ELECTRONICS AND ELECTRICAL ENGINEERING
ISSN 1392 – 1215 2010. No. 4(100)
ELEKTRONIKA IR ELEKTROTECHNIKA

AUTOMATION, ROBOTICS
T 125
AUTOMATIZAVIMAS, ROBOTECHNIKA

Algorithm for Estimation of Measurement Devices Software's Metrological Reliability

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, parameters, characteristics and functionality of measurement devices basically depend on microprocessors software. Software used in measurement devices overtakes functions from a hardware part of the device. For example, data processing algorithms, auto diagnostic control functions are implemented using software. Basically during verification of the device, just a hardware part of the device is verified. But devices metrological reliability depends on implemented auto diagnostic algorithm. In this article algorithm on how to estimate metrological reliability of measurements devices software will be presented.

Faults of measurement devices software

Each measurement device or system can be considered as resistant to faults if the software protection algorithms can identify and correct critical or minor faults. Otherwise, the results received by measurement devices software without protection algorithms can be faulty. It can be considered that the measurement device tolerates faults if software is able to finish measurement successfully after faults have been detected [1].

All disorders of measurement devices are called by known factors, i.e. faults in a software code, simplified protection algorithm or others. In literature, the following sorts of faults are mentioned:

- Coding faults;
- Faults of measurements converters;
- Inadmissible influences of a user;
- Hardware faults.

Software of measurement devices that measures the same object is different by means of its complexity. Complex software clearly can have considerably more faults compared to simple one. Therefore, measurement devices must be divided into groups by means of software complexity and functionality thereof. The groups can be described in the following way:

- Measurement devices with software data processing;
- Measurement devices with software control of measurement converters and data processing;
- Measurement systems.

Software faults influence metrological characteristics of the measurement device. The following characteristics may be distinguished:

- Characteristics of the measurement result – measurement converters faults, stability of the measurement device, fault detection, etc.;
- Characteristics of measurement precision – instrumental, random, method faults;
- Dynamical characteristics of the measurement device.

Heat metering systems auto diagnostic algorithm

For illustration how software faults and simplified protection algorithms effects measurement results we will use heat meters calculator. Basic configuration of the calculator presented at Fig. 1. Here we have:

- two lines (flow and return) of heat conveying liquid,
- heating devices connected on the end of lines,
- temperature sensors connected to each line,
- flow sensor connected to the flow line,
- All measurement sensors connected to the heat meters calculator.

Measured data received from temperature sensors of flow and return lines, flow sensors data after digitalization passes auto diagnostic algorithm. After this data is processed using data processing algorithm and then measurement results are indicated on the indicating device[2].

The quantity of the heat given up implemented in data processing algorithm is calculated applying the expressions presented in the OIML recommendations [3] i.e.:

$$Q_{ref} = k\Delta\Theta V , \tag{1}$$

here Q_{ref} – reference quantity of the heat given up, $\Delta\Theta = \Theta_{flow} - \Theta_{ref}$ temperature difference between the flow and return of the heat exchange circuit; V – volume of the passed liquid; k – heat coefficient calculated using the following expression:

$$k(p, \Theta_{flow}, \Theta_{return}) = \frac{1}{\nu}\frac{\Delta h}{\Delta\Theta}, \qquad (2)$$

here $\Delta h = h_{flow} - h_{ref}$ – specific enthalpy difference between the flow and return enthalpies, ν – specific liquid mass, p – pressure of the liquid.

