A Reliability Analysis of State Estimation Software Based on SCADA and WAMS

Irina Kolosok, Elena Korkina*, and Alexander Tikhonov

Melentiev Energy Systems Institute Siberian Branch of the Russian Academy of Sciences, Irkutsk, Russia

***Abstract* — The procedure of state estimation of electric power system (EPS) remains relevant for control of electric power facilities. Traditionally, the quality of state estimation results depends on the quality of input information, i.e. telemeterings and synchronized phasor measurements; proper design of equivalent circuit, specification of its parameters; and characteristics of communication channel. However, the quality of the applied SE software is normally not analyzed during its operation, while it is necessary to periodically check and assess whether or not it operates properly in real conditions. Moreover, timely updating of applied algorithms is also very important to increase the reliability of software operation. The paper proposes an analytical approach to the assessment of SE software operability based on the fault tree technology. The focus is made on various aspects of improvement in reliability of power system SE software during its operation.**

***Index Terms* — State Estimation, Fault Tree, SCADA, WAMS, Reliability of SE software.**

I. INTRODUCTION

There are universal indicators including those based on the queuing theory for software reliability that can reveal the level of its working capacity in terms of functionality and delivery of result. Application packages, different libraries of standard algorithms, and operating systems belong to this software type. SE software is specialized, and it is intended to provide quality results (based on a deep analysis of the input data) affecting the control of a technological process. It is extremely important to know how the SE software behaves under real conditions. Therefore, it is crucial to develop such indicators for software fault tolerance that would enable the protection of weak components of program blocks, and, thereby, increase the SE software operability in any hostile environment.

Being a complex property, reliability in general includes fault-free operation, durability, maintainability, and storability [1]. The software reliability level features the probability of its fault-free operation over a certain time interval. In [2], the authors propose the following reliability indicators: the average number of correctly solved problems over a certain time interval Δt1, the mean number of errors for that interval, the probability of solving a set number of problems for the time interval Δt2, the probability of emergence of a set number of errors for that interval, etc. Such indicators may be used for any software.

In [2], to analyze the software reliability the authors enumerate the factors leading to software faults: errors in the program, use of non-optimal and imperfect algorithms (for example, heuristics use), restricted real-time operation (the system state changes faster, than the computing cycle lasts). Interaction of several factors and hardware problems in the computing system may also lead to software faults.

For the software designed for on-line control of EPS to be as less vulnerable to various failures as possible, it should be based on:

- Accurate mathematical models of computational schemes and equipment applied in them;
- Highly redundant measurement system, provided by backup measurement devices;
- Robust algorithms for initial data verification (a priori, a posteriori, robust);
- Repeated testing of software, firstly on simulation and then on real-world data.

Failure of software operating in control system can lead to serious consequences for the process of control.

Knowing well the SE software object domain, we will try to develop other reliability indicators that account for the specific features of the solved problems and feature the fault tolerance of this very software.

In this paper, we propose a fault tree to analyze the consequences of technical failures and faults in the data acquisition and data processing systems (SCADA and WAMS) and in the SE software. Based on the analysis of the measurement information and on the SE results, we

* Corresponding author.
E-mail: korkina@isem.irk.ru

http://dx.doi.org/10.25729/esr.2018.01.0012
Received February 27, 2018. Revised April 15, 2018. Accepted April 21, 2018. Available online: April 25, 2018.

determine the reliability level for SE software, and propose some measures to increase its reliability

II. Electric Power System State Estimation Software And Computing Environment

A. EPS data processing by State Estimation software

The effectiveness of the EPS control is largely determined by reliability and quality of data on state variables of EPS.This information, coming to EPS Control centers, represents telemeasurements, telesignals, PMU measurements, pseudo measurements (zero injections at transit nodes, nodal loads from dispatcher records, etc.), a priori data on the accuracy of measurements (variances) and parameters of the EPS network. The available information is normally insufficient to control the state of the entire system. Random errors in the initial data as well as bad data can lead to wrong control decisions. This explains the need to solve the SE problem within the software for on-line control of optimal conditions of the Unified Power System of Russia.

An analysis of EPS observability [3] in state estimation determines if it is possible to estimate the power system state based on a set of available measurements. Considerable corruption in the SE results can be avoided by including a Bad Data Detection (BDD) algorithm in the SE software. EPS SE is performed for an equivalent circuit of network, and, normally, in real time. Errors in telesignals can lead to a wrong topology of the computational scheme.

