### DESIGN OF PRECISE AND LONG-TERM ACCURATE TEMPERATURE REGULATION USING FEATURES OF A LOW-POWER MICROCONTROLLER

Marjan Jenko

### Laboratory for Digital Systems and Electrical Engineering, Faculty of Mechanical Engineering, University of Ljubljana, Ljubljana, Slovenia

Key words: ageing effects, component-built software, fuzzy logic, long-term measurement accuracy, low-power microcontroller, pasteurized soft-boiled eggs, precise temperature measurement, precise temperature regulation, ratiometric measurement

**Abstract:** Low-power microcontrollers were primarily developed for applications with weak energy sources such as batteries and solar cells. The contribution of this paper is the design of precise and long-term accurate temperature regulation with the help of ratiometric temperature measurement, fuzzy logic and features of a low-power microcontroller. Low-power mode is used for measurement to minimize noise. Based on fuzzy logic, a code-optimized software component for precise temperature regulation was developed. The resulting regulated temperature is within +-0.25 °C of the required value at the reference spot of a water-filled volume of ten liters. Temperature regulation is designed to remain accurate for a period of fifteen years, which is the lifetime of the apparatus in the case study.

The two components – for precise and long-term accurate temperature measurement, and for precise temperature regulation – are self-contained. As such, they can be used in any application with stringent requirements for precision and long-term accuracy.

### Zasnova precizne in dolgotrajno točne temperaturne regulacije z uporabo lastnosti mikrokontrolerja za majhno porabo moči

Kjučne besede: staranje vezij, komponentna gradnja programske opreme, mehka logika, trajna točnost meritve, mikrokontroler za majhno porabo energije, pasterizirana mehko kuhana jajca, precizna meritev temperature, precizna regulacija temperature, razmerna meritev

**Izvleček:** Mikrokontrolerje, narejene za izrazito majhno porabo energije, običajno uporabljamo v napravah z baterijskim ali šibkim napajanjem (sončna energija, gibanje ipd.) V prispevku je glavno opravilo mikrokontrolerja z nizko porabo energije precizno krmiljenje termičnega procesa. Stanje nizke porabe z izklopom večjega dela mikrokontrolerja uporabljamo za maksimiziranje razmerja signal/šum pri izvajanju precizne temperaturne meritve. Za samo temperaturno regulacijo pa je na osnovi mehke logike razvit kodno optimiran algoritem preciznega krmiljenja temperature. Na referenčnem mestu je dosežena regulacija +-0.25 °C za volumen desetih litrov. Zahtevano delovanje prikazane naprave je petnajst let, brez vmesnih kalibracij.

Razvita elektronska sestavna dela: precizna in dolgotrajno točna temperaturna meritev, in precizen temperaturni regulator sta samostojni komponenti, uporabni tudi za krmiljenje drugih temperaturnih procesov z zahtevami po preciznosti in dolgotrajni točnosti.

### 1 Introduction

The foundation for the contribution of this paper is the development of an autonomous thermal process for industrial production of soft-boiled eggs, which is a novelty on the market. Requirements in industrial food preparation are a superset of domestic requirements. Besides taste and appearance, the food needs to systematically adhere to microbiological constraints – it must be pasteurized or potential bacteria need to be destroyed by some other means. The industrial food preparation process must be time-invariant, and traceability needs to be built in according to HACCP (Hazard Analysis Critical Control Points) directives.

Pasteurization and cooking of soft-boiled eggs have contradictory requirements. For the former, salmonellae, if present in the center of the egg's mass, need to be definitively destroyed. For the latter, the yolk is to remain soft, i.e. coagulation must not take place. Experiments, along with thermal simulations and microbiological tests, led to a patented thermal profile /1, 2/, where both pasteurization and cooking of soft-boiled eggs in fact coexist. The profile is shown in Figure 1.



#### Fig. 1: Required temperature profile

The required precision of temperature regulation at a reference spot is to be within +- 0.25 °C for the 15-year lifetime of the apparatus. When regulating within +-0.25 °C, measurement imprecision must be at most within +-0.1 °C. It is commonly understood that the professional kitchen appliances market does not tolerate much servicing during the lifetime of a product. Calibration of the temperature measurement circuit every some years of operation is thus out of the question.

