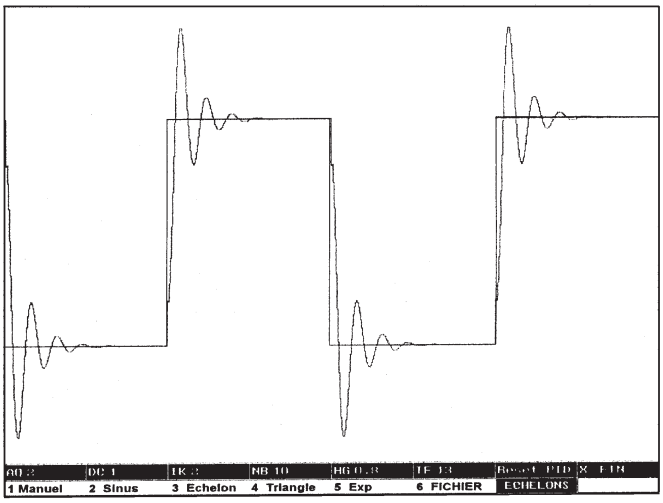

Fig. 4 View of the graphic display at the start of the software

Setting PID parameters

For PID the numeric parameters are directly set by 3 couples of keys. For example, keys A and Q allow the modification of the proportional term with respective incre-

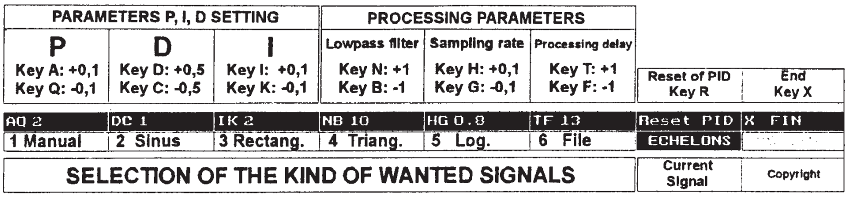

Fig. 5 Detailed view of the menu bar

ment +0.1 and –0.1. Couples of keys D/C and I/K control respectively the derivative and integral parameters.

Setting the processing parameters

The lowpass filter of the derivative command law, the sampling rate and the time delay between two consecutive calculations are set respectively by the couples of keys N/B, H/G and T/F. Figure 5 specifies the values of the corresponding increments.

Selection of signals for the desired temperature

This selection is made by pressing the numeric key corresponding to the name of the signal; for example, press key 4 to choose a triangular shape for the desired temperature.

Resetting the parameters

Key R allows the user to reset all the above parameters. The default values are then taken into account.

The signal chosen is permanently displayed on the menu bar. Key X is used to exit the software. The calculations of all the discrete quantities (for example $u(kh)$) are indexed on the number of horizontal pixels of the computer display. So we have established a correspondence between the evolution of the discrete steps kh (which have a time dimension), the evolution of the number of screen pixels i which corresponds to the successive calculations, and the time delay between two calculations. Thus we have the relation:

$$kh \quad \rightarrow \quad i\ delay \tag{35}$$

with:

i – number of the screen pixels (from 1 to i_{max}=640 for a VGA display)

delay – time between two consecutive computer calculations

The numeric equations developed for the generation of the set point signal are forecast to have a representative behaviour in the width of the display. The time delay is chosen by the user. It represents a break between two calculations, it is expressed in seconds. When the delay is equal to zero, the software does not wait between calculations, so the effective delay between calculations is the one that is really taken by the instructions of the program.

References

1 R. Longchamp, ‘Commande numérique de systèmes dynamiques’, Presses polytechniques et universitaires romandes, Lausanne 1995.
2 C. Sermondade and A. Toussaint, Régulation, Tome 1: ‘Régulation élémentaire,’ Nathan, Paris 1994.
3 C. Sermondade and A. Toussaint, Régulation, Tome 2: ‘Identifications stabilité réglages’, Nathan, Paris 1994.
4 M. Zelazny, F. Giri and T. Bennani, ‘Systèmes asservis: commande et régulation, Tome 1: ‘Représentations analyse performances’, Eyrolles, Paris 1993.
5 M. Rivoire and J. L. Ferrier, ‘Cours d’automatique’, Tome 2: ‘Asservissement régulation commande analogique’, Eyrolles, Paris 1997, 112.
6 P. Boucher and D. Dumur, La commande prédictive, Editions Technip, Paris 1996, 51.
7 M. Rivoire and J. L. Ferrier, Cours d’automatique, Tome 3: Commande par calculateur, Eyrolles, Paris 1997.
8 H. Scholten, Logique floue et régulation PID, Publitronic, Nieppe (France) 1994.
9 B. Bouchon-Meunier, La logique floue, Collection Que sais-je?, Presses Universitaires de France, Paris 1993.
10 M. Zelazny, F. Giri and T. Bennani, Systèmes asservis: commande et régulation, Tome 2: Synthèse applications instrumentation, Eyrolles, Paris 1994, 309.
11 F. Cottet, Traitement des signaux et acquisition de données, Dunod, Paris 1997.
12 A. Rachid, Systèmes de régulation, Masson, Paris 1996.
13 D. Brunel and L. Elegant, J. Therm. Anal. Cal., 55 (1999) 1.