Software test and validation of wireless sensor nodes used in nuclear power plant

DENG Changjian[1,2]**, CHEN Dongyi**[3]**, and ZHANG Heng**[4]

1. Department of Control Engineering, Chengdu University of Information Technology, 610225, P.R. China (chengli_dcj@163.com)
2. School of Automation Engineering, University of Electronic Science and Technology of China, 611731 P.R. China. Correspondent author.
3. School of Automation Engineering, University of Electronic Science and Technology of China, 611731, P.R. China (713911@qq.com)
4. School of computer and information science, Southwest University, 400715, Chongqing, P.R. China, (8131731@qq.com)

Abstract: The software test and validation of wireless sensor nodes is one of the key approaches to improve or guarantee the reliability of wireless network application in nuclear power plants (NPPs). At first, to validate the software test, some concepts are defined quantitatively, for example the robustness of software, the reliability of software, and the security of software. Then the development tools and simulators of discrete event drive operating system are compared, in order to present robustness, reliability and security of software test approach based on input-output function. Some simple preliminary test results are given to show that different development software can obtain almost same measurement and communication results although the software of special application may be different than normal application.

Keyword: software test; wireless sensor nodes; software robustness; software reliability; software security

1 Introduction

As demand of security, robustness, reliability, the wireless equipment condition monitoring system and its sensor nodes of NPPs need to conduct on software test and validation.

Traditional embedded software test is often based on Parnas' rational design process [1]. And it includes: hazards analyses, trip computer design requirements, trip computer design description, requirements review, software design description, software design review, software design verification, coding, code review, code verification, a variety of testing procedures, a procedure for maintenance and revision of all documents and so on [2].

But Wireless Sensor Network (WSN) is a distributed and embedded computation system consists of autonomous, and cooperating embedded sensor nodes. Each sensor node should acquire and process test data, communicate to network access appoint, store valid and middleware data, and so on. In recent years, there are lots of researches of its software test and validation method.

For example, researches have studied the providing instruments for the evaluation in form of dedicated test beds [3, 4, 5, 6, 7], simulators [8,9,10], emulators [11],and so on. They often concerned two problems: (i)functional properties such as the amount of successful collected data from the WSN, and (ii)non-functional properties such as the energy-efficiency of the system.

Some researches study the online software test and validation method, for example they use SBST (Software-Based Self-Test) method to ensure the reliability of WSN nodes using in harsh environment [12, 13], and so on.

In IEEE-1012(2004), validation is defined as: "(A) the process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements. (B) The process of providing evidence that the software and its associated products satisfy system requirements allocated to software at the end of each life cycle activity, solve the right problem(*e.g.*, correctly mode physical laws, implement business rules, use the proper system" [14]

There are lots of V&V examples of wired digital I&C system in NPPs, but how to test software of the

Received date: June 1, 2015
(Revised date: June 3, 2015)

wireless sensor nodes and do V&V process, is still a challenging subject.

The authors of this paper focus on using the software test methods to estimate the normal art-of-use WSN protocol software or programs, and to this end, the security of software, the robustness of software and the reliability of software should be defined firstly, before the trial software test are undertaken. The goal of paper is to give an instance of software test and validation of wireless sensor nodes; to present the robustness, reliability and security of software test method based on input-output function in NPPs; and to study its theory and technological fundamentals.

The rest of paper is organized as follow: the section II is problem formulation and the analysis of exist method; Section III present a robustness, reliability and security of software test method based on input-output function; Section IV is some simple alpha software test result; Section V is conclusions.

2 The problem formulation and software test methods analysis

2.1 The reliability of software

In NPPs, the reliability of software is an import issue in a safety critical system.

There are three main methods to estimate software reliability: the Verification and Validation (V&V) quality-based method, the Software Reliability Growth Model (SRGM), and the test-based method.
The adopted method currently is often based on software reliability growth model (SRGM).it is not appropriate to be applied to safety-critical software. Although the V&V method can be utilized in safety critical software, but before to complement it, the test-based methods need to be developed [15].