The problem of implementation of the required temperature function divides into precise measurement and precise regulation. Both undertakings are developed as two self-sustained components. As such, they are ready for integration into other applications governing functionalities of other embedded systems with similar functional requirements.

### 2 Technical constraints



#### Fig. 2: Scheme of internals of the apparatus

The scheme of the internals of the egg cooker is shown in Figure 2. It is intentionally designed as the simplest solution for the required functionality: a heater heats the water and the eggs immersed in the inner tank; a hot water pump circulates the water to minimize the temperature gradient. A cold water tank encircles the hot water tank. The temperature of the hot water is regulated by two mechanisms, one being heating with an electric heater and the other cooling by mixing hot and cold water. An electromagnetic three-way valve controls water circulation or mixing when needed for cooling the hot water. The cold water is cooled down in a radiator with the help of a cold-water pump and a fan. Volumes of the cold water tank and radiator, and the fan power are designed for the cooling needs of consecutive iterations of the thermal process in Figure 1. The precision of the required temperature regulation depends on temperature measurement at the reference spot and on the temperature regulator, which supplies power for heating and cooling.

## 2.1. Component built software for the required functionality

The primary functionality of the embedded system is temperature regulation and measurement. Other tasks are interaction with the user, governing proper water levels in both tanks, cooling the water in the outer tank, pumping water out of the system when cleaning the apparatus and communication with the diagnostic program on a personal computer – for assembly tests, initial calibration and field diagnostics. Embedded software is structured into functional components by the concept developed in /3/. The structure of the component-built embedded software is shown in Figure 3.



## Fig. 3: Structuring the application into functional components

The functional components are self-sufficient and reusable. As such, they can be individually tested and verified.

The component approach to embedded software design minimizes, to the greatest extent possible, the probability of introducing functional errors into the application.

### 3 Component for precise and longterm accurate temperature measurement

Measured analog variables, such as temperature, are usually converted to voltage, which is sampled by an analogto-digital converter. Long-term measurement accuracy can be achieved by periodic calibration or by systematic elimination of those components that are most sensitive to ageinduced drift, or by both – design and periodic calibration. Implementations of ratiometric measurement have the potential to be more robust to time-induced drifts than implementations of absolute measurements /4/. The required precision for temperature measurement is within +-0.1 °C for the lifetime of the apparatus, which is declared as fifteen years.

Implementation of the scheme of ratiometric temperature measurement, by /4/, is shown in Figure 4.



Fig. 4: Scheme for a ratiometric temperature measurement, by /4/

R(T) in Figure 4 is a platinum temperature sensor, in which resistance is a function of temperature. What results is measurement of the resistance. This is measured by comparing capacitor discharge times through a reference resistor and through a temperature-dependent resistor, by (1)/4/.

$$R(T) = \frac{t_{R(T)} - t_{Rref1}}{t_{Rref2} - t_{Rref1}} (R_{ref2} - R_{ref1}) + R_{ref1}$$
(1)

where  $t_{Rx}$  in (1) is the time to discharge the capacitor C1 from  $V_{co}$  to  $V_c$  via the resistor Rx. Temperature measurement by equation (1) is not sensitive to values of voltages  $V_{co}$ ,  $V_c$ , capacitor C, offset voltage of comparator C1 and resistances of analog switches S1 to S4 in Figure 4.

R1 is used to charge capacitor C1 to  $V_{co}$ . The capacitor is discharged to  $V_c$  via Rref1, (T) and RRef2.

# 4 Component for precise temperature regulation

First, on/off regulation was evaluated, not as a design option but rather to gain insight into temperature delays. The existence of time delays is a significant problem for control of any process. In this application, time delays between heating and cooling differ significantly, since different amounts of power and different mechanisms are used for heating and cooling. As expected for on/off temperature regulation, large temperature swings (up to 6 °C) occurred, and thus simple on/off regulation cannot meet the requirements.