SE software calculates steady state, using the measurements for the current computational scheme of the EPS. The obtained steady state is used as a reference state for solving various on-line control problems. Therefore, the obtained estimates of the state variables should lie within a feasibility region, i.e. meet the equality and inequality constraints. The estimates going beyond the feasibility region generate the need to consider constraints directly during performance of each individual control function, which can cause a considerable delay in decision making. Therefore, the program intended for the consideration of inequality and equality constraints specified for both measured and unmeasured state variables should be included in the real-time SE software [4, 5, 6].

The EPS SE software is intended to obtain the EPS current state model from telemetery and telesignals arriving from SCADA and phasor measurements from WAMS.

For the SE software to operate with WAMS, a traditional algorithm of linear SE (LSE) is used on the basis of state vector in rectangular coordinates [7]. The linear state estimation is also successfully performed using the test equation (TE) method developed at the Melentiev Energy Systems Institute of Siberian Branch of the Russian Academy of Sciences [4, 8]. This method has a number of advantages over the traditional non-linear approach, the main of them being a possibility of a-priori bad data detection.

B. Supervisory Control and Data Acquisition (SCADA) fault tolerance

SCADA system includes: remote telemetry units (RTUs) installed at EPS substations to take telesignals on the switching equipment state and measurements of the state parameters, communication channels, database (DB), systems of on-line display of the state parameters, as well as the software (EMS-application) to process the measurement and to form control commands for dispatching management objects. Figure 1 presents the structure of a SCADA installed at the control center of a regional network company.

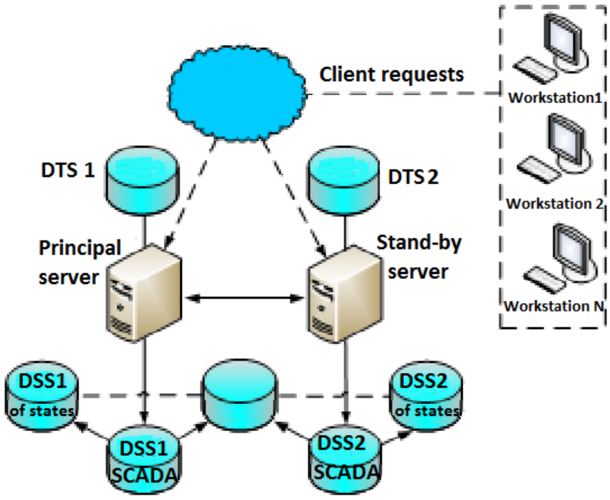

Figure 1. SCADA structure.

The SCADA operates at two independent servers, Principal (1) and Standby (2). The servers constantly exchange requests. For each server, their own data transmission system (DTS) and data storage system (DSS) are provided. Herewith, DSS 1 and DSS 2 constantly synchronize their data. Every 30 minutes, the data are transmitted to the historic server, at which reservation is provided continuously.

An absence of response from Principal Server 1 is considered as a fault, and the system switches to Standby Server 2. This occurs at minimal delays not noticeable for the user.

There can be two types of failures: 1) hardware failures, and 2) software failures. After a failure at the Principal Server, the software is restarted (in case, it was a program failure), or it is switched to a Standby Server. Client requests go to the Principal Server, and, in case of its failure, they are redirected to the Standby Server with no request latency change.

In case of the DSS1 failure, the Principal Server switches to DSS2 with minimal delays. In case of the DTS1 failure, the incoming data stream is directed to DTS2. As practice shows, failures seldom arise in such systems. Software failures occur once a month, on average, hardware failures occur once a year.

C. WAMS Data Acquisition (DA) Automatic System (AS) fault tolerance [9]

In 2009-2011, to solve the problems in acquiring and storing the WAMS information, a WAMS Data Acquisition Automatic System (WAMS DA AS) shown in Fig.2 was created and put into industrial operation.

PMU, being the lowest hardware level transmits phasor measurements (PMs) to the system under Protocol C37.118-2008/2011 to phasor data concentrators (PDCs) for further use in calculations. PMs are relayed to a higher level of dispatching control to the super-PDC corresponding to the control hierarchy. This architecture is simple, reliable, and perfectly suitable to solve problems in the absence of restrictions for the computing and telecommunication infrastructure. The measurements are kept in the DB of own design [9]. The system servers are connected in a cluster operating synchronously: they interact with each other and exchange the information with data sources and clients.

