

Software for Thermogravimeter

František Dušek

Dept. of Process Control
University of Pardubice
Pardubice, Czech Rep.
<https://orcid.org/0000-0002-6549-3112>

Daniel Honc

Dept. of Process Control
University of Pardubice
Pardubice, Czech Rep.
<https://orcid.org/0000-0001-8440-5272>

Libor Havlíček

Dept. of Process Control
University of Pardubice
Pardubice, Czech Rep.
<https://orcid.org/0000-0001-6874-5971>

Jan Merta

Dept. of Process Control
University of Pardubice
Pardubice, Czech Rep.
<https://orcid.org/0000-0002-1521-5050>

Abstract—Software for thermogravimeter constructed at the Department of Inorganic Technology, University of Pardubice is presented in the paper. Used hardware and design of the experiment is described briefly. Main part of the paper focusses to the software solution in MATLAB environment. Except control, plotting and data exporting tasks it was necessary to communicate with Eurotherm nanodac Recorder / Controller through the TCP/IP Modbus protocol and with Sartorius weigh cell through the serial USB ASCII-based protocol. Simple example of Constant Rate Thermal Analysis experiment is given at the end of the paper.

Keywords—Thermal Analysis, Thermogravimetry, control, Eurotherm nanodac, Sartorius weigh cell, MATLAB

I. INTRODUCTION

Phase transformations of solids are accompanied by heat absorption or heat release, changing dimensions, changing magnetic, electrical, mechanical and other properties. Thermal analysis is one of the methods how to explore such processes [1], [2]. Classical Thermal Analysis (TA) is based on the registration of released or absorbed heat by the phase change or chemical reaction. More sensitive method is a Differential Thermal Analysis (DTA) based on measuring the temperature difference of the measured and reference sample. Dynamic methods of the thermal analysis are used more frequently for the solids - the sample is heated or cooled according to the specified temperature profile. One of the basic methods of the dynamic thermal analysis is Thermogravimetry (TGA) where the sample weight change is monitored during its continuous heating or cooling. The result is so-called thermogravimetric curve expressing change in weight as a function of the temperature. Commercially available thermogravimeters are e.g. products from American company TA Instruments [3] or French company Setaram [4].

The experimental device for the dynamic thermal analysis is principally composed of two parts. A measured sample is placed in the first, heated part, and the second part is a sample weight measurement system. The device must be able to change the heating power and measure the temperature and weight of the sample. The experimental data are obtained by measuring the sample weight at a temperature according to the specified temperature time profile. The temperature can be also changed to maintain the sample weight time profile – so called Controlled (Constant) Rate Thermal Analysis (CRTA). In both cases it is an active experiment i.e. following the desired temperature profile is ensured by a change of the heating power. In the first case the temperature profile is planned before the experiment and in the second case the temperature profile is continuously changed so that the desired sample weight profile is followed.

The paper describes principle of the software functionality and its implementation in the MATLAB environment for a device developed at the Department of Inorganic Chemistry at the Faculty of Chemical Technology, University of Pardubice. Device is constructed according to the thermogravimeter at CSIC, University of Seville [5], [6]. The program allows both types of the experiments to be automatically executed. The program is not designed for the final use. Its purpose is to verify the functionality of the prototype and applicability for the thermal analysis up to the temperature about 800 °C.

The paper is structured as follows. Introduction is in chapter 1. Experimental device is described in chapter 2. In chapter 3 design of the TGA experiment is formulated. Software components are presented in chapter 4. Results and conclusions are given in chapter 5.