PID regulation gives excellent results for control of systems with symmetrical response (heating, cooling), however the egg cooker has unsymmetrical responses. The physical nature of the mechanisms for heating and cooling is different. The course of cooling is dependent on the temperature difference between the hot and cold water, which is changing in the process. The available quantity of power, when heating up and when cooling down, and the time delays of both processes are different. Our understanding is that thorough mathematical modeling of this cooking process is a complex activity.

After some days of manually experimenting with controlling the thermal process, we were able to describe how to successfully control it. We were generally too slow in reacting to temperature readouts in the constant temperature region, where precision is needed. The results of manually experimenting with regulation gave us the confidence to proceed with fuzzy logic.



*Fig. 5: Membership functions for inputs and outputs of the temperature regulator* 

Fuzzy-type regulation has previously been used in food processing operations less sensitive to precision: peanut roasting /5/, microfiltration /6/, frying /7/ and baking /8/. Fuzzy controllers are conceptually simple. They consist of a) an input stage, b) a processing stage and c) an output stage. They mimic the human approach to regulation in a sense: if the process variable is somewhat off the value, some small corrective action must be taken. For regulation effectiveness, many rules are considered at once, and the terms "somewhat", "small" and "similar" are defined in an exact manner, i.e. in mathematical language. This is done by upgrading variables to probability functions, named membership functions. Figure 5 shows the relevant input and output membership functions for the temperature regulation.

The input stage of the fuzzy controller maps measured temperature to the input membership functions of the Temperature Error (TE, difference between measured and required temperature) and of the Temperature Error Gradient (TEG). The advantage of having more membership functions that overlap at many input values (TE and TEG) is that more membership functions can influence the regulation at the same time. This gives more experimental space when developing and fine-tuning the temperature control.

The processing stage is defined by the behavioral rules in Table 1.

The output stage converts the combined result into an output value of specific process control. Figure 6 shows the required temperature function /1, 2/ and results of fuzzy temperature control after fine-tuning iterations.

Table 1: Behavioral rules of	of the processing stage
------------------------------	-------------------------

Rule number	Temperature	Temperature derivative	Heater power	Valve opening
1	very low		full	closed
2	low		modest	closed
3	normal low	very low	full	closed
4	normal low	low	modest	closed
5	normal low	neutral	low	closed
6	normal low	high	minimum	closed
7	normal low	very high	zero	closed
8	normal high	very low	low	closed
9	normal high	low	minimum	closed
10	normal high	neutral	zero	closed
11	normal high	high	zero	closed
12	normal high	vory high	zero	partially
	normai nign	very mgn		open
13	high		zero	partially
	ingn			open
14	very high		zero	open



Fig. 6: Required and achieved curves of temperature versus time

### 5 Implementation of precise and long-term accurate temperature measurement, fuzzy-type precise temperature regulation and other required functionality

In developing a control algorithm, it is crucial that the finished algorithm meets the requirements and that it be developed in a reasonable time. The algorithm has to be developed on a powerful workstation supplemented by accurate sensors and controls for actuators. The necessity of a well-loaded workstation for development is advocated for two reasons: one is the richness of available Rapid Application Development (RAD) and mathematical tools with excellent Graphical User Interfaces (GUIs) that are available for use on a workstation. The other reason is that the target platform (microcontroller and peripherals) is not available at the outset of the project. In the project, a combination of MATLAB, Simulink, LabVIEW and measurement and actuating hardware was used to develop temperature regulation based on fuzzy logic. The schematic of the setup is shown in Figure 7.



Fig. 7: Schematic of the setup for developing the algorithm for precise temperature regulation

As the algorithm is developed and requirements for other, simpler functional components become known, one chooses the microcontroller and designs the peripherals. Software development systems for microcontrollers have less functionality then RAD tools for developing programs for workstations. This is the reason to do as much development work as possible on a workstation, and not on the target platform. The microcontroller of choice must have the needed periphery integrated – for improved reliability, lower assembly cost and smaller size.

## 5.1. Implementation of temperature measurement

 $V_c$  at the comparator, Figure 4, is to be set within the steep part of the capacitor discharge curve to result in sharp transitions on the comparator's output. Voltages  $V_{co}$ ,  $V_c$  and comparator power are to be filtered by block capacitors to keep the signal-to-noise ratio of the measurement high.

