Control Sensor Network Configuration Management

Nikolay Zakharov
MTUCI University
Moscow, Russia
nazakharov@npp-dozor.ru

Vladimir Klepikov, Dmitry Podkhvatilin
JSC "KEMZ"
Moscow, Russia
viklepikov@mail.ru, dspodkhvatilin@npp-dozor.ru

Abstract—**An approach to the configuration management of a sensor network used in a distributed control system with redundant components is proposed. The approach is based on a formalized vector-matrix representation of sets of system components, possible system configurations and functional relationships between them. Configurations and their components availability evaluation is based on configurations results comparison.**

I. INTRODUCTION

The development of sensor network (SN) technologies makes it possible to create distributed control systems (DCS) for critical control applications that operate under difficult operating conditions. The example of integrated Control and Safety system inside Oil and Gas plants based on SN is given in [1]. This system is composed of both Safety instrumented control system for emergency process conditions and Distributed control system for normal process conditions. DCS for critical control applications are to continue operation in case of failures of their components. Such DCS implementations include, for example, industrial plants, especially chemical and petrochemical, gas transportation, power generation, aviation systems [2, 3] and, in particular, gas turbine engine control systems [4]. The use of SN for controlling and monitoring the condition of aircraft equipment reduces the use of heavy and complex wiring harnesses, which limit the versatility of the system, and complicate its maintenance. The addition of wireless technologies to wired networks opens up opportunities for reducing the number of cables, faster deployment of sensors and networks, increased flexibility in data collection, and lower costs for cable production and maintenance.

SN can be represented as a set of jointly and purposefully functioning distributed dynamic objects and in accordance with modern systems theory is classified as a complex dynamic system. In relation to critical control applications, such as aviation, in such systems, according to the RTCA DO-254/ EUROCAE ED-80 Design Assurance Guidance for Airborne Electronic Hardware special requirements should be provided, such as redundancy, ensuring heterogeneity of communication channels, hardware and software components, and algorithmic support. To ensure the reliability and fault tolerance of control and condition monitoring systems under conditions of possible failures of system components and taking into account the heterogeneity of system elements the concept of building a DCS based on maintenance-free modular electronics is currently being actively developed [2]. For such systems, the method of dynamic reconfiguration of distributed system resources [5] is useful.

There are different methods of SN components diagnostics. We offer using different configurations results comparison to detect unhealthy configurations. Comparison of healthy and unhealthy configurations components sets allows detection of faulty components.

II. SN BASED REDUNDANT DCS

Further, the article discusses building a SN based DCS with wired and wireless communication channels. The DCS has an excessive number of hardware, software, communication, and software components. In general, the components are:

- sensors and input devices;
- actuators and signal output devices;
- computing nodes (controllers);
- communication lines between nodes and with adjacent systems;
- built-in mathematical models.

All the components regardless of their functional purpose and physical nature are connected by communication links and participate equally in the implementation of a certain system function. All these components will be called resources, and each variant of the resources set will be called a configuration. In the SN various resources can be used to implement each system function – sensors, controllers, communication channels, actuators, etc. During the operation of the SN, failures of individual components may occur, and failures may occur in the processes of measurement, calculation, and data transmission. Due to hardware, computing, information and time redundancy the SN can continue to function combining in configurations remaining serviceable components. The process of serviceable components selection and building the required configurations from them is called "DCS resources redundancy management".

To manage the redundancy of the DCS, various methods are used, in particular, the method of configuration supervisors (CS) [6], [7]. In the original method [2] CS refers to software and hardware modules used to monitor the health of configurations and identify the leading configuration in between supervisors arbitration.

A feature of the SN based DCS is the capability of it's computing nodes to simultaneously, almost synchronously receive information from multiple sensors and from many other nodes. This allows you to execute in parallel multiple configurations that implement identical system functions in different ways without additional coordinating actions.

