ELECTRONICS AND ELECTRICAL ENGINEERING

ISSN 1392 – 1215 *2007. No. 7(79)*

ELEKTRONIKA IR ELEKTROTECHNIKA

AUTOMATION, ROBOTICS

T125

AUTOMATIZAVIMAS, ROBOTECHNIKA

Testing of Data Processing Software in Heat Metering Systems

V. Knyva, M. Knyva

Department of Electronics and Measurements Systems, Kaunas University of Technology,
Studentu str. 50, LT-51368, Kaunas, Lithuania, phone: +370 37 300535; e-mail: vytautas.knyva@ktu.lt

Introduction

Nowadays, measurement system parameters and characteristics basically depend on microprocessors and their software. Hence, verification of measurement systems software (MSS) is very important. After adoption of the Measurement Instruments Directive (MID), verification of MSS will be necessary [1]. The MID started in the early nineties, was approved in spring of 2004, and after transition period it became fully functional in autumn of 2006. The MID introduces a "new approach" to the measurement system software and its verification. Due to MID birth, in 1997 WELMEC (Western European Legal Metrology Cooperation) formed the work group WELMEC-SOFTWARE. The main object of this group was extended formulation of essential MDS requirements. Some time later, PTB (Physikalich-Technische Bundesanstalt) formed another work group MID-SOFTWARE, which elaborated and concretized WELMEC requirements. On the basis of MID and documents published by these groups, three stages of MDS verification were proposed in [2]. Testing procedure of data processing software in heat metering systems is a part of a software functionality investigation stage and it will be analyzed in this paper.

Heat Metering System

For development of a modern metering system modular architecture often is used, for example sensors of heat conveying liquid, flow sensors are separated from a measurement system control module (calculator). Block scheme of such a system presented in Fig.1. The system can be controlled using user or communication interfaces, i.e. a user may carry out configuration of the system. Control software periodically reads data from sensors and addresses it to the data processing software. After calculations, measurement results are written to the memory of devices. A user can access instantaneous measurement results and an archive of measurements through a user interface. Besides, measurement results can be sent to the central city's workstation via a communication interface.

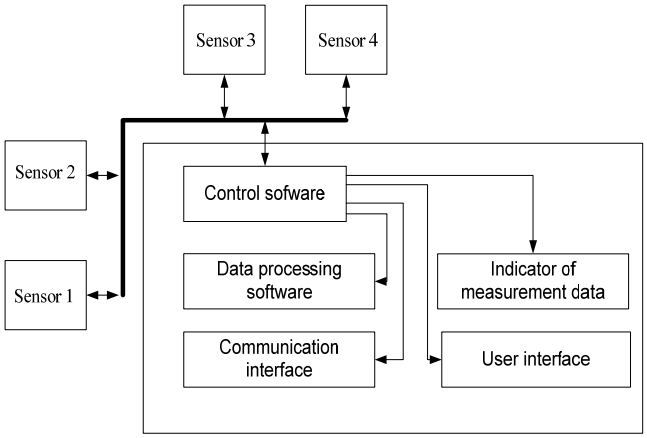

Fig. 1. Block scheme of the measurement system

The simplest heat metering system is shown in Fig. 2.

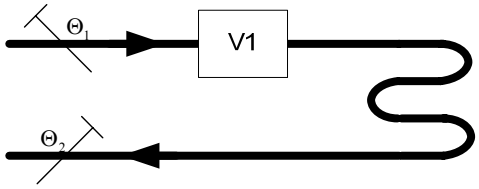

Fig. 2. Closed heating system. Flow sensor in a flow line

Here, we have 3 sensors: sensor of feeding flow V1, two temperature sensors in flow (Θ_1) and return (Θ_2) lines. Quantity of the heat given up can be calculated:

$$Q = \int_{V_0}^{V_1} k\Delta\Theta \ \mathrm{d}V, \ \mathrm{kWh} \qquad (1)$$

where Q is quantity of the heat given up, $\Delta\Theta = \Theta_1 - \Theta_2$ temperature difference between flow and return of the heat exchange circuit. V - volume of the liquid passed, k - heat coefficient calculated using the following expression

$$k(p,\Theta_1,\Theta_2) = \frac{1}{\nu}\frac{\Delta h}{\Delta\Theta} \qquad (2)$$

where $\Delta h = h_1 - h_2$ is the specific enthalpy difference between the flow and return enthalpies, ν - specific liquid mass, p - pressure of the liquid.