Most of required circuitry and components should be integrated to minimize assembly and to improve reliability. Instrumentation microcontrollers are built for such a purpose. Instrumentation implies a low-power design of the microcontroller for two reasons: one is the potential for field measurement where an electrical grid is not always available, and the other is the capability of switching off some or most functional blocks when performing measurements, i.e. when microcontroller noise must be minimized in order to maximize measurement accuracy.

Switching off the different functional blocks of a microcontroller is performed via different sleep modes. Returning to higher performance is achieved by different triggers, in our case by an interrupt from the comparator, as the capacitor discharges until  $V_c$ .

Block scheme of the selected low-power microcontroller is in Figure 8.



Legende: \* Bus convertor 16/8, \*\* Comparator, \*\*\* USART

### Fig. 8: Block scheme of the selected low-power microcontroller

When measuring, only the comparator and the timer 1, set as a free-running counter, are active. As the comparator changes output, the free-running counter is stopped. The resulting number times the period of the counter clock equals the capacitor discharge time. Figure 9 is illustrative of the measurement inaccuracy, by measuring in the lowpower state of the microprocessor, by equation (1) and by the circuit in Figure 4.



Fig. 9: Inaccuracy of temperature measurement

Each measurement takes about twenty-five milliseconds. Measurements can be smoothed out, e.g. by calculating a running average or average only when memory locations are scarce. It is only that the measurement responsiveness needs to stay high enough to fulfill the required dynamics of the algorithm for temperature control.

# 5.2. Implementation of the fuzzy logic temperature regulator

A low-power microcontroller was selected because of temperature measurement constraints. Development tools for such a microcontroller have fewer capabilities and less perfect user interaction than RAD tools, the implication being that the algorithm needs to be simple and well thought out in advance, before coding. Such an approach minimizes the need for debugging and fixing. The latter has devastating effects on maintenance costs, sales and reputation if performed after product release.

The bases for implementation of fuzzy control of the thermal process are the membership functions in Figure 5 and behavioral rules in Table 1. The procedure, which results in two real numbers between zero and one that define power to be applied for heating and cooling, is the following:

Each point in the plane defined by temperature error TE and temperature error gradient TEG is checked for applicability of each input membership function, according to Figure 5. For each point in the plane (TE, TEG), i.e. for each applicable combination of membership functions in m(TE) and m(TEG), appropriate output membership functions in m(heater H) and m(valve V) (Figure 5) are selected by the rules in Table 1. Probabilities of the selected output functions in m(H) and m(V) are calculated from the probabilities of the corresponding input functions in m(TE) and m(TEG). Since there can be more applicable combinations of input functions in m(TE) and m(TEG) for each point in the plane (TE, TEG), there can be more relevant rules in Table 1, which select different output functions in m(H) and m(V) for the particular point in the plane (TE, TEG). In the particular case being elaborated, it so happens that five is the maximum number of rules that contribute to the final solution at some points in the (TE, TEG) plane. At other points, fewer rules, but not less than one, select the output membership functions in m(H) and m(V).

The weighted average of the selected output membership functions in m(H) and m(V) defines heating and cooling power for each point in the plane (TE, TEG). The results of weighted averaging are two surfaces, both defined over the plane (TE, TEG). The surface that represents heating power is shown in Figure 10, while the surface that represents the three-way valve opening for cold water is shown in Figure 11.



Fig. 10: Surface of heating power



## Fig. 11: Surface of cold water opening at the three-way valve

The corresponding programming, from the plane (TE, TEG) to the surfaces in Figures 10 and 11, using the membership functions in Figure 5 and rules in Table 1, written in the RAD environment for development in C++, is available for download and examination from http://www2.arnes.si/ ~mjenko9/EggCooker.

The two surfaces that define heating and cooling power for precise temperature regulation are simple to fill in memory locations or to be described by constants and linear functions in simple *if ... then* or *switch ... case* statements, which is the case in the implementation of the egg cooker. This is an important implementation simplification. All of the manipulation with the probability functions is coded and executed only once on a well-equipped workstation in a suite of modern RAD tools that permits ease of debugging and visualization. The only result of this, which is easily reproduced with simple C decision statements and realtype variables, is actually coded into the target system.