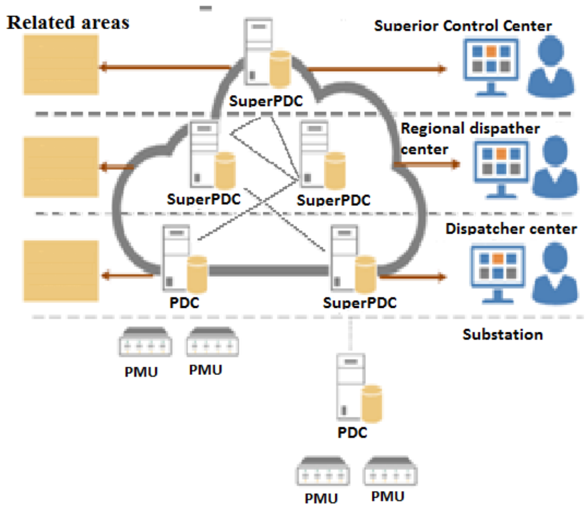

Figure 2. Structure of WAMS Data Acquisition Automatic System [9].

The DBs included in the cluster are synchronized among themselves. The facilities ensuring fault tolerance within the system allow us to create clusters of 2 and more servers. The storage is scaled over almost unlimited number of servers (up to 65535). Failure of any two servers will not lead to information loss.

On the one hand, the DA AS architecture is hierarchical, because the servers are at all the levels of the dispatcher control; on the other hand, the cloudy technology is used, when the places of data storage and their traffic routes are not anchored rigidly. This increases the DA AS fault tolerance.

III. Block Diagram Of The SE Software For The Reliability Analysis

SE software will operate correctly, as long as all three elements are operable: measurements, the network, and SE algorithms:

- analyze the network observability;
- identify and detect gross errors in PM and telemetry (TM);
- filter random TM errors, i.e., to receive their estimates and to finally calculate non-measured parameters.

Let us represent the SE software, a program tool, as a technical system that is to operate safely and properly. For the initial analysis, we make a block diagram (Fig.3). As the Figure demonstrates, most of the SE software components are reserved.

Measurements: when there are no PMs, the SCADA measurements are used; if there are no both PM frame and SCADA snapshot, SE software can operate with archival snapshot. Moreover, the incoming measurements are recorded in the real-time DB (RT DB) that is also reserved;

Network: the data on the diagram are stored on the DB server of constant information; at computer centres of major power facilities, a standby server and a standby DB are provided; a special algorithm forms the current (operational) network based on the basic scheme from telesignals;

Algorithms: algorithms for the observability analysis (OA), for the measurement validation, and for the SE are reserved by alternative blocks.The SE software produces the fastest solution, when the LSE algorithm functions. Therefore, the OA program should determine, whether the network is PMU-observable. If the OA answer is negative and LSE is impossible to run, OA is performed by a set of SCADA measurements (here is the example of the OA algorithm redundancy) for non-linear SE (here is the example of the SE redundancy). Redundancy for the algorithm of the a-priori validation by test equations is represented by a-posteriori validation by SE remainders. When analyzing the reliability of complex systems, however, it is necessary to find which of the elements are critical and whose serious faults affect the system operability to a greater extent, in general. Typical criticality indicators [2] are the fault probability, the severity of consequences, the element tolerance to malicious activities, the risk value due to a fault, the possibility of fault localization, the controllability of the element state during the operation, reserving, etc. Ranging the elements by the criticality degree is possible at different levels of structuring the system objects. Critical elements may be visually provided by the Fault Tree technology

IV. Fault Tree Technology

For the first time, the term "fault tree" in Russian literature was mentioned in Yu. Guk's book [10]. Known since 1960s, the Fault Tree Analysis technology applied by expert systems in military aviation, then in nuclear power, and in some other industries [11, 12], appeared a convenient means to analyze the operation capacity (fault tolerance) of any technical system or its separate complex nodes. The fault tree is presented in the form of a hierarchical structure:

- Level 1 - tree root - is the addressed technical system;

- Level 2 is the system indicators featuring this system;
- Level 3 - system elements - is the details of system indicators;
- Level 4 - tree leaves - is the events leading to a fault of the system operability (technological problems);
- Level 5 – the lowest level is the measures to suppress the fault causes.