II. EXPERIMENTAL DEVICE

The scheme of the device including the communication interfaces is shown in Figure 1. The ceramic sample cell is indirectly heated by the electric heater with the power controlled with the Pulse Width Modulated (PWM) control signal generated by the Eurotherm nanodac Recorder / Controller unit. The relative power P can be set between 0 and 100 %. 100 % corresponds to the heating power

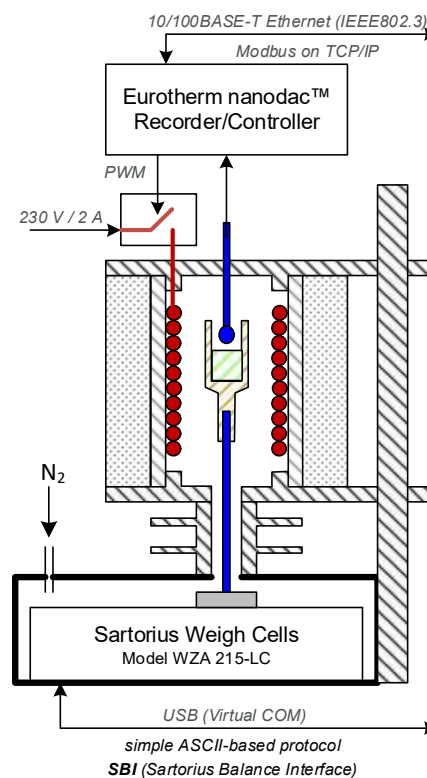


Fig 1. Thermogravimeter scheme



Fig. 2. Thermogravimeter

approximately 460 W. The temperature of the sample T_m is measured by a K thermocouple (measurement up to about +1000 °C) located above the sample cell – see Figure 2. The thermocouple signal is evaluated by the Eurotherm unit. Temperature in degrees with resolution 0.1 °C and the relative heating power are transmitted via the Ethernet communication interface through the Modbus TCP protocol (port 502). The weight of the sample is measured by the Sartorius weigh cell WZA215-LC. The weighing capacity of this weight unit is 210 g with readability 0.01 mg. Measured weight is

transmitted via USB 2.0 (virtual COM port) as ASCII characters through the Sartorius Balance Interface (SBI) protocol.

III. DESIGN OF THE EXPERIMENT

The experiment is divided into two phases as shown in Figure 3. In the first phase the temperature of the sample should linearly increase from the initial measured value T_0 to the set-point temperature T_{w1} in time t_1 . The set-point is kept constant for time slot t_2 . The temperature T_m and the mass of the sample expressed as relative mass p_m are measured. The relative mass p_m is based on the weight m_0 of the sample measured at the beginning of the experiment. The sample temperature is controlled by the heating power P – it follows the desired profile $T_w(t)$.

The second phase of the experiment begins at time $t_1 + t_2$ from the beginning of the experiment. At this point the desired weighting profile $p_w(t)$ is calculated from the current value of the relative weight of the sample p_{m2} , the time slot t_3 and the desired final weight p_{w3} . At the same time cascade control is triggered - it ensures following of the desired weight profile $p_w(t)$ by changing the desired temperature $T_w(t)$ of the sample. The experiment is terminated after the time $t_1 + t_2 + t_3$ is reached from the beginning of the experiment.

IV. SOFTWARE

A. Programming environment

MATLAB was chosen as a development and programming environment due to the fact that it was necessary to ensure proper functionality of the device but also to analyze and design the temperature and weight control systems. Except number of data processing and visualization functions MATLAB is also an object-oriented programming language that enables complex programs to be created, with easy and fast editing capabilities allowing simple tuning and testing. Support of the standard communication interfaces is however