In IEEE-1012(2004), verification is defined as: “(A) the process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. (B) The process of providing objective evidence that the software and its associated products conform to requirements(*e.g.*, for correctness, completeness, consistency, accuracy) for all life cycle activities during each life cycle process(acquisition, supply, development, operation, and maintenance); satisfy standards, practices, and conventions during life cycle processes; and successfully complete each life cycle activity and satisfy all the criteria for initiating succeeding life cycle activities(*e.g.*, building the software correctly)” [14].

The other methods are developed from these three methods; for example, a software reliability estimation method based on the software failure model and test results is presented to estimate safety-critical software reliability, which comes from SRGM. Wang, *etc.* had used this method to estimate the software reliability of the reactor protection system (RPS) for Lungmen ABWR[16].

To analyze the reliability of software in quantity, it should define some concept in quantity, for example, software reliability is defined as the probability for failure-free operation of a program for a specified time under a specified set of operating conditions.
It can be formulated as

$$R = \frac{1}{\lambda} \tag{1}$$

where R is reliability degree, λ is loss function rate of system.

2.2 The robustness of software

From the view of computer science, robustness is the ability of a computer system to cope with errors during execution. Robustness can also be defined as the ability of an algorithm to continue operation despite abnormalities in input, calculations, *etc*.

Various commercial products perform robustness testing of software systems, and are used for a process of failure assessment analysis.

Robust programming, also called bomb-proof programming, is a style of programming that prevents abnormal termination or unexpected actions. A robust program and software test adherence to the following four principles:
(i)Paranoia: Don't trust anything you don't generate.
(ii)Stupidity: Assume that the caller or user is an idiot, and cannot read any manual pages or documentation.

(iii)Dangerous Implements: A "dangerous implement" is anything that your routines expect to remain consistent across calls.
(iv)Can't happen: never say `never', impossible cases are rarely that; most often, they are merely highly unlikely.

The robustness of software means two things in quality: one is how many unexpected input or change it can stand than ordinary software, the other is how many errors of system can be decreased under harsh environment than ordinary software (or a predefined reference software).

It can be formulated as

$$M_{ROS} = SOE_{RS} - SOE_{OS} \quad (2)$$

Or

$$E_{ROS} = EOS_{RS} - EOS_{OS} \quad (3)$$

In Eq. (2), M_{ROS} is degree of robustness of software measured by unexpected input or change number, SOE means stand of error number, RS means robust software, OS means ordinary software.

In Eq. (3) E_{ROS} is degree of robustness of software measured by error number using different software, EOS means error of software, and RS means robust software, OS means ordinary software.

In test method, there are stress software test, anti-structure change test, and so on.

2.3 The security of software or its properties of the anti-attack

The security of software means the less risk of software the more attacks can be defended.
To defend attacks in sensor networks, one should study the properties of attacks: Outsider attacks are attacks from nodes which do not belong to a WSN, its counterpart is Insider attacks. Active attacks involve some modifications of the data stream or the creation of a false stream, its counterpart is passive attacks.

It is different with Security software, the security software is any computer program designed to enhance information security.
To measure the security of software in quality, in the paper it is defines as

$$SES_{ATA} = \sum ATA \times risk_{ATA} \quad (4)$$

In Eq. (4), SES is the security of software, ATA is kinds of attack, $risk_{ATA}$ is the risk value of attack.
In WSN, attacks include[17]:
1) Denial of Service (DOS): it attack the adversary attempts to subvert, disrupt or destroy a network.
2) Desynchronization: Disruption of an existing connection is desynchronization.
3) Data Integrity Attack: Data integrity attacks are caused by changing the data contained within the packets or injecting false node whiles the data travelling among the nodes in WSN.
4) Sybil attack tries to degrade the integrity of data security and resource utilization that the distributed algorithm attempts to achieve.
5) Blackhole attack means a malicious node acts as a blackhole in the range of the sink attracts the entire traffic to be routed through it by advertising itself as the shortest route.
6) Wormhole attack is a significant attack in WSN. This attack occurs at the initial phase when the sensors start to discover the neighboring information, and so on.