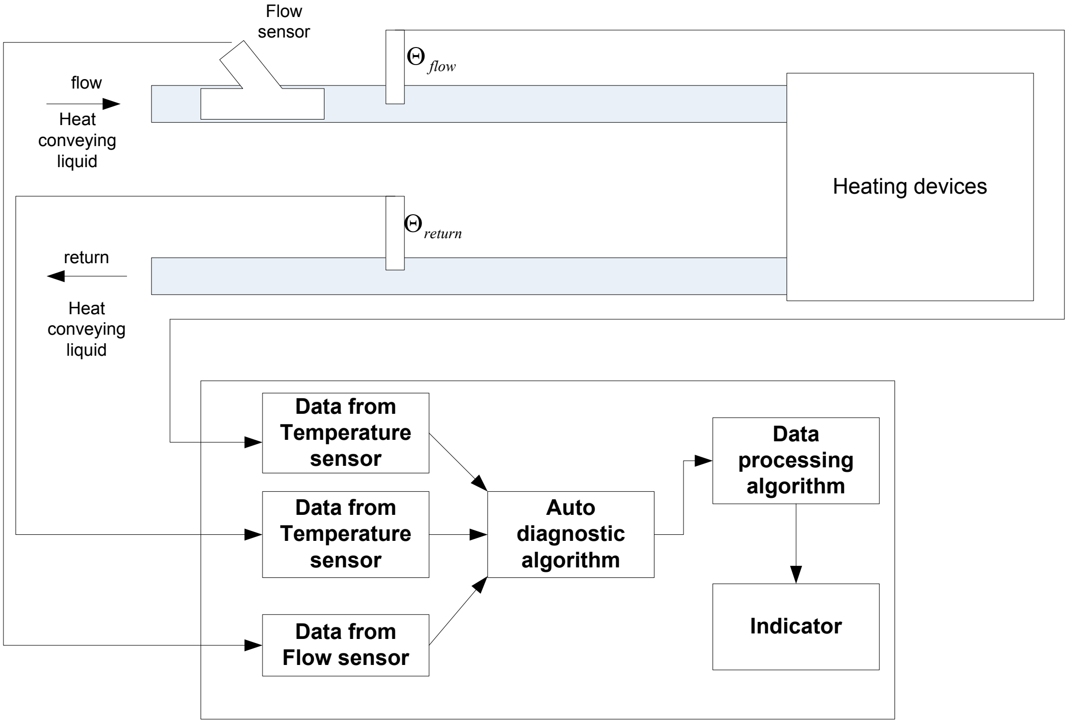

Fig. 1. Heat metering system

At Fig. 2 presented typical auto diagnostic algorithm of the heat metering system (fig. 1) Here we have 3 main branches or as it can be said 3 main checks:

1. Is temperature sensor at flow line connected?
2. Is temperature sensor at return line connected?
3. Is flow sensor connected?

If one of the sensors isn't connected then heat meters auto diagnostic software informs user about error in heat metering chain. It must be stated that at this point if error is detected heat metering processes must be stopped until removal of the problem. If all sensors are presented when auto diagnostic software checks measured values:

1. Measured temperature at flow line Θ_f must be $\Theta_{\min} < \Theta_f < \Theta_{\max}$;
2. Measured temperature at return line Θ_r must be $\Theta_{\min} < \Theta_r < \Theta_{\max}$;
3. Measured flow q must be $q_{\min} < q < q_{\max}$.

At this stage auto diagnostic system checks ranges of the measured values. If temperatures are too low or too high auto diagnostic software reports about error in measuring chain. The last check is made after some data processing – in recommendations for heat meters is stated that difference between temperatures of the flow and return lines must be positive value. At this point if difference is negative autodiagnostic software reports about error.

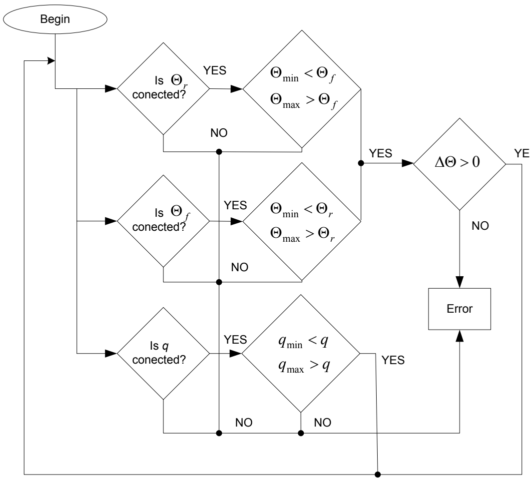

Fig. 2. Auto diagnostic algorithm of heat metering system

So this is "ideal" version of auto diagnostic software's functionality. But experimental results showed that sometimes few faults are left in auto diagnostic algorithm. For example our previous experiments [4] showed that, heat metering systems software detected fault only at negative temperature differences and minimum temperature values. But it failed with exceeded temperature values and minimum or even negative return liquid temperature values. Or other our experiments [5,6] showed that in some cases, especially when parameters of the signals (pulse amplitude and duty cycle) received from sensors reach marginal but still permissible values, the value of heat given up is calculated by the heat meter calculator wrongly. 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. metrologicaly unreliable.