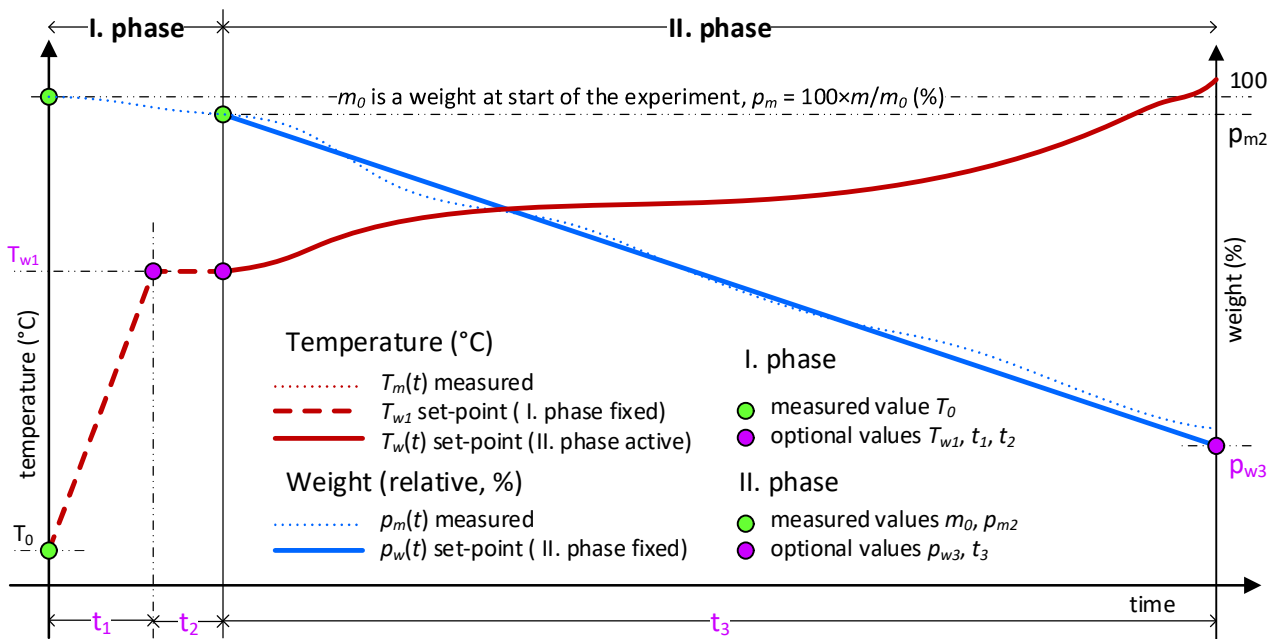


Fig. 3. Planned course of the experiment

also very important feature. MATLAB contains support for the serial communication using COM port and the Instrument Control Toolbox includes support for the TCP/IP network communication.

B. Communication with Eurotherm nanodac

The standard network card was used to communicate with this device. The physical connection was made with the network cable with RJ45 connector. The software solution for the communication with the device uses the `tcPIP` object from the MATLAB Instrument Control Toolbox. When creating the `tcp` instance of the `tcPIP` object, the device's IP address and the port 502 that is reserved for the Modbus protocol must be specified.

```
tcp=tcPIP('192.168.0.111',502);
```

The default `fopen` function is used to establish the communication.

```
fopen(tcp);
```

To obtain the measured temperature it is necessary to send a message in the binary format according to the Modbus protocol (`fwrite` function). Among other information the message contains the address on the device from which the value is read. The address and its format (data type) must be obtained from the device documentation.

```
mes=[transID;protID;length;ufCOD;Adr;regNum];
fwrite(tcp, mes, 'uint16')
```

After receiving a message (answer) in binary format according to the Modbus protocol the `fread` function removes the appropriate number of bytes from the system buffer and removes the requested value from the appropriate position in the response. Its format (data type) and meaning must be obtained from the device documentation again.

```
header=fread(tcp,9); % read first 9
    bytes (message header)
value=fread(tcp,1,'uint16'); % read
    next 2 bytes (value)
```

The heating power setting procedure is similar. The message being sent is in different format. Except the address it contains also the value to be written. It is necessary to wait for the response – it serves as a check whether the command was executed or not.

C. Communication with Sartorius weigh cell

The weighing unit is equipped with a USB-A connector cable. The device is recognized as Virtual COM and can be operated as a standard serial port. The software solution for the communication uses a `serial` object that is supported in the MATLAB. Communication is based on the Sartorius Balance Interface (SBI) protocol, a simple protocol that uses only basic ASCII characters. Message termination (byte sequence) is indicated by a pair of CR (13 Carriage Return) and LF (10 Line Feed) bytes. If the message starts with ESC (27 Escape) byte it is a command otherwise it is a message containing data (text string). Reading out of the sample weight is done by sending the "measure the weight" message (command) and waiting for the answer. The read request can wait until the sequence of successive bytes CR, LF is recognized and then sequence of received bytes is returned –

this must be set in the `Terminator` property of the `serial` object. Such a feature allows very simple communication assuming that waiting is not taking too long time. In order to avoid blocking of the program while waiting for a response it is possible to specify maximum wait time. This is done in the object's `Timeout` property - the read request is canceled and the execution of the program commands can continue after the timeout expiration. Unfortunately during the testing it was found that the unit in the default settings evaluates a change in the sample weight as a non-stabilized measurement (due to a temperature change) and does not send the response. This causes longer delays or even stopping the program for the whole reading timeout period. This is unacceptable for the correct functionality of the digital controllers - the interval between the measurement and the application of the control action compared to the sampling period must be as small as possible.