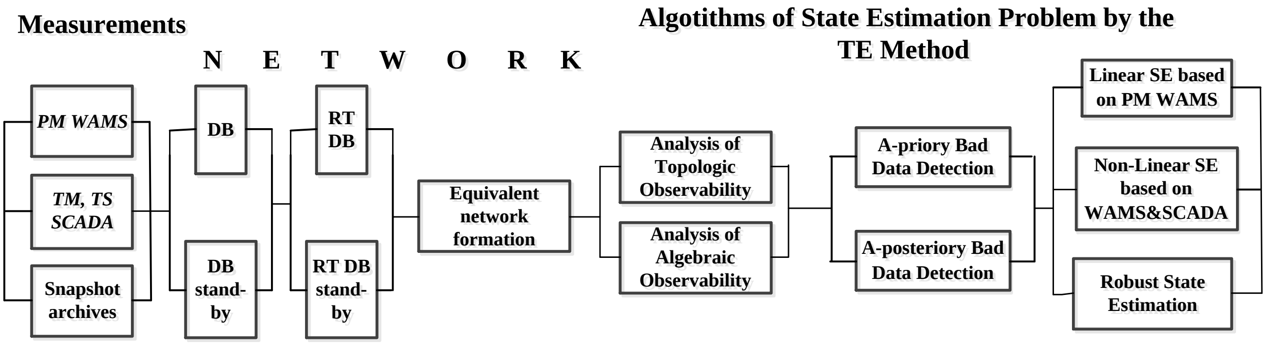

Figure 3. State Estimation Software block diagram.

Figure 4 presents the SE software fault tree to analyze the reliability of its operation. At the top level, there is SE software itself. The system indicators (measurements, network, algorithms) is the second level. Those indicators contain basic elements (measurements types, databases, procedures etc.). Basic elements are exposed to these or those failures or technical faults which are on the forth level of Fault Tree. At the tree lowest level, there are counter-measures written in italics. The set of counter-measures enables the calculation of the indicators for the EPS SE algorithm operation efficiency and fault tolerance.

Figure 4 shows the block "Analyzing program", the program determining the most vulnerable SE software components in terms of fault tolerance. This determination is based on the statistic block that stores the calculated indicators for a certain period of time (both blocks are given in bold).

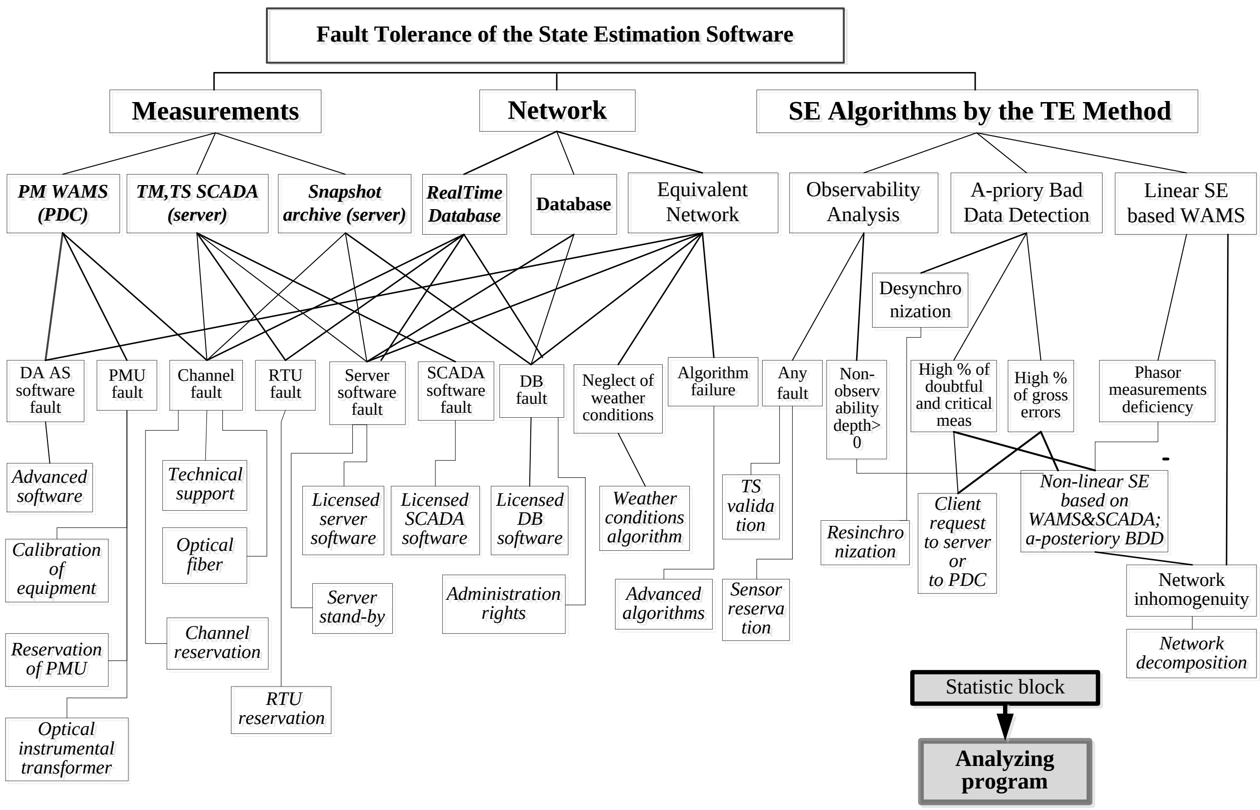

Figure 4. Electric Power System State Estimation Software Fault Tree.