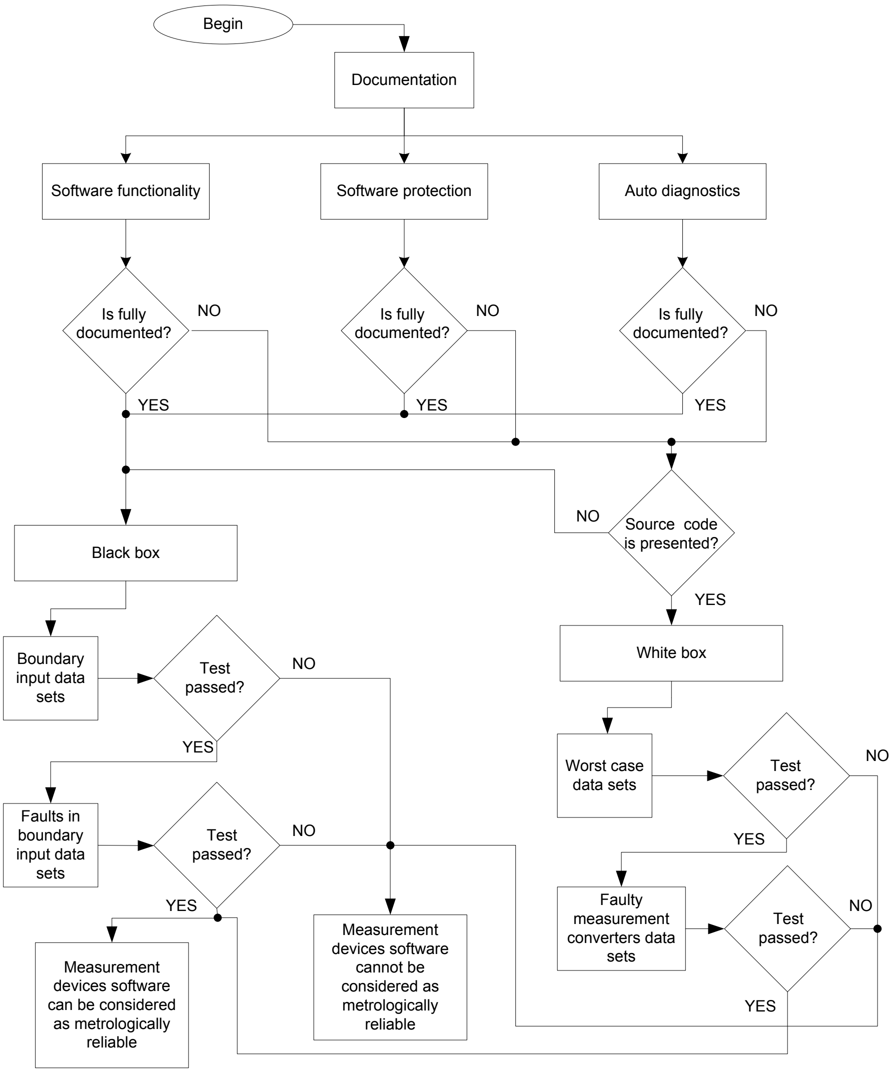

Fig. 3 Algorithm for estimation of measurement devices software's metrological reliability

As result of our previous works at Fig. 3 algorithm for estimation of measurement devices software's metrological reliability is presented. Algorithm is divided into two stages: analysis of measurement instruments documentation and analysis of measurement instruments functionality when source code of the software is unknown ("black box" method) and when source code is known ("white box" method). Measurement devices documentation must be analyzed by three aspects - how software functions, how software is protected, how auto diagnostic algorithm works. If all aspects of software are documented well when "black" box method for functional software analysis can be used.

For this analysis, specific test sequences required:

- Boundary input data sets. For investigation of measurement devices software functionality when measurements are performed using boundary values of measurement converters. For heat meters example, measurement of the minimum or maximum temperature;
- Faults in boundary data sets show how software protection algorithms respond to single faults. For heat meters example, negative temperature of heat conveying liquid .

If one of the aspects mentioned above aren't documented at all or documented purely when "white" box method for measurement devices software analysis must be used. For this analysis, specific test sequences required:

- Worst case boundary data sets demonstrate how software protection algorithms react to regular logical faults in measurement data.
- Faulty measurement converters signal data sets illustrate how software protection algorithms react to illegal signals received from measurement converters.

Experiments were made with "faulty" conveying liquid temperature values. Nominal values of flow and return liquids are 80 and 40 ^{o}C, whereas a standard temperature measurement range of heat meters is 0-160 ^{o}C. Few experimental results are presented in Table 1 and Fig. 4. Analysis of the experiment results highlighted that in the cases of particular combinations of temperature values (but still permissible values) of conveying liquids the quantity of heat given up is calculated wrongly, and the main thing is that the software gives no report about the faults.