The relationship among the reliability of software, the robustness of software and the security of software can be explained in Fig.1. To calculate their value, the boundary of these three concepts is defined. For example, for the robustness of software test error or fault, it does not test failure, so it is normally used in unit testing, integration testing, and system testing; the reliability of software often test failure, so it is normally used in system testing, acceptance testing, application testing; and the security of software test loss or damage, so it is normally used in application testing, system testing, acceptance testing. Note: in a board sense, all these three concept are called the reliability of software in some application,

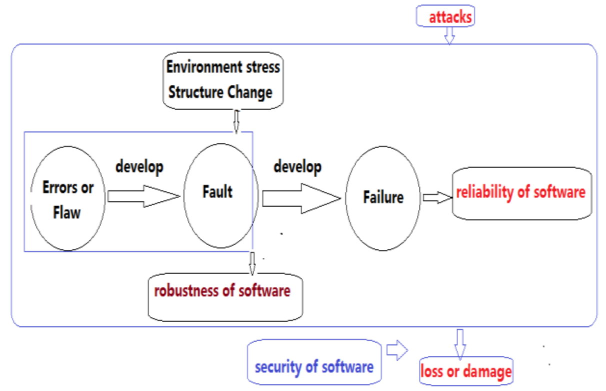

Fig.1 The relationship among the robustness of software, the reliability of software and the security of software.

2.4 The software test method analysis

As is introduced in the Section I, Wireless Sensor Network (WSN) is a distributed and embedded computation system; its software test method is more different or complex than ordinary field test method. There are four kinds of development tools other used in wireless sensor networks, as shown in Table 1.

Table 1 Normal development tools in WSN and its feature

Development tools	**Core**	**Feature**
IAR-51 …	C51 (CC2530...)	Operating System or no OS
IAR-ARM	ARM7,CORTEX M3(MC13224...)	Often have Real Time Operating System
Cygwin or Linux	MSP430, AVR (MICAZ…)	TinyOS
QUATUIS II	FPGA	Software define

It is different to test software of WSN nodes in some application fields, so it is important to simulate or estimate the software test. It is very useful in WSN, where the black box, white box, or grey box software tests methods are difficult to test the distributed network event in an operating system.

Consider that many of the current simulators are unable to model many essential characteristics of the real world, so only common simplified assumptions errors, faults and even the failures in the software are tested in simulators.

And the other characteristics should be test independently, or in special way. This is why the paper uses input-output function to test the software.

The referenced paper [18] introduced 63 simulators or simulation frameworks, 14 emulators, 19 data visualization tools, 46 test-beds, 26 debugging tools, 10 code reprogramming tools and 8 network monitors.

Table 2 compares normally used software test simulators in developing the wireless sensor nodes. And in wireless sensor nodes there are four software test levels: unit testing, integration testing, system testing, and acceptance testing.

The normal part (is not distributed and networked) of software testing method also based on white-box and black-box testing to ensure its local accuracy and effectively. It is same to analyze the robustness of software, the reliability of software and the security of software.

In the distributed and network software test or program part of wireless sensor nodes, often use static and dynamic instrumentation test, sniffer , and so on.

In unit testing and integration testing the debugger tools in Table 2 are often adopted methods. In system testing, acceptance testing, and protocol analysis, the simulators, emulators, and hardware protocol analyzer (or sniffer) are applied.

Table 2 Normal software test simulator in WSN

simulator	**Feature**
System test level simulators	
NS 3(NS 2)	discrete-event network simulator
OpenNET	Commercial discrete-event network simulator
TOSSIM	TINYOS-- algorithms and implementations
OMNet++	An Eclipse-based IDE and a graphical discrete event simulator.
JavaSim	component-based, compositional simulation environment
Sinalgo	the verification of network algorithms, and abstracts from the underlying layers
Capricorn	large-scale discrete-event wireless sensor network simulator
motesim	full-system, cycle-accurate simulator for Mica-2 and Mica-Z motes
Emulator and application simulator	
VMNet	takes parameter values and running status of application code
Freemote	lightweight and distri-buted Java based emulation tool for developing WSN

	softwares
Debugger and Code test	
Clairvoyant	comprehensive source-level debugger for wireless, embedded networks
Sympathy	detecting and debugging failures in sensor networks
Marionette	system that allows calling functions and inspecting and changing memory locations in deployed nodes through an RPC-based system
Passive Distributed Assertions (PDA)	detect such failures and provides hints on possible causes
Post Deployment Performance Debugging (PD2)	a data-centric approach that focuses on the data flows that an application generates, and relates poor application performance to significant data losses or latencies of some data flows