The program can use an asynchronous reading capability of the `serial` object. Asynchronous read operation means that the request is passed and the master program continues immediately with the next command. The service program starts independently of the master program if the message is received.

When creating an instance `ser` of a `serial` object it is necessary to specify the hardware settings parameters,

```
ser = serial('COM3','BaudRate',9600);
ser.DataBits = 8; ser.StopBit = 1;
ser.Parity = 'odd';
```

to define terminator and set timeout,

```
ser.Terminator='CR/LF'; ser.Timeout=5;
```

to enable asynchronous reading, to set read event (`ReadRoutine` handle function),

```
ser.ReadAsyncMode = 'continuous';
ser.BytesAvailableFcnMode = 'terminator';
ser.BytesAvailableFcn = @ReadRoutine;
```

and to make a program of the event handler that contains reading out of the data and another read request.

```
function ReadRoutine(obj,event)
% routine is called at an event
    'terminator is received'
% obj is a reference to serial object
s = fgetl(obj); % read string
    (without terminator)
obj.UserData.s = s; % save string
    into free object attribute
fwrite(obj,[ESC,P,CR,LF],'uint8'); %
    send new request
readasync(obj); % read without
    waiting for an answer
end
```

Now it is possible to open the communication with `fopen` function and to initiate communication by the request to measure the weight (`fwrite` function) followed by an asynchronous reading request (`readasync` function).

```
fopen(ser);
```

```
fwrite(ser, [ESC, P, CR, LF], 'uint8');
    % send new request
readasync(ser); % read without
    waiting for an answer
```

Since now the communications is running independently of the master program. In the master program it is possible to read a string containing the sample weight information at any time from the `ser.UserData.s` property of the object - it contains the newest weight measurement.

D. Control algorithms

The heuristic adaptive controller designed by Ing. Jaroslav Maršik, CSc. at UTIA CSAV Prague [7], [8] was applied for the temperature control. This controller was used because the nonlinear behavior of the controlled system and because that the properties depend on the thermocouple placement, the temperature, the quantity and the characteristics of the sample. It is difficult to find fixed parameters of a standard controller that would suit for the entire operating temperature range (100-800 °C) and for different sample weights.

It is a digital PID controller in positional form

$$P(k) = \alpha(k)[\beta(k)e(k) + \gamma(k)\Delta e(k)] + \sum_{i=1}^k \alpha(i) e(i) \quad (1)$$

$$e(k) = T_w(k) - T_m(k) \quad \Delta e(k) = e(k) - e(k-1)$$

where

$P(k)$ is heating power (0-100),

$T_w(k)$ is temperature set-point (°C),

$T_m(k)$ is controlled temperature (°C) and

α , β , and γ are updated parameters (every time instant).

Parameters α , β , and γ are changed so that the oscillation coefficient k_w of the control error e is maintained at the desired value k_{ww} ($0 \leq k_{ww} \leq 1$). The oscillation coefficient is evaluated indirectly i.e. there is no need that the control error crosses zero value. The algorithm from the literature [8] also includes saturations of the control action with wind up effect suppression.

The standard digital PID controller in incremental form was used to control the relative weight of the sample

$$T_w(k) = T_w(k-1) + q_0 e(k) + q_1 e(k-1) + q_2 e(k-2) \quad (2)$$

$$e(k) = p_w(k) - p_m(k)$$

where

$T_w(k)$ is temperature set-point (°C),

$p_w(k)$ relative weight set-point (%),

$p_m(k)$ is controlled relative weight (%) and

q_0 , q_1 , q_2 are fixed parameters of the control law.