For validation of measurement systems software, a "black box" method can be used [3]. I.e. data processing software can be validated simulating the signals of heat meter external sensors [4]. In Fig. 3, block scheme of such validation is presented.

According to the boundary value analysis, a testing sequence generator builds a test sequence [5]. In parallel with a mathematical model, which can be called reference software, data processing software gets test data. The maximum permissible error can be calculated when evaluation of the results received from the model and data processing software is completed [6].

If the calculated maximum permissible error is satisfactory, measurement systems data processing software can be treated as metrologically reliable.

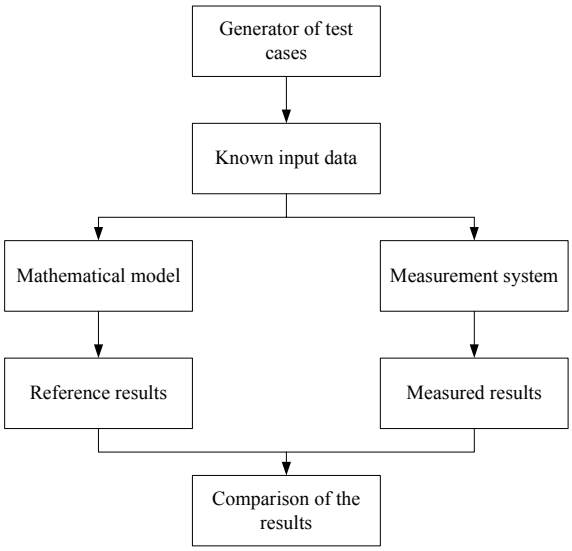

Fig. 3. Verification block scheme of data processing software

Development of Test Sequences

Any program can be considered as function $F(x_1,x_2,...,x_n)$ with the results depending on input data. Further, a standard case will be discussed. Function F has two variables x_1,x_2 with the following boundaries: $a \le x_1 \le b$ and $c \le x_2 \le d$. The input space of function F is shown in Fig. 4. Any point within the rectangle is legitimate input to function F.

Heat metering data processing software operating by the scheme presented in fig. 1 may serve as an example of a data processing function with two variables. Here, variables represent temperatures Θ_1 and Θ_2 of flow and return heat conveying liquid, whereas flow of heat conveying liquid is a constant value.

Boundary value analysis focuses on the limits of input variables to identify the test cases. It is logical that errors in software tend to occur near the extreme values of an input variable. So, each test sequence can be developed from values – nominal (*nom*), minimal (*min*) or maximal (*max*) and values near the bound (*min*+, *max*-) with an assumption that failures only rarely are the result of the simultaneous occurrence of two errors. Thus, the boundary value test sequence is developed by holding one variable at its nominal value and in pair selecting another variable with an extreme value. It is proposed that variables of *min*+ and *max*- have 5% higher or lower value of *min* or *max*.