Unlike the known term "decision tree" [13], the fault tree suggests that the total of possible options to solve a problem decreases up to the number of options obviously threatening the operability of the addressed technical system. In fact, the fault tree is a diagnostic method offering a way out of a specific problem situation."

V. Fault Probabilities For The Fault Tree elements

The Analyzing program is run several times per day after a certain time, in which N runs of the SE software are executed. Due to series connections of basic components ("Measurements" - "Network" - "Algorithms", see Fig.3) the probability for the SE software fault tolerance is calculated as follows [14]:

$$P_{SE\_soft} = P_{meas} \times P_{network} \times P_{\mathrm{alg}}, \tag{1}$$

in any other case, the SE software is non-operative, i.e.,

$$\mathrm{Q}_{SE\_soft} = 1 - P_{SE\_soft}. \tag{2}$$

In turn, due to parallel connections of elements, the probability for the basic component "Measurements" fault is calculated as follows

$$Q_{meas} = Q_{PM} \times Q_{\mathrm{TM}} \times Q_{archive}, \tag{3}$$

where Q_{meas} is the fault probability of the measuring part, $Q_{\mathrm{PM}}, Q_{TM}, Q_{archive}$ are the faults of phasor measurements, telemeasurements and pseudomeasurements, respectively; $Q_{meas} = 1$ means that no measurements are available for the SE software.

Equivalent circuit (current network) formation is switching of the network elements, which imposes telesignals on the basic scheme, integrating the adjacent nodes into one node at the switched-on bus-tie switch, and substituting several parallel lines with a single equivalent line. The procedure of equivalent circuit formation can fail only if the algorithms for its construction contain errors. Fault tolerance of some elements can be improved by their reservation [15] (see Fig.3 – DB stand-by and RT DB stand-by):

$$\mathrm{Q}_{network} = 1 - (1 - \mathrm{Q}_{TS})(1 - \mathrm{Q}^2_{\mathrm{DB}})(1 - \mathrm{Q}_{\mathrm{alg}\_forming\_network}) \tag{4}$$

Procedure for the observability analysis (OA) initially checked if the number and structure of measurements in the SCADA snapshots corresponded to the network graph. A special algorithm was used for allocation of SCADA sensors to improve the quality of network observability. This algorithm indicated the sites poorly equipped with sensors and provided recommendations on placement of sensors and measuring channels [3]. At present, along with SCADA measurements, phasor measurements are also used in EPS. Therefore, now, the algorithms for the WAMS sensor allocation are developed considering possible failures of individual connections, sensors, and loss of measurements [16].

Observability analysis enables the detection of the observable and unobservable fragments in the network. It is known that, voltage phasor measurements at a node and current phasor measurements in the adjacent line can be used to obtain a calculated PM value at a node on the other end of this line. Such a sensor location in the network (area), when all the absent nodal measurements may be calculated by the available PMs, is called "non-observability depth equal to 0" [17]. If the available real and calculated PMs do not provide a completely PM-observable network, we consider the network "unobservability depth" equal to 1 and more (which implies measurement insufficiency for the LSE operation) as a failure of the OA procedure for the LSE.

$$Q_{OA} = \frac{n_{\mathrm{stop}\_unobserv=1}}{\mathrm{N}_{\mathrm{run}\_SE\_software}}. \tag{5}$$

If the LSE cannot be performed, the procedure of nonlinear SE is run based on the SCADA and WAMS data.

A failure of the OA procedure for the nonlinear SE is practically impossible due to special options supporting observability in up-to-date nonlinear SE software, $Q_{OA\_for\_nonlinearSE} = 0.0001$. Thus, for LSE or nonlinear SE the failure is:

$$Q_{OA} = Q_{OA\_LSE} \times Q_{\mathrm{alg}\_OA}, \quad \text{or} \quad Q_{OA} = Q_{OA\_SE} \times Q_{\mathrm{alg}\_OA}. \tag{6}$$