Table 1. Experimental results

Meas no.	Θ_{flow}, ^oC	Θ_{return}, ^oC	Q_{et}, MWh	Q_{meas}, MWh	δ_Q, %
1	160	0	0,38	0,34	-10,52
2	152	0	0,75	0,67	-10,67
3	8	0	0,042	0,04	-4,76
4	168	-8	0,62	0,57	-8,06
5	160	-8	1,16	1,12	-3,44
6	152	-8	1,56	1,59	1,92
7	168	0	2,11	2,04	-3,31

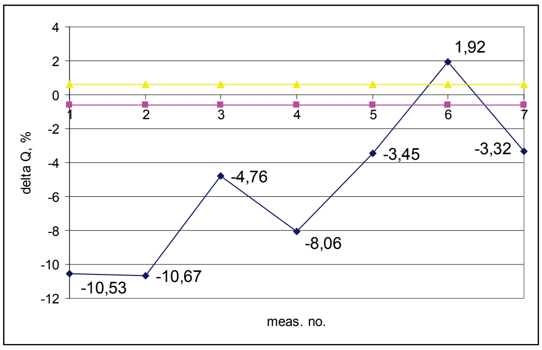

Fig. 4. Experimental results

Measurement devices software which passes algorithm without failure can be stated as metrologicaly reliable. If software fails the manufacturer must be informed and device can not be used for legal metrology purposes.

Conclusions

1. Algorithm for estimation of measurement devices software's metrological reliability presented. As example heat metering system analyzed.
2. Metrological reliability of heat metering systems auto diagnostic algorithm analyzed. It has been designated that measurement systems auto diagnostic software sometimes fails, i.e. detects faults only at some cases.
3. Proposed algorithm for estimation of measurement devices software's metrological reliability can help to discover faults in software which leads to incorrect measurement result, i.e. it can help to prove that measurement systems software is metrologicaly unreliable.

References

1. Software Requirements on the Basis of the Measuring Instruments Directive // WELMEC guide 7.1, 1999. [www.welmec.org].
2. **Šípal J.** Adaptive Modeling of District Heating Network // Electronics and Electrical Engineering. – Kaunas: Technologija, 2009. – No. 8(96). – P. 65–68.
3. Heat meters // OIML–R75–1, 2, 2002 (edition). [www.oiml.org].
4. **Knyva V., Knyva M**. Testing of Data Processing Software in Heat Metering Systems // Electronics and Electrical Engineering. – Kaunas: Technologija, 2007. – No. 7(79). – P. 11–14.
5. **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.
6. **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.

Received 2010 02 16

V. Knyva, M. Knyva. Algorithm for Estimation of Measurement Devices Software's Metrological Reliability // Electronics and Electrical Engineering. – Kaunas: Technologija, 2010. – No. 4(100). – P. 47–50.

Nowadays, parameters, characteristics and functionality of measurement devices basically depend on microprocessors software. Software used in measurement devices overtakes functions from a hardware part of the device. For example, data processing algorithms, auto diagnostic control functions are implemented using software. Basically during verification of the device, just a hardware part of the device is verified. But devices metrological reliability depends on implemented auto diagnostic algorithm. In this article algorithm on how to estimate metrological reliability of measurements devices software is presented. Few experimental testing results of heat meters autodiagnostic algorithm software presented. Ill. 4, bibl. 6, tabl. 1 (in English; abstracts in English, Russian and Lithuanian).

В. Книва, М. Книва. Алгоритм для оценки программного обеспечения измерительных устройств // Электроника и электротехника. – Каунас: Технология, 2010. – № 4(100). – С. 47–50.

Большинство функций в современных системах измерения тепловой энергии реализованы программным путем. Поэтому надежности программного обеспечения необходимо выделять особое внимание. Для оценки надежности программного обеспечения систем измерения тепловой энергии предложен алгоритм тестирования программного обеспечения обработки данных измерения. Представлены результаты тестирования программы измерения тепловой энергии. Ил. 4, библ. 6, табл. 1 (на английском языке; рефераты на английском, русском и литовском яз.).

V. Knyva, M. Knyva. Algoritmas matavimo priemonių programinės įrangos metrologiniam patikimumui įvertinti // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2010. – Nr. 4(100). – P. 47–50.

Daugumą šiuolaikinių matavimo priemonių funkcijų valdo mikroprocesorius ir jo programinė įranga. Akivaizdu, kad programinė įranga turi įtakos, o daugeliu atvejų netgi lemia priemonės funkcinę gebą ir metrologines charakteristikas. Tačiau matavimo priemonių metrologinio įvertinimo metu faktiškai yra tikrinama tik aparatinė dalis. Peržiūrėjus OIML rekomendacijas ar prietaisų patikros dokumentus, matyti, kad programinės įrangos tikrinimo ir metrologinio laidavimo praktika dar nenusistovėjusi. Todėl šiame darbe pasiūlytas matavimo priemonių programinės įrangos metrologinio laidavimo algoritmas. Pateikti eksperimentinių tyrimų, atliktų su šilumos skaitiklio skaičiuotuvu, rezultatai. Il. 4, bibl. 6, lent. 1 (anglų kalba; santraukos anglų, rusų ir lietuvių k.).