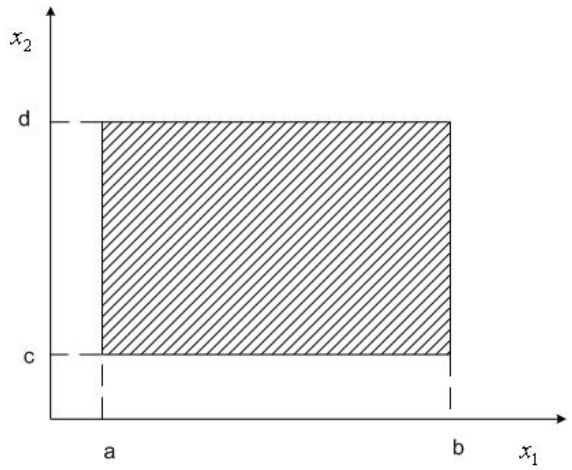

Fig. 4. The input space of function $F(x_1,x_2)$

Temperature measurements flow and return liquids of a heat metering system should be analyzed. In each case, temperature can vary within 0÷160ºC limits. Nominal values for flow liquid temperature are Θ_1=80ºC, and for return liquid temperature they are Θ_2=40ºC [7]. The boundary value analysis test case for function F of two variables (temperatures) can be written as shown: (80,0); (80,40); (80,152); (80,157); (3,40); (8,40); (152,40); (160,40).

Boundary value analysis can be extended by "faulty" values in test cases [8], i.e. *max*+ and *min*- values can be introduced. Then a test sequence will be extended by (168,40); (-8,80); (80, -8); (80, 164) values. Test cases with "faulty" values can be used for verification of a software protection algorithm against faults.

Boundary value analysis is based on a "single fault" assumption, but in the real world usually much more than one fault is observed. So, for full data processing software verification the worst case testing method can be used. Worst case test cases for the function of two variables is shown in Fig. 5.

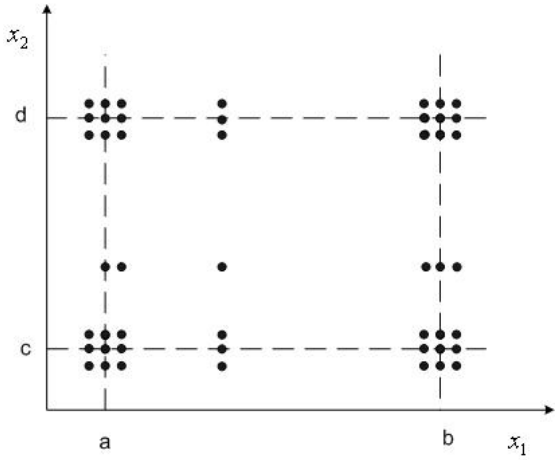

Fig. 5. Worst case test cases for the function of two variables

Using this method 5^n testing cases can be developed, whereas boundary value analysis methods allows developing only $4n+1$ testing cases, here n – number of variables. A test sequence developed using the worst case method with two variables will look as follows: (160,0); (152,0); (160,8); (152,8); (0,0); (8,0); (0,8); (8,8); (160,160); (160,152); (152, 160); (152, 152); (0, 160); (0, 52); (8,160); (8,152).
Worst case test cases can be extended by adding "faulty" values: (160, -8); (168,0); (168,168); (152,168); (168,152); (0,168); (-8,168);(-8,-8); (8,-8).

Testing of Temperature Measurement Software

An experiment was made in order to verify metrological reliability of temperature measurement software. Verification was carried out in the following order:

- Resistor bridges R1, R2 simulate temperature sensors;
- Heat measuring system measures Θ_{1m}, Θ_{2m}, $\Delta\Theta_m$ temperatures;
- Comparison between the calculated Θ_1, Θ_2, $\Delta\Theta$ and measured Θ_{1m}, Θ_{2m}, $\Delta\Theta_m$ temperatures was made.