Besides Table 2, Matlab is a powerful software tool often used to valid the science arithmetic and communication protocol, especially the ‘truetime’ network control simulator, and the ‘V&V base on model’ simulation tool. For it is widely accepted by its powerful calculate ability and simply realized.

When in the process of testing software, there is another issue need to be noticed that some nodes may be multi core, some may be single core, or be FPGA core. There are concentrate test method, distributed test method, and digital signals test method as shown in Table 3.

Table 3 the different wireless nodes and there test methods

	Software feature	Test method
Multi core	Depended and independent	Concentrate test Or distributed test
Single core or single FPGA	OS or no OS	concentrate test
Composite FPGA and MCU	System architecture	Digital signal test

3 The robustness, reliability and security of software test method based on input-output function

As analysis in Section 2, the main problem of software test in a distributed system is how to solve the problem of the error cannot be reproduced for the uncertainty of program executing. Although self-adaptability source code trace and replay technology can be used to diagnose some error of program. But in general, there need a method to estimate, to evaluate the software test, and validation the test.

To simplify the problem, paper focuses on studying on the properties of software test under harsh environment stress, attack, and unexpected failures. The software or programs to be tested or analyzed in the paper are shown in Table 4.

Table 4 the software and its example source to be tested

	Software	Source
MCU	ZIGBEE and Parameter test	1) MCCZ 2) OPEN-ZB 3) ZSTACK(CC2530)
	MESH	4) MESH(CC2538)
FPGA	SPI	Actel

3.1 Examples of general unit software testing and integration testing

Firstly, let look at a diagram of WSN node working program, as is shown in Table 5. Here it is considered a very simple case that the program executed in a normal execution sequence mode. The software test of left part in table is in familiar way, but the right part may be confused for there is an operating system in it.

Table 5 the application software of CC2531

No OS simplest procedure (often simple transmitting application)	OS simplest procedure (Zigbee application)

// Config basicRF
// Initialize board peripherals
// Initialize hal_rf
// Indicate that device is powered
// Set application role-test
// Transmitter application
// Receiver application
	// Turn off interrupts
	// Initialization for board
	// Make sure supply voltage
	// Initialize board I/O
	// Initialize HAL drivers
	// Initialize NV System
	// Initialize the MAC
	// Determine the extended address
	// Initialize basic NV items
	// Initialize the operating system
	// Allow interrupts
	// Final board initialization
	//run the operating system

But real time operating system are often event based having priority system. Then can reduce the number of test cases by equivalence partitioning in black-box testing method.

Secondly, the software of wireless sensor nodes based on Tinyos is used in many cases.

Table 6 the application software of MICAZ

.NC in application layer

```
// includes …
configuration ZwkDemo{
  }
implementation
{
   components    Main,
   /*StdControl*/
   Main.StdControl -> …

   //ADC
   //Temp sensor
   //light sensor
   // iic
   //timer
   //…..

   //declaration sap (service access point in application layer)
}
```

Although it is different from the normal C51 application, it is run only in two modes: an asynchronous mode, event triggered mode, or in synchronous mode, where events happen in parallel in fixed time slots.

- Synchronous Simulation: A simple example is triggering a task in nesC of TinyOS
- Asynchronous Simulation: it is event based. OS normally picks the most recent event and executes it.

For a discrete event operating system and NesC program language, software test based on Tinyos is harder than OS of CC2530, its test methods: static and dynamic instrumentation test, sniffer, self-adaptability source code trace and replay seem complex but clear , so in most case, the white-box testing and black-box testing can be used in the system.