Designed for critical control applications DCS must have built-in diagnostics methods. For sensor network nodes, it is most effective to use tolerance control [6], digital filters, and state observers [8] based on built-in mathematical models. An example of the Kalman filter implementation in an industrial plant is given in [9]. Application of certain algorithms based on a bank of Kalman filters to detect parametric faults in sensors and actuators in a dedicated complex for configuration, support and diagnostics of control systems is considered in [10].

Obtaining the results of many different configurations that implement identical functions allows, first, to get a more reliable estimate of the overall result by averaging particular results, and second, to compare the results obtained in order to identify failures and faults of included in the configuration components.

III. REDUNDANCY MANAGEMENT METHOD

Next, the article discusses a method of an SN-based DCS redundancy management, in that:

- values (partial results) of a certain system function are generated in parallel by several different configurations of SN components,
- the trustworthiness of configurations is checked by comparing the obtained partial results with each other,
- based on the results of checking the trustworthiness of configurations, estimates of the serviceability of SN components that participated in the operation of the corresponding configurations are formed,
- estimation of the overall result of the system function is obtained by averaging the partial results recognized as reliable,
- SN components that are found to be faulty are excluded from the process of creating new working configurations.

Let's consider an SN (Fig. 1) containing m resources and intended for implementing l system functions. Then n variants of configurations can be constructed for such a SN. Different configurations can implement either different or the same system functions, configurations can have completely disjoint sets of components, or they can use some common resources.

Let the SN (Fig. 1) contains m = 12 nodes: sensors $x_1...x_5$, actuators x_{11}, x_{12}, routers x_6, x_7, and servers x_8, x_9, and x_{10}. Let us also assume that communication links between nodes are possible in this network, shown as a dotted line. The SN is designed to implement l = 3 system functions (f^1, f^2, f^3); the functions f^1 and f^2 are control functions and the results of their operation (z^1, z^2) are issued to the actuators, respectively x_{11} and x_{12}; the function f^3 is diagnostic and the result of it's operation z^3 is used to detect failures of identical sensors x_1 and x_5.

A configuration is a set of components of the system $C = \{x_1, x_2,..., x_m\}$ that provide a certain function. The configuration can provide both the object management function, i.e., end with the calculation of the output value of the system, and perform intermediate calculations that determine the values of parameters necessary for other configurations. For example, function can calculate the most reliable values of the DCS input parameters based on the readings of several sensors. The same function in the control system can be implemented in different configurations, depending on which components are currently in good condition.

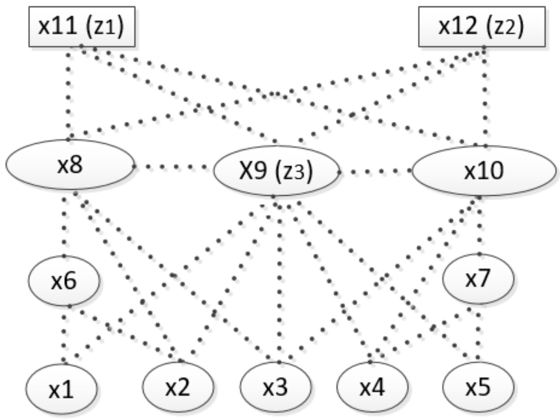

Fig. 1. Example of a control sensor network

In this example, the function f^1 can be implemented in various alternative choices of sensors x_1 and x_2, (result $z^1{}_1$) or sensors x_3 and x_4 (output $z^1{}_2$), or sensors x_2 and x_5 (the result $z^1{}_3$); function f^2 can be implemented two different options for sensors x_2 and x_3 (the result $z^2{}_1$), or – in sensors x_4 and x_5, (result $z^2{}_2$). Thus, we can say that each system function can be implemented by many configurations, which may have completely disjoint sets of components, or may use some common resources for them.