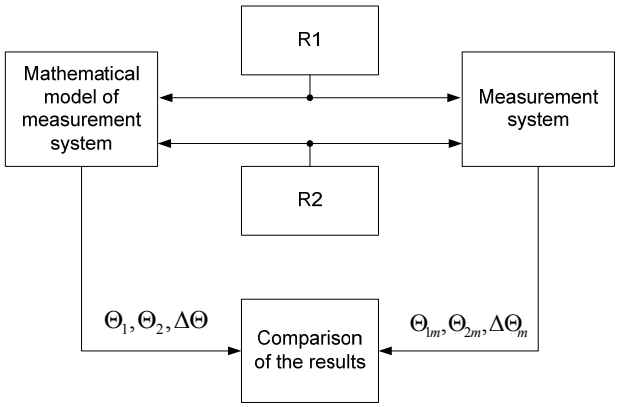

Fig. 6. Experimental stand

Table 1. Experimental results with boundary value analysis

No.	Θ_1, °C	Θ_2, °C	$\Delta\Theta$ °C	Θ_{1m}, °C	Θ_{2m}, °C	$\Delta\Theta_m$ °C	f_{in}.	f_{ms}
1	80	0	80	79.91	0	79.91	Y	Y
2	80	8	72	79.92	7.59	72.33	N	N
3	80	40	40	79.86	39.52	40.34	N	N
4	80	152	-72	79.9	155.49	-75.59	Y	Y
5	80	157	-77	79.91	156.56	-76.65	Y	Y
6	3	40	-37	2.35	39.55	-37.2	Y	Y
7	8	40	-32	7.76	39.49	-31.73	Y	Y
8	152	40	112	152.07	39.5	112.57	N	N
9	160	40	120	160.8	39.5	121.3	N	N
10	168	40	128	168.21	39.5	128.71	Y	N
11	-8	80	-88	0	79.65	-79.65	Y	Y
12	80	-8	88	79.9	0	79.9	Y	Y
13	80	164	-84	79.97	163.6	-83.63	Y	Y

Results are presented in Table 1. Measurements 1-9 were made using boundary value analysis test cases. Measurements 10-13 were made using "faulty" test cases. Each time then heat measuring systems software receives a "faulty" signal from temperature sensors, it must generate a warning on fault.

Here, f_{in}= Y shows that "faulty" data was sent to measurement systems software, whereas f_{in}= N demonstrates that correct data was sent to measurement systems software. f_{ms}= Y shows that measurement systems software detected "faulty" data, and f_{ms}= N indicates that software failed.

Experimental results illustrated that measurement systems software detected negative difference between temperatures. At 10 measurements, software detected no faulty value of flow liquid temperature. For detailed verification, the worst case testing case was used. Experimental results are presented in Table 2. Here, only the test cases where measurement systems software failed are provided.

Table 2. Experimental results with worst case testing

No.	Θ_1, °C	Θ_2, °C	$\Delta\Theta$ °C	Θ_{1m}, °C	Θ_{2m}, °C	$\Delta\Theta_m$ °C
1	160	0	160	159.9	0	159.9
2	152	0	152	151.93	0	151.93
6	8	0	8	7.47	0	7.47
17	168	-8	176	167.9	0	167.9
18	160	-8	168	159.9	0	159.9
19	152	-8	160	151.9	0	151.9
21	168	160	8	167.9	159.8	8.1
27	160	-8	168	159.7	0	159.7
28	168	0	168	167.8	0	167.8
31	168	152	16	167.8	151.7	16.1
35	8	-8	16	7.61	0	7.61

As it was expected, heat metering systems software detected fault only at negative temperature differences and minimal temperature values. But it failed with exceeded temperature values and minimal or even negative return liquid temperature values. Such performances of measurement systems software contradict with the essential measurement systems requirements presented in the measurement instruments directive. Besides, it can be stated that measurement results obtained with the measurement system using such software can be falsified or incorrect, i.e. metrologically unreliable.

Conclusions

1. Testing methodology for heat metering systems software was developed. An experimental stand and mathematical model were built.
2. Metrological reliability of temperature metering software in heat metering systems was estimated. It has been designated that measurement systems software detects faults only at negative temperature differences and near the minimum temperature bound. But software failed when simulated temperatures reached or exceeded the upper bound of permissible temperature.
3. Temperature measurement software failed at negative and minimum temperatures of return liquid. Such

performances of measurement systems software contradict with the essential measurement systems requirements presented in the measurement instruments directive. Besides, it can be stated that measurement results obtained with the measurement system using such software can be falsified or incorrect, i.e. metrologically unreliable.