Parameters q_0 , q_1 and q_2 are calculated for given sample time T_s from the continuous PID controller parameters r_0 , T_i and T_d according to

$$q_0 = r_0 \left(1 + \frac{T_s}{2T_i} + \frac{T_d}{T_s}\right) \quad q_1 = r_0 \left(1 - \frac{T_s}{2T_i} + \frac{2T_d}{T_s}\right) \quad q_2 = r_0 \frac{T_d}{T_s} \quad (3)$$

The influence of the different initial sample weights is reduced by the controlling of relative weight of the sample. Other nonlinearities are also compensated by the temperature controller.

E. Master program

The program is designed as a simple script without a user interface. All required parameters (see Table 1) are set as variables at the beginning of the script. The program starts to communicate with both measuring subsystems (Sartorius cell and Eurotherm nanodac), measures initial temperature T_0 and the mass m_0 of the sample, calculates the temperature profile T_{w1} ($0 \leq t \leq t_1 + t_2$) and the weight profile p_w ($t_1 + t_2 \leq t \leq t_1 + t_2 + t_3$) (for $p_{m2} = p_{m0}$) and plots the data of the experiment into graph - Fig. 4 without measured variables.

The first phase of the experiment starts - the temperature controller follows the temperature profile $T_{w1}(t)$ with use of the heating power $P(t)$. The actual measured values and other information are displayed at the MATLAB command line. Current heating power $P(t)$ and actual measured values of the temperature $T_m(t)$ and the relative weight $p_m(t)$ are continuously plotted to the graph.

After the time $t_1 + t_2$ from the beginning of the experiment the weight profile p_w ($t_1 + t_2 \leq t \leq t_1 + t_2 + t_3$) is updated based on the actual weight p_{m2} and the second phase of the experiment is started. A cascade controller follows the weight profile by changing the temperature set-point $T_w(t)$. The variables are saved to a file as a binary data as the experiment is finished.

V. RESULTS AND CONCLUSIONS

The aim was to verify the functionality of the dynamic thermal analysis device and to create a MATLAB program for the experiments. In the first phase it was necessary to verify the properties of the weight and temperature measurement subsystems and the heating power subsystem. It was also necessary to design temperature and relative weight control loops. Based on the experience gained in this phase the final version of the software will be created.

Experiments were performed on the prototype which was not blown with the inert gas and the heated part was not isolated. For these reasons, the maximum achievable temperature was approx. 400 °C and it was not possible to carry out experiments with substances with unwanted reactions due to the presence of the air oxygen. Therefore the verification experiments were carried out only with the drying of silica gel at a temperature of up to 200 °C.

Table 1 lists the optional parameters of the experiment. The sample weight of the silica gel was 0.2576 g. The parameters of the adaptive digital PID temperature controller and the digital equivalent to continuous PID controller of the relative weight were tuned by the trial-error experimental method.

The CRTA experiment is recorded in Figure 4. The experiment illustrates the measurement and control functionality in the worst operating temperature range. Due to the absence of active cooling the temperature increase is significantly faster than the decrease - caused only by the heat