Let us denote as $C^j{}_l$ the l-th configuration of the SN component that implements the j-th system function. Such a configuration will be called functional configuration. As an example, Fig. 2 shows the possibility of implementing the function f^1 functional configurations $C^1{}_1 = \{x_1, x_2, x_6, x_8\}$, $C^1{}_2 = \{x_3, x_4, x_8\}$ and $C^1{}_3 = \{x_2, x_5, x_9\}$, the functions f^2 – configurations $C^2{}_1 = \{x_2, x_3, x_9\}$ and $C^2{}_2 = \{x_4, x_5, x_7, x_{10}\}$ and the function f^3 – configurations $C^3{}_1 = \{x_1, x_9\}$ and $C^3{}_2 = \{x_5, x_9\}$.

Thus, in the SN under consideration, seven configurations can work simultaneously, forming the following results:

$$
\begin{aligned}
z^1{}_1 &= f^1{}_1(x_1, x_2, x_6, x_8),\\
z^1{}_2 &= f^1{}_2(x_3, x_4, x_8),\\
z^1{}_3 &= f^1{}_3(x_2, x_5, x_9),\\
z^2{}_1 &= f^2{}_1(x_2, x_3, x_9),\\
z^2{}_2 &= f^2{}_2(x_4, x_5, x_7, x_{10}),\\
z^3{}_1 &= f^3{}_1(x_1, x_9),\\
z^3{}_2 &= f^3{}_2(x_5, x_9).
\end{aligned} \tag{1}
$$

In Fig. 2 communications between the components that implement the configuration of the function f^1 are shown by solid lines, communications between the components that implement the configuration function f^2, are shown by dotted lines, communications between the components that implement the configuration function f^3 point are shown by the dotted lines.

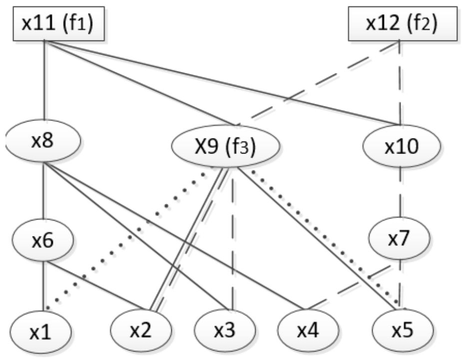

Fig. 2. Example of functions implementation by multiple components

Sensor networks that control complex objects can have hundreds of components that, depending on their current state, can be combined into dozens of different configurations to perform various functional tasks. The formation of such configurations must be performed either in advance at the system design stage, or generated automatically in real time, depending on the current situation. Without using formalized design methods, both approaches are very labor-intensive and require a lot of “manual” work at the stages of system design and testing. This makes it necessary to develop analytical methods for representing the set of configurations of available resources and managing these configurations in real time.

The developed approach to SN configurations management involves:

- a formalized representation in vector-matrix form of all components, used configurations and sets of components involved in each configuration;
- a formalized procedure for direct calculation of the vector of healthy configurations $\boldsymbol{y}$ based on the original vector of components health $\boldsymbol{x}$;
- calculation of the estimates vector $\hat{\boldsymbol{y}}$ of operable configurations based on the configurations outputs comparison;
- a formalized procedure for component health vector reverse calculation based on the vector $\hat{\boldsymbol{y}}$ of healthy configurations estimates.

Components in expressions (1) can be placed in arbitrary order, because these expressions are intermediate and serve for informal representation of configurations. Further, expressions (1) for sets of components x_k included into functional configurations C^l_j can be written as a matrix K of dimension (n×m), where n is the number of configurations under consideration, and m is the number of all system components involved in the building of these configurations.

We will say that the matrix K describes the system configuration and consists of functional configurations written in the rows of the matrix K, i.e. K = $[K_1, \dots K_i, \dots K_n]^T$, where K_i is the i-th row of the matrix K describing the corresponding functional configuration. The element k_{ij} of the matrix K is equal to 1 if the component x_j is involved in the K_i-th functional configuration. For this example, the system configuration matrix is written as:

$$K = [C^1_1, C^1_2, C^1_3, C^2_1, C^2_2, C^3_1, C^3_2]^T =$$

$$= [K_1, K_2, K_3, K_4, K_5, K_6, K_7]^T$$

or as following:

$$K = \begin{bmatrix} 1\,1\,0\,0\,0\,1\,0\,1\,0\,0\,0\,0\,0 \\ 0\,0\,1\,1\,0\,0\,0\,1\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0 \\ 0\,1\,1\,0\,0\,0\,0\,0\,1\,0\,0\,0 \\ 0\,0\,0\,1\,1\,0\,1\,0\,0\,1\,0\,0 \\ 1\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0 \\ 0\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0 \end{bmatrix}.$$