#### 5.3. Implementation of other required functionality into the embedded system

Implementation of other required functionalities is structured into other functional components. These encapsulate finite state machines that define their functionality. For example, control of blinking LEDs and periodic buzzing (warning signals) is encapsulated in hardware-dependent components.

USART, in Figure 8, is governed by a hardware-dependent component for serial communication, which is needed for computer-supported tests and calibration in production and for field servicing. The twelve-bit analog-to-digital converter, which is an integral component of the microcontroller, is not used. For about ninety percent of the operating time, the microcontroller is in sleep mode, during which temperature measurement takes place. All other procedures, including temperature regulation, are performed in the time between measurements of capacitor discharge time. Figure 12 illustrates the timing of microcontroller operation. The signal in Figure 12 is voltage on the capacitor C1 in Figure 4, i.e. voltage on the comparator's input.



Vertical scale: 2V/div, Horizontal scale: 5ms/div White background: regulation and other functionality, active mode Gray background: measurement, sleep mode

#### Fig. 12: Timing of microcontroller operation

Since the implementation of temperature regulation is optimized on the system level, the microcontroller's active time needed for one iteration of all required procedures never exceeds ten percent of the time needed for one temperature measurement. As such, temperature measurement is a continuous and consistent process.

### 6 Case study

Figure 13 shows the apparatus for industrial production of pasteurized soft-boiled eggs, in which precise temperature regulation is needed. Figure 14 shows the custom electronics board.

Two details are extremely important in the design of the electronics board in Figure 14: in the low-power circuitry, ground and power planes are as complete as possible, and the microcontroller's IOs are equipped with filters to prevent potential functional glitches that could result from voltage transients.



Fig. 13: Apparatus



Fig. 14: Electronics board in the apparatus

On the right in Figure 14 is the temperature measurement circuit, consisting of two precision resistors, capacitor and integrated circuit with analog switches. Circuitry with grid voltage, consisting mostly of thyristor switches, corresponding drivers and protection from high voltage surges is on the left. The board is enclosed in a metal box for shielding and protection from potential water spills.

### 7 Discussion

One would not choose a low-power microcontroller to control heating with approximately three kilowatts of available electric power at first thought. After some elaboration, however, one finds that the low-power low-noise modes and rich set of built-in peripherals define a low-power microcontroller as a useful platform for precise temperature measurement and control.

All the electronics circuitry, for the grid and for low voltage, is on the same Printed Circuit Board (PCB) in the case study for reasons of engineering optimization. Less wiring results in higher reliability. Such an approach requires careful design of ground and power planes to minimize inductance /9/. Effects of a less-than-perfect layout would be stochastic errors in the operation of the microcontroller induced by voltage transients. In the hypothetical case of somewhat sloppy coding, it would be close to impossible to know which functionality error would have an electrical and which would have a programming cause.

A professional approach to software design, to layout, to filtering potential causes of noise, and having a low-power microcontroller and grid components on the same PCB results in a robust system that passes the EMC test of superimposed four-kilovolt bursts to the grid voltage with excellent results.

Low-power electronic components are being designed and used primarily to lengthen operation time between consecutive battery chargings. Improvements in low-power consumption are achieved by developments in two directions: one, the ongoing shrinking of circuit dimensions and the other in circuit design. The most common approaches are throttling the system clock, lowering supply voltage when the processing load is low and turning off different functional blocks when they would otherwise be idling. In the present application, low-power mode is used to minimize measurement noise. As a result, the precision of the temperature readout is improved.

In the design of the embedded software, a component approach was used. The application was first structured into functional components. As self-sufficient entities, these were individually designed, coded and verified. Then, the application was built from already verified functional components. Such an approach results in verifications on the component level and on the application level. This approach on the system level eliminates the need for software fixing.

When developing complex components such as the present precise temperature regulator, it is most beneficial to use a rich and productive environment of RAD and visualization tools to the greatest extent. In the case study, the environment of MATLAB, Simulink, LabVIEW and hard-

ware extensions for measurement and activation were used to define membership functions and rules. Algorithm design, refinement and verification were performed using this rich set of tools. The RAD suite was used to perform the coding related to probability functions and fuzzy rules. This was performed in the C++ language with the most convenient means for encapsulation and overall structuring. Only the most essential code – definition of section-wise flat surfaces – is implemented in the target system.