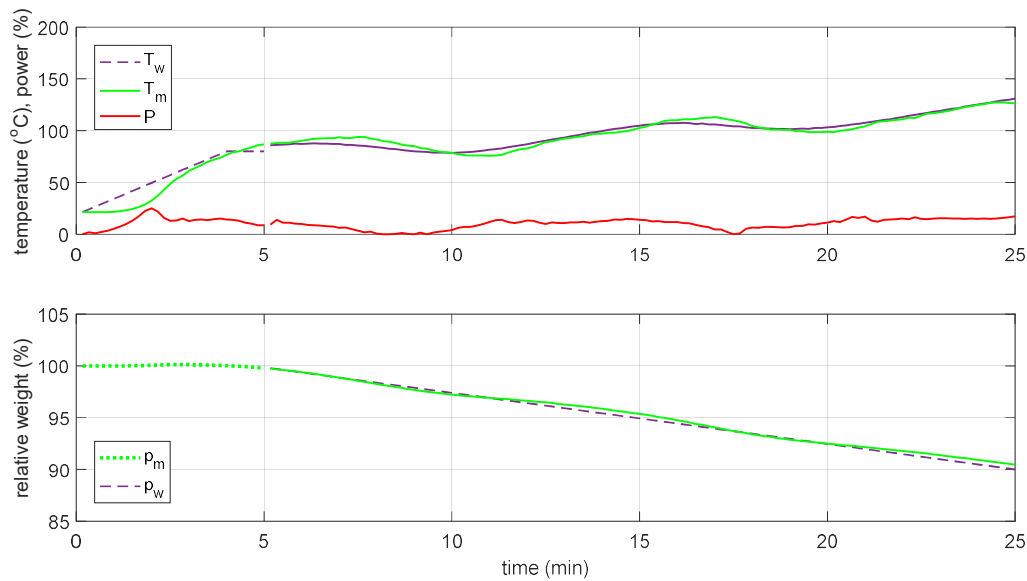


Fig. 4. CRTA experiment

losses to the environment. Oscillations of both the relative weight and the measured temperature occur for faster desired weight decrease. Such a dynamic processes degrade the shape of the thermogravimetric curve. This must be solved in the final version of the thermogravimeter hardware and software.

By the experiments, the time of the controlled weight decrease must be significantly longer (in hours) so that all the ongoing processes are driven closer to their steady states. Also the parameters of the controllers must be set in a way that there are no oscillations caused by the control.

TABLE I. PARAMETERS OF THE EXPERIMENT

Symbol	Value	Dimension	Meaning
t_s	10	s	Sample time
t_1	4	min	Time span for temperature to get the value T_{w1}
T_{w1}	80	°C	Desired initial temperature of the second phase of the experiment
t_2	1	min	Time spat for temperature to keep the value T_{w1}
t_3	20	min	Time span for relative weight to decrease to the value p_{w3}
p_{w3}	80	%	Desired relative weight of the sample at the end of the experiment
k_{ww}	0.2	-	Adaptive digital PID temperature controller parameter ($0 \leq k_{ww} \leq 1$)
r_0	-20	-	Gain of the digital PID weight controller
T_i	100	s	Integral time constant of the digital PID weight controller
T_d	0	s	Derivative time constant of the digital PID weight controller

ACKNOWLEDGMENT

This research was supported by Institutional support of The Ministry of Education, Youth and Sports of the Czech Republic at University of Pardubice and SGS grant at Faculty of Electrical Engineering and Informatics.

REFERENCES

- [1] M.E. Brown (editor), "Handbook of thermal analysis and calorimetry," Vol. 1 - Principle and practice, Elsevier, 1998
- [2] E. Benavidez (editor) "Thermal analysis applied to complex systems: Slags, Glasses and Ceramics," Asociación Argentina de Materiales, 2013
- [3] Thermogravimetric Analyzers, online <https://www.tainstruments.com/products/thermal-analysis/thermogravimetric-analysis/> (cited 7.1.2019)
- [4] LABSYS EVO STA (SIMULTANEOUS THERMAL ANALYSIS), online <https://www.setaram.com/setaram-products/thermal-analysis/simultaneous-thermogravimetry-differential-scanning-calorimetry-differential-thermal-analysis/labsys-evo-sta/> (cited 7.1.2019)
- [5] C. Real, M.D. Alcalá and J. Criado, "Development of a new equipment for applying the constant rate thermal analysis (CRTA) to temperature programmed oxidation (TPO) of catalysis," Journal of Thermal Analysis and Calorimetry, 38, 797-802, 1992
- [6] M.J. Diáñez, L.A. Pérez-Maqueda and J.M. Criado, "Direct use of the mass output of a thermobalance for controlling the reaction rate of solid-state reactions," Rev. Scient. Instr., 75, 2620-2624, 2004
- [7] UTIA authors, "Algorithms for adaptive microprocessor control of technological processes," in Czech "Algoritmy pro adaptivní mikroprocesorovou regulaci technologických procesů," report, UTIA CSAV, 1982
- [8] UTIA authors, "Adaptive PID controllers with monolithic microcomputers," in Czech "Adaptivní PID regulátory s monolitickými mikropočítači," report, UTIA CSAV, 1990