The procedure of the a-priori BDD based on the test equation method is performed before the SE procedure. It shows: the number of reliable and non-reliable measurements received by the SE software, the number of critical measurements which when excluded lead to non-observable parameters whose errors cannot be detected; and the groups of doubtful measurements, in which it is impossible to detect the erroneous measurements. The measurements with the detected gross errors (bad data) are replaced with specified ones during the algorithm operation. Thus, their variance values increase, which reduces the trust in such measurements. It is much worse, if the measurements contain critical ones that do not belong to test equations, and doubtful ones, whose quality cannot be checked at a given set of test equations. Therefore, we will consider a high percent of the doubtful and critical measurements in the incoming snapshot as a failure of the a-priori validation procedure:

$$Q_{apriori\_doubt} = \frac{n_{\mathrm{run}\_SE\_with\_high\%\_doubt\_meas}}{\mathrm{N}_{\mathrm{run}\_SE}} \tag{7}$$

$$Q_{apriori\_critical} = \frac{n_{\mathrm{run}\_SE\_with\_high\%\_critical\_meas}}{\mathrm{N}_{\mathrm{run}\_SE}} \tag{8}$$

Herewith, the SE software operation does not stop, because, further, the algorithms for the robust SE and for the a-posteriori BDD are started. Correspondence between the SCADA snapshot time tags and those of WAMS is a strict requirement for the BDD procedure.

The LSE procedure is solved non-iteratively. An indispensable condition for its start is the observability of the entire equivalent network through PMs. Therefore, the LSE algorithm fault is:

$$Q_{LSE} = Q_{OA} \times Q_{a\lg\_OA} \quad (9)$$

In our Fault Tree, the algorithms for nonlinear SE, robust SE and for the a-posteriori BDD are presented as the counter-measures against the SE software operability faults (see Fig.4). Given the series connection of blocks of the system indicator "Algorithms", we obtain

$$Q_{a\lg} = 1 - (1 - Q_{OA})(1 - Q_{BDD})(1 - Q_{SE}) \quad (10)$$

VI. Case Study

We apply the Fault Tree Technology (2)-(10) to calculate the fault tolerance probability value of SE software (1)

a) The block "**Analyzing program"** based on block "**Statistica"** is run 4 times per day, 1 time every 6 hours.
b) SE software is executed 3 times per 2 min (1 time per 40 s or 2/3 min). This is equal to 540 runs every 6 hours c) The initial probability of any algorithm failure is

$$Q_{a\lg\_OA} = Q_{a\lg\_BDD} = Q_{a\lg\_SE} = 0.0001.$$

d) SCADA software failures occur, on average, once a month, hardware (server) failures occur ones a year

$$Q_{SCADA\_software} = \frac{4}{30 \times 24 \times 60} \times \frac{2}{3} = 6.17 \times 10^{-5};$$

$$Q_{server} = \frac{4}{365 \times 24 \times 60} \times \frac{2}{3} = 5.07357 \times 10^{-6};$$

e) The number of SCADA snapshot failures is 3, the number of telesignal failures is 7, so

$$Q_{TM} = 3/540 = 0{,}00555; \quad Q_{TS} = 7/540 = 0{,}01296$$

As a result, **for SCADA** we obtain:

$$Q_{TM\_SCADA} = 1 - (1 - 0.00555)(1 - 5.07357 \times 10^{-6})(1 - 6.17 \times 10^{-5}) = 0.0056$$

$$Q_{TS\_SCADA} = 1 - (1 - 0.01296)(1 - 5.07357 \times 10^{-6})(1 - 6.17 \times 10^{-5}) = 0.0130$$

f) There are 6000 WAMS frames per 2 min (1 per 20 ms), but only 3 frames are available for SE software (for SCADA). Totally 540*6000=3240000 frames come to PDC during 6 hours. We consider averaged phasor values on a small time interval when phasor angle is insignificantly changed. Thus, we have 540 accurate frames because we consider systematical errors as faults of communication channels and assume that

$$Q_{PM\_snapshot} = 0.0$$

g) Research [18] on the data transmission from a measurement point to a PMU device, considering communication channel and instrumental transformer errors result in the fault probability:

$$Q_{PMU} = 0.0879 \ [17]$$

h) Reliable transfer of PMs from PDC to dispatcher center suggests sending 50x60x60 packages per hour, but with the leased WAMS channels some packages can be lost. Minimal losses of packages per 1 channel are estimated at about 500-1000, consequently,

$$Q_{WAMS\_channel} = \frac{500}{50 \times 60 \times 60} = 0.0028$$

i) PDC fault probability depends on fault probabilities of PMU and communication channels:

$$Q_{PDC} = 1 - (1 - 0.0879)(1 - 0.00555) = 0.0904$$

j) Let DA AS faults occur 2 times per month:

$$Q_{DA\_AS\_soft} = \frac{2 \times 4}{30 \times 24 \times 60} = 1.2346 \times 10^{-4}$$