The precondition for precise temperature regulation is precise temperature measurement. Different approaches were studied. The presented ratiometric measurement, using the low-power mode of the microcontroller, yields precision and long-term accuracy. In an environment with more noise one could filter the measurement results, however filtering slows down the responsiveness of the measurement, which has a negative influence on precise temperature regulation.

### 8 Conclusion

A component approach to the development of an embedded system is presented. The design requirement is regulation precision of +-0.25 °C in the constant regions of the required temperature profile. As a precondition, temperature must be measured with a precision of +-0.1 °C for the lifetime of the product, which implies the need for longterm accuracy.

The contribution of this paper is the development of two self-sufficient functional components: one performs precise and long-term accurate temperature measurement; the other is a precise temperature regulator.

Precise and long-term accurate temperature measurement is performed by a ratiometric method and with utilization of one of the microcontroller's low-power modes. Since continuous measurement is needed and the system needs to be responsive at the same time, coding is optimized for efficiency and, because of hardware constraints, for a small footprint.

The development of a precise fuzzy-type temperature regulator is structured into a) iterative work on the definition of membership functions and rules in the environment of MATLAB, Simulink, LabVIEW and dedicated hardware; b) coding in C++ on a workstation that starts with membership functions and rules, and results in surfaces that govern heating and cooling controls; and c) the use of surfaces for real-time regulation in the target system, which is programmed in ANSI C.

Both components, for precise and long-term accurate temperature measurement, and for precise temperature regulation are self-sufficient. As such, they can be reconfigured and reused in other applications with similarly stringent requirements for precision of measurement and regulation, and long-term measurement accuracy.

The overall embedded system application was first structured into functional components. These were designed, coded and verified as self-sufficient entities. Then, the application was built from the components. This approach has proved to generate reliable embedded software products /3/.

The case study involved the development of a new type of industrial kitchen appliance that simultaneously pasteurizes and cooks soft-boiled eggs. This is a novelty on the industrial food preparation appliance market.

### 9 References

- /1/ Otto Koch. Vorrichtung zur Zubereitung von Huhnereieren. Europaische Patentschrift EP 0 920 273 B1, July 1997.
- /2/ Otto Koch. Method and device for preparing chicken eggs. US Patent, No.: 6,162,478, December 2000
- /3/ M. Jenko, N. Medjeral, P. Butala, Component-based software as a framework for concurrent design of programs and platforms - an industrial kitchen appliance embedded system. Microprocessors and Microsystems, 2001, vol. 25, no. 6, pages 287-296.
- /4/ M. Jenko, Ratiometric measurement for long-term precision, reasoning and case study, submitted to Informacije MIDEM, october 2009
- /5/ V. J. Davidson, R. B. Brown, and J. J. Landman. Fuzzy control system for peanut roasting. Journal of Food Engineering, 41(3– 4):141–146, August 1999.
- /6/ N. Perrot, L.M 'e, G. Trystram, J.-M. Trichard, and M. Decloux. Optimal control of the microfiltration of sugar product using a controller combining fuzzy and genetic approaches. Fuzzy Sets and Systems, 94(3):309–322, March 1998.
- /7/ Ryszard Rywotycki. Food frying process control system. Journal of Food Engineering, 59(4):339–342, October 2003.
- /8/ N. Perrot, G. Trystram, D. Le Guennec, and F. Guely. Sensor Fusion for Real Time Quality Evaluation of Biscuit during Baking. Comparison between Bayesian and Fuzzy Approaches. Journal of Food Engineering, 29(3–4):301–305, August–September 1996.
- /9/ H. Ott, Noise reduction techniques in electronic systems, 2nd ed., Wiley, 1988, ISBN 0-471-85068-3, pages 274-298.

Marjan Jenko Laboratory for Digital Systems and Electrical Engineering, Faculty of Mechanical Engineering, University of Ljubljana, 1000 Ljubljana, SI

Prispelo (Arrived): 02.11.2009 Sprejeto (Accepted): 09.09.2010