Let us define the vector x that characterizes the current state (healthy/faulty) of all components of the SN. If all the components of the SN are healthy, then the vector $\boldsymbol{x}$ is represented as:

$$x_s = 1,\ s = 1 \dots m.$$

If there are faulty components in the system, the corresponding elements of the vector $\boldsymbol{x}$ will be zero.

Let us denote by y_j, $j = 1...n$ the trustworthiness of the K_j-th functional configuration result and define the vector $\boldsymbol{y}$ of the trustworthiness of all configurations. If the result of the K_j-th configuration is recognized as trustworthy, then $\boldsymbol{y}_j = 1$, otherwise $\boldsymbol{y}_j = 0$, $j = 1 \dots n$.

The trustworthiness of the functional configuration is determined based on the following obvious statement:

Statement 1. The result of the K_j configuration is recognized as trustworthy if:

1) all SN components involved in the implementation of the Kj configuration are in good condition

$$\bigcap_{i=1}^{m} \left(k_{i,j} \supset x_i\right) = 1\ ,\ j = 1 \dots n.$$

where the $\supset$ symbol denotes the implication function described by the truth table shown in the Table I:

TABLE I. THE IMPLICATION FUNCTION TRUTH TABLE

$k_{i,l}$	1	0	1	0
x_i	1	1	0	0
$k_{i,l} \supset x_i$	1	1	0	1

2) the result z^1_a configuration C^1_a, which implements a function of the f^1_a, with some acceptable error ξ coincides with the result z^1_b, at least one of the other configurations C^1_b that implements the same outcome function f^1_b, and sets of components P^1_a of configuration C^1_a, components P^1_b of configuration C^1_b configurations have the only common element x_c, performing the results of z^1_a and z^1_b comparison, i.e.

$$(P^1_a \cap P^1_b) = x_c. \quad (4)$$

If the results of two configurations C^1_a and C^1_b do not coincide within the specified error, then these configurations should be considered inoperable, i.e.

$$y_a = 0, y_b = 0.$$

Pairwise comparison of the implementing identical functions configurations results allows us to form a vector of configurations availabilities:

$$\boldsymbol{y} = [y_1, \cdots y_j, \cdots y_n]^T.$$

A zero value of any component of the configurations availabilities vector indicates that the corresponding configuration has one or more failed SN components.

The health of the SN components is determined based on the following statement:

Statement 2. If at least one configuration that uses the s-th component is recognized as healthy, then this component is recognized as healthy.

Next, we define the function:

$$x_s = \bigcup_{j=1}^{n} \left(k_{j,s} \supset \hat{y}_i \right), s=1 \ldots m, \quad (2)$$

where the $\bigcup$ symbol denotes the logical "Or" function; the $\supset$ symbol denotes the implication function.

The meaning of expression (2) is that for each s-th element of the state vector of the component the operability of all $j = 1...n$ configurations in which it is involved is checked. If all configurations in which this component is involved are found to be inoperable, the component is identified as failed ($x_s = 0$).

The configuration management algorithm is shown in the Fig. 3.

The configuration management algorithm is executed cyclically. In the initial loop, all elements of the component health vector are assumed to be equal to 1. During the execution of configurations, the component health vector is corrected using the built-in SN diagnostics methods. After all the configurations are executed, the $\boldsymbol{y}$ configurations availabilities vector is formed by comparing the results of their work, and then a new vector of health ratings of the SN components is calculated based on it. If faulty configurations are detected, a new configuration matrix K is formed by replacing the faulty configurations with new ones or by replacing the faulty components in the current configurations with serviceable ones.