Thus, **for WAMS** we obtain

$$Q_{PM} = 1 - (1 - 0.0)(1 - 0.0904)(1 - 1.2346 \times 10^{-4}) = 0.0905$$

The system indicator "Measurements" (3). Assuming $Q_{BD} = Q_{SCADA\_software}$, we calculate

$$Q_{meas} = 0.0905 \times 5.07 \times 10^{-6} \times 0.0056 = 5.9854 \times 10^{-9}$$

The system indicator "Network" (4) is

$$Q_{network} = 1 - (1 - 0.0130)(1 - 5.07357 \times 10^{-5}) * (1 - 0.0001) = 0.0131$$

k) The fault probability of Observability Analysis Algorithm (6), assuming $Q_{OA\_LSE} = Q_{PM}$ is:

$$Q_{OA\_LSE} = 1 - (1 - 0.0905)(1 - 0.0001) = 0.0906$$

l) The fault probability of a-priori BDD algorithm (7), (8). 15 SCADA snapshots with doubtful measurements and 5 SCADA snapshots with critical TM were detected during 6 hours:

$$Q_{apriori\_doubt} = 15/540 = 0.028;$$

$$Q_{apriori\_critical} = 5/540 = 0.00926;$$

$$Q_{apriori\_BDD} = 1 - (1 - 0.028)(1 - 0.00926)(1 - 0.0001) = 0.0369$$

$$Q_{aposteriori\_BDD} = 1 - (1 - 0.00926)(1 - 0.0001) = 0.0094$$

The system indicator "Algorithms" for LSE (9) is:

$$Q_{ALG\_LSE} = 1 - (1 - 0.0905)(1 - 0.0369)(1 - 0.0001) = 0.1243$$

The system indicator "Algorithms" for nonlinear SE and robust SE (here $Q_{OA\_nonlinearSE} = 0$) is:

$$Q_{ALG\_nonlinearSE} = Q_{ALG\_robustSE} = 1 - (1 - 0.0)(1 - 0.0094)(1 - 0.0001) = 0.0096$$

m) We obtain the fault tolerance of the SE software based on Q_{meas}; $Q_{network}$; Q_{alg} and approach (2):

$$P_{meas} = 1 - 5.9854 \times 10^{-9};\ P_{network} = 1 - 0.0131;$$
$$P_{ALG\_LSE} = 1 - 0.1243;\quad P_{ALG\_nonlinearSE} = 1 - 0.0096;$$

As a result, **the SE software fault tolerance probability** is (1):

$$P_{LSE\_software} = 0.8642;\quad P_{nonlinearSE\_software} = 0.9774$$

The fault probability of the SE software with reserved algorithms (see Fig.3) is:

$$Q_{alg} = 1 - (1 - 0.0906 \times 0.0)(1 - 0.0369 \times 0.0094)(1 - 0.0001^2)$$
$$= 3.5417 \times 10^{-4}$$

Finally, **the SE software fault tolerance probability when the algorithms are reserved** is:

$$P_{SE\_software\_when\_reserved} = 0.9865$$

VII. Conclusion

Development of any program product is determined by an urgent need to accelerate the processing of raw data and obtain calculation results in the form of tables, plots, diagrams, diagnostic and expert evaluation. The program product (software) implementation is based on the selection of adequate processing algorithms and appropriate programming languages (environments). Testing of the program product on reference samples and its application to a real object are the final stages of its design. Acceptance of the program product (software) is performed according to the established state standards [19, 20, etc.].

Further, the program product operation starts and whether or not it is successful depends on the quality of initial data and possibility of its processing by reliable algorithms. In [19], the authors claim, for example, that “the software tools and programs included in the program product cannot reach the state when their control by user is impossible, and the data should be neither corrupted or lost”. In the real-time operation of the state estimation software, however, the cases of data losses (underdelivery) are commonplace. Therefore, for the failure-free operation of the applied program product it is necessary to have adaptive algorithms that can solve a problem depending on the composition and quality of the initial information. Thus, a fault tree analysis, one of the analytical approaches, is proposed in this paper to the personnel working with the state estimation software in real time.

Acknowledgment

This study is supported by the Siberian Branch of the Russian Academy of Sciences (Project III.17.4.2) of the Federal Program of Scientific Research (№ AAAA-A17-117030310438-1).

References

[1] Industrial product dependability. General concepts. Terms and definitions. [Online]. Available: http://docs.cntd.ru/document/gost-27-002-89 (in Russian)

[2] A.M. Polovko, and S.V. Gurov, *Reliability theory grounds*, Saint-Petersburg: BKhV-Petersburg. 2006. 704p., (in Russian).

[3] A.Z. Gamm, I.I. Golub, *Observability of electric power system*, Moskow: Nauka, 1990, 220p, (in Russian).