References

1. Directive 2004/22/ec of the **European Parliament** and of the Council of 31 March 2004 on Measuring Instruments // Official Journal of the European Union. – 2004. Accessed at: www.europa.eu.int.
2. Software Requirements on the Basis of the Measuring Instruments Directive // **WELMEC** guide 7.1. – 1999. Accessed at: www.welmec.org.
3. **Jorgensen P. C.** Software Testing. – CRC Press. – 2002. – 315 p.
4. **Čitavičius A., Knyva V., Knyva M.** Verification of User Interface of Supply to the Customer by Mains Measuring Devices // WSEAS Transaction on Systems. – Athens: WSEAS Press, 2006. – Vol. 5, No. 10. – P. 2450–2455.
5. **Cox M. G., Harris P. M.** Design and use of reference data sets for testing scientific software // Analytica Chimica Acta. – 1999. – Vol. 380, No. 2. – P. 339–351.
6. **Čitavičius A., Knyva V., Knyva M**. Problems of Heat Meters Software Verification // WSEAS Transactions on Systems. – Athens: WSEAS Press, 2007. – Vol. 6, No. 5. – P. 1004–1008.
7. **Čitavičius A., Knyva V., Knyva M.** Investigation of Heat Meters Software Functionality // Digest of Conference on precision electromagnetic measurements (CPEM 2006), Torino, Italy. – 2006. – P. 418–420.
8. **Pomeranz I., Redy S.** Worst-Case and Average-Case Analysis of Test Sets. Proceedings of Design // Automation and Test in Europe Conference. – 2005. – P. 115–119.

Submitted for publication 2007 04 06

V. Knyva, M. Knyva. Testing of Data Processing Software in Heat Metering Systems // Electronics and Electrical Engineering. – Kaunas: Technologija, 2007. – No. 7(79). – P. 11–14.

Nowadays measurement system parameters and characteristics basically depend on microprocessors and their software. Hence, verification of measurement systems software (MSS) is very important. After adoption of the Measurement Instruments Directive (MID), verification of MSS will be necessary. For investigation of heat metering systems metrological reliability testing methodology was created. Experimental stand and mathematical model were build. Experimental testing results of temperature measurement software are presented. Ill. 6, bibl. 8 (in English; summaries in English, Russian and Lithuanian).

В. Книва, М. Книва. Тестирование программного обеспечения обработки данных в системе измерения тепловой энергии // Электроника и электротехника. – Каунас: Технология, 2007. – № 7(79). – С. 11–14.

Большинство функций в современных системах измерения тепловой энергии реализованны программным путем. Поэтому особое внимание необходимо выделять надежности программного обеспечения. Также необходимо отметить, что система измерения тепловой энергии является объектом правовой метрологии, поверка программного обеспечения которого должна проводится в обязательном порядке по директиве ЕС об измерительных приборах. Для оценки надежности программного обеспечения систем измерения тепловой энергии предложена методика тестирования программного обеспечения обработки данных измерения. Применен принцип поверки методом “черного ящика“. Предложен алгоритм формирования последовательностей тестирования. Представлены результаты тестирования программы измерения температур. Ил. 6, библ. 8 (на английском языке; рефераты на английском, русском и литовском яз.).

V. Knyva, M. Knyva. Šilumos matavimo sistemų duomenų apdorojimo programinės įrangos testavimas // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2007. – Nr. 7(79). – P. 11–14.

Dauguma šiuolaikinių šilumos matavimo sistemų funkcijų paprastai atliekamos programiniu būdu. Todėl vis svarbiau yra užtikrinti programinės įrangos patikimumą. Be to, šilumos matavimo sistema yra teisinės metrologijos objektas, kuriam programinės įrangos patikra yra privaloma pagal ES Matavimo prietaisų direktyvą. Siekiant įvertinti šilumos matavimo sistemų programinės įrangos patikimumą, pasiūlyta duomenų apdorojimo programinės įrangos testavimo metodika. Pasirinktas „juodosios dėžės“ tikrinimo principas. Pasiūlytas testavimo sekų sudarymo algoritmas. Pateikti temperatūros matavimo programinės įrangos testavimo rezultatai. Il. 6, bibl. 8 (anglų kalba; santraukos anglų, rusų ir lietuvių k.)