Some software test based on WSN can be often simulated by tools in Table 2. Especially the tools in Table 2 are very useful in the system software tests and acceptance software testing level.

3.2 The robustness, reliability, security of software test method based on input-output functions

(1) The robustness of software

The robustness of software test has been discussed in paper [19], for example, seem to FTA analysis, a vibration module in wireless sensor nodes calculate it's acquired accelerate value. There are eight input (or stress) can produce the fault of test.

Here use Eq. (2) or (3), and the robust software has high value of M_{ROS} or E_{ROS} , and it can replace the plot like Fig. (2), and obtain the robustness of software value. This method is based on input-output functions.

(2) The reliability of software

In time reliability mode, the SRGM is explained as:

$$R(t) = e^{-C((E_0/I - E_c(\tau))t} \qquad (5)$$

$$MTTF = \int_0^{\infty} R(t)dt = \frac{1}{C(CE_0/I - E_c(\tau))} \qquad (6)$$

And

$$E_0 = K\frac{(N_1 + N_2)\log_2(n_1 + n_2)}{3000} \qquad (7)$$

Here E_0 is number of failure in initial time; $E_c(\tau)$ is the number of correct failure in [0,Γ]; C, K are constant number; I is number of instruction in software. N_1 is the sum number of operate code, N_2 is the sum number of operate number, n_1 is number of kinds of operate code, n_2 is number of kinds of operate number.

The time mode is hard to test for most wireless sensor network applications, the data mode based on input-output functions is used.

As shown in Table 5, there are lots unexpected input are not considered in simplest program, and some of them may make the software be failure (cannot do it work), for example, self-test and calibration, interfere decrease, communication channel or quality test and so on.

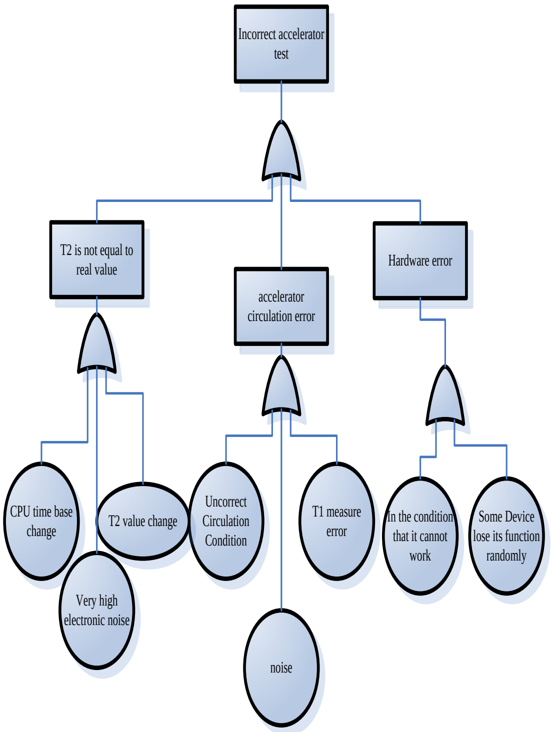
Incorrect accelerator test
T2 is not equal to real value
accelerator circulation error
Hardware error
CPU time base change
Very high electronic noise
T2 value change
Uncorrect Circulation Condition
noise
T1 measure error
In the condition that it cannot work
Some Device lose its function randomly

Fig.2 An FTA example of vibration test in wireless sensor node.

In a single task, the number of a input set is ‘E’, and the number of unexpected input set is ‘E_e’, then the reliability of software is defined as Eq. (8), n is number of executing.

$$R(n) = (1 - \frac{E_e}{E})^n \tag{8}$$

The classic failure in wireless sensor nodes can be listed in Table 7.

Table 7 the different kinds of fault of software

Kinds of fault	Description (software testing)
Communication fault	Not consider fading phenomena, interference and noise
data fault	No process method of missing data or duplicate data
Congress fault	Not consider number of communication nodes or their throughput
Data confuse fault	No change , disorder , no meaning of confuse data (buffer overflow, replace, missing, and so on)
Protocol arithmetic system fault	Simulate use tools in Table 2

(3) The security of software

To simplify the problem, each of attack is corresponded with an input set. The example input set is list in Table 8. Then use Eq. (4) to calculate the security of software, and this method is also based on input-output function.