[4] A.Z. Gamm and I.N. Kolosok, “Test Equations and Their Use for State Estimation of Electrical Power System”, *Power and Electrical Engineering: in Scientific Proc. of Riga Tech. University*. Riga: RTU, 2002, pp. 99-105.

[5] M.V. Khokhlov, “The threshold properties of robust procedures for electrical power systems state estimation”, *Electrichestvo*. 2010, No. 4. pp. 2-12. (in Russian).

[6] P.M. Erohin, V.G. Neuymin, N.G. Shubin, D.M. Maksimenko, “The use of optimization methods interior point for power system state estimation,” *Proceedings of NTC of the Unified Power System*, No. 1 (66), 2012, pp. 39-46. (in Russian).

[7] I.N. Kolosok, E.S. Korkina, E.A. Buchinsky, “Monitoring of EPS Facilities Using PMU-Based Linear State Estimation”, in *Proc. of the 5th Intern. Conf. Liberalization and Modernization of Power Systems: Smart Technologies for Joint Operation of Power Grid*, Irkutsk, Russia, 2012.

[8] I.N.Kolosok, E.S.Korkina, E.A.Buchinsky,“The Test Equation Method for Linear State Estimation Based on PMU Data”, in *Proc. of the 18th Intern.Power System Computation Conference (PSCC-2014)*, Poland, Wroclaw, 2014.

[9] F. Gaidamakin, A. Kislovskiy, “The evolution of WAMS of SO UPS. Technical overview”, in *Proc. of the 5th Intern. Scientific and Technical Conf. Actual trends in development of power system relay protection and automation*, Sochi, 2015. (in Russian).

[10] Yu.B. Guk, *Computing the energy facility reliability*, Leningrad: EnergoAtomIzdat, 1988, 224p. (in Russian).

[11] J. Endrenyi, *Reliability modeling in electric power systems*, New York: John Wiley and Sons, 1978, 338 p.

[12] C.A. Ericson, “Fault tree analysis”, [Online]. Available:http://www.eecs.ucf.edu/~hlugo/cop4331/ericson-fta-tutorial.pdf

[13] J. R. Quinlan, “Simplifying decision trees”, *Intern. Journal of Man-Machine Studies*, 1987, Vol. 27 (3): 221. doi:10.1016/S0020-7373(87)80053-6.

[14] Ye.S. Venttsel, *Teoriya veroyatnostey [Probability theory]*, Moscow: Nauka, Fizmatgiz, 1969. 576 p. (in Russian).

[15] G. Kovalev, “Reliability assessment of interconnection

tie-lines with Intermediate substations”, *Electrichestvo*, No. 4, 2017, pp.14-21. (in Russian).

[16] Ya. Kuzkina, “The observability ensuring of electrical distribution networks with distributed generation through Smart Meter”, in *Proc. of Intern. Conf. Relay protection and automation for electric power system*, 2017.

[17] R.F. Nuqui, “State Estimation and Voltage Security Monitoring Using Synchronized Phasor Measurement”, Ph.D. dissertation, Virginia Polytechnic Institute & State University, Blacksburg, USA, 2001.

[18] O. Gomez, C. Portilla, M.-A Rios, “Reliability analysis of substation monitoring systems based on branch PMUs”, *IEEE Trans.on Power Systems*, Vol. 30 (2), 2015.

[19] GOST R ISO/IEC 12119-2000. “Information technology. Software packages. Quality requirements and testing”, [Online]. Available: http://docs.cntd.ru/document/gost-r-iso-mek-12119-2000 (in Russian).

[20] GOST R ISO/IEC 12207-2010 “System and software engineering. Software life cycle processes”, [Online]. Available: http://docs.cntd.ru/document/1200082859 (in Russian).

Kolosok Irina N. received Engineering degree from the Saint-Petersburg Technical University in 1972. Since 1985 she has been with Melentiev Energy Systems Institute (MESI) SB RAS, currently as a leading researcher. She received the Ph.D’86 and D.Sc’04 degrees. Her scientific interests are real-time control problems, SCADA and WAMS. She is senior member of IEEE PES. M’ 1997, SM’ 2007.

Korkina Elena S. graduated from Irkutsk Polytechnic Institute in 1978. Since 1987 she has been working at the MESI SB RAS in the laboratory of electric power system operation control problems. She received the Ph.D in 2009. Her scientific interests are real-time control problems, SCADA, WAMS.

Tikhonov Alexander V. graduated from Irkutsk State University in 2007. Since 2009 he has been working at Joint Stock Company «Irkutsk electrical network company». He received the Ph.D in 2017 (MESI SB RAS). His scientific interests are EPS state estimation, FACTS devices and their mathematical modeling.