Table 8 the different attacks and their input set

Kinds of attack	Example test input set and method
Denial of Service	Example: More communication throughput, very quickly frequency of transmitting data. In detail, 1) Jamming 2) Tampering 3) Collision 4) Exhaustion 5) Unfairness 6) Flooding ----simulate in tools in table 2, or alpha test.
Data Integrity Attack	Example: incorrect transmitting packet in communication ----simulate in tools in table 2, or alpha test.
Sybil attack	Example: more distribution of subtasks and redundancy of information ----simulate in tools in table 2, or alpha test
Blackhole attack	Example: high rate of packet loss ----simulate in tools in table 2, or alpha test
….	….
Wormhole attack	Example: transmitting data between unexpected nodes Simulate use tools in Table 2

4 Some alpha test results

(1) The temperature alpha software test using Tinyos or C51

Paper analyzes the temperature test in different software. The test results using different software are almost same at the certain temperature value.

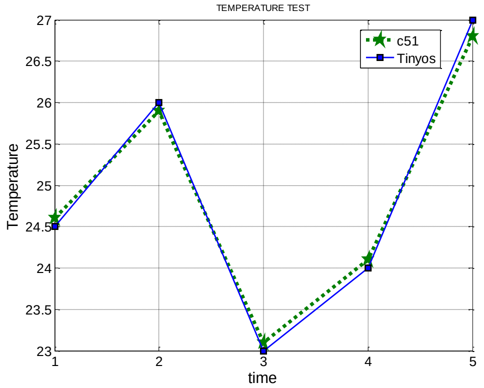

Fig.3 The temperature test using tinyos and C51.

(2) The communication alpha software test using Tinyos or C51

Using Atmega 128, in one time, node use Tinyos and in the time use AVR c51 program, realize point to point communication test. The communication distance test result is shown in Fig.4. And the ZIGEE protocol test have almost same value.

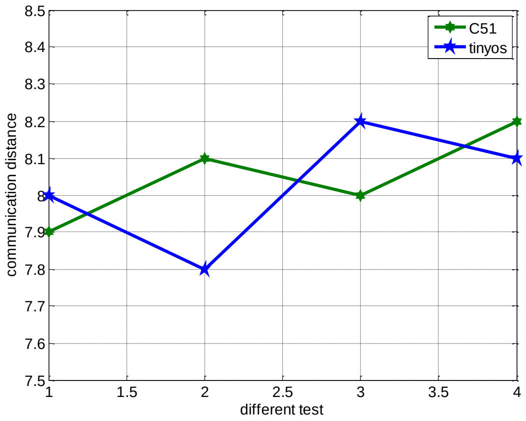

Fig.4 The communication test using tinyos and C51.

(3) Other test

The robustness of software, the reliability of software, and the security of software test results for Table 4 are large based the software design feature. For example, they are almost not very robustness, security and reliability, for they are designed to apply in normal situation. Not design for harsh environment and safety related use.

But fortunately, the software can be tested and simulated, so the worry of unreliability of network application or wireless application may be unreasonable in most case. For in ordinary application, these values may be not very critical.

5 Conclusions

Paper analyzes the software test methods which can be applied into the wireless sensor. Normal software test simulator and development tools in WSN are compared.

To estimate and validate the WSN software or programs, the security of software, the robustness of software, and reliability of software are defined in the paper. And use these performances to validate a software use in NPPs.

As their calculations are difficult and indirect, paper presents the robustness, reliability and security of software test method based on input-output function in NPPs.

By these ways, the software of wireless sensor nodes can be tested and validated. And these efforts can less the worry about the unreliability of network application or wireless application.

Nomenclatures

NPPs: Nuclear power plants
V&V: Verification and Validation
SRGM: Software Reliability Growth Model
WSN: Wireless sensor network
FTA: Fault tree analysis
OS: Operating system
ABWR: Advanced Boiling Water Reactor

Acknowledgement

This paper is supported by Canada-China Scientific and Technological Collaboration Project 2009DFA12100 “Development of Wireless Sensor Network based Devices and Systems for Equipment Health Monitoring in Nuclear Facilities” .

References

[1] IEEE: Draft Guide for Application of Monitoring to Liquid Immersed Transformers and Components. IEEE Unapproved Draft Std. PC57.143/D20, Apr 2008

[2] KALDENBACH, B. J., MOORE, M. R., EWING, P. D., MANGES, W. W., DILLARD, C. L., KORSAH, K., and KISNER, R. A.: Assessment of Wireless Technologies and Their Application at Nuclear Facilities, Oak Ridge National Laboratory: NUREG/CR-6882.2006

[3] HOWLADER, M., KIGER, C. J., and EWING, P. D.: Coexistence Assessment of Industrial Wireless Protocols in the Nuclear Facility Environment, Oak Ridge National Laboratory: NUREG/CR-6939.2007.

[4] KORSAH, K., HOLCOMB, D. E., MUHLHEIM, M. D., and MULLENS, J. A., *et al*: Instrumentation and Controls in Nuclear Power Plants: An Emerging Technologies Update, Office of Nuclear Regulatory Research: NUREG/CR-6992,2009

[5] HOWLADER，M. K., EWING， P. D.， and DION, J: Issues associated with deploying wireless systems in

nuclear facilities, Las Vegas, Nevada NPIC&HMIT 2010, November 7-11, 2010,

[6] EPRI: Plant Support Engineering: Aging Management Program Development Guidance for AC and DC Low-Voltage Power Cable Systems for Nuclear Power Plants, Report No. 1020804, Palo Alto, California: Electric Power Research Institute (EPRI), 2010.

[7] AKYOL, B. A., KIRKHAM H, CLEMENTS, S. L., and HADLEY, M. D.:A Survey of Wireless Communications for the Electric Power System, Technique report of PNNL:PNNL-19084,January 2010

[8] KUROTANI, K., and TODAKA, Y.: Current Status and Prospects of Measurement and Control Technologies. Technique report of IAEA:2012

[9] HEO, G. Y.: Condition monitoring using empirical models: technical review and prospects for nuclear applications. Nuclear engineering and technology.Vol.40 No.1 February 2008.

[10] EPRI: Electromagnetic Interference Testing of Power Plant Equipment. TR-102323.2006

[11] RUSAW, R.: Implementation Guideline for wireless networks and wireless equipment condition monitoring. EPRI reports, December 2009

[12] BANEJEE, B.: Performance Implications for Wireless Systems and Networks. EPRI reports, November 2001

[13] WOOD, R. T., EWING, P.D., KERCEL, S. W., and KORSAH, K: Electrometric compatibility in nuclear power plants. DE-AC05-96OR22464

[14] EWING, P. D., and WOOD, R. T.: Recommended Electromagnetic Operating Envelopes for Safety-Related I&C Systems in Nuclear Power Plants, NUREG/CR-6431, U. S. Nuclear Regulatory Commission, 1999.

[15] HASHEMIAN, H.: Wireless Sensors for Predictive Maintenance of Rotating Equipment in Research Reactors, Ann. Nucl. Energ, 2011, 38:665-680.

[16] ZHANG, H., Li, L., CHEN, D. Y., WU, Y. W., and LING, J. ZH.: A novel wsn relay nodes placement strategy in NPP equipment monitoring application: WiCOM Shanghai, 2012.

[17] BOND, L. J., COBLE, J., and BOND, J. W. D.: Economics of On-line Monitoring and Nuclear Power Plant Life Extension, Richland, Washington: Pacific Northwest National Laboratory, 2012 (draft).

[18] BOND, L. J., and MEYER, R. M.: Online Monitoring to Enable Improved Diagnostics, Prognostics and Maintenance. In International Symposium on Future I&C for Nuclear Power Plants, Cognitive Systems Engineering in Process Control, International Symposium on Symbiotic Nuclear Power Systems (ICI 2011), Daejeon, Korea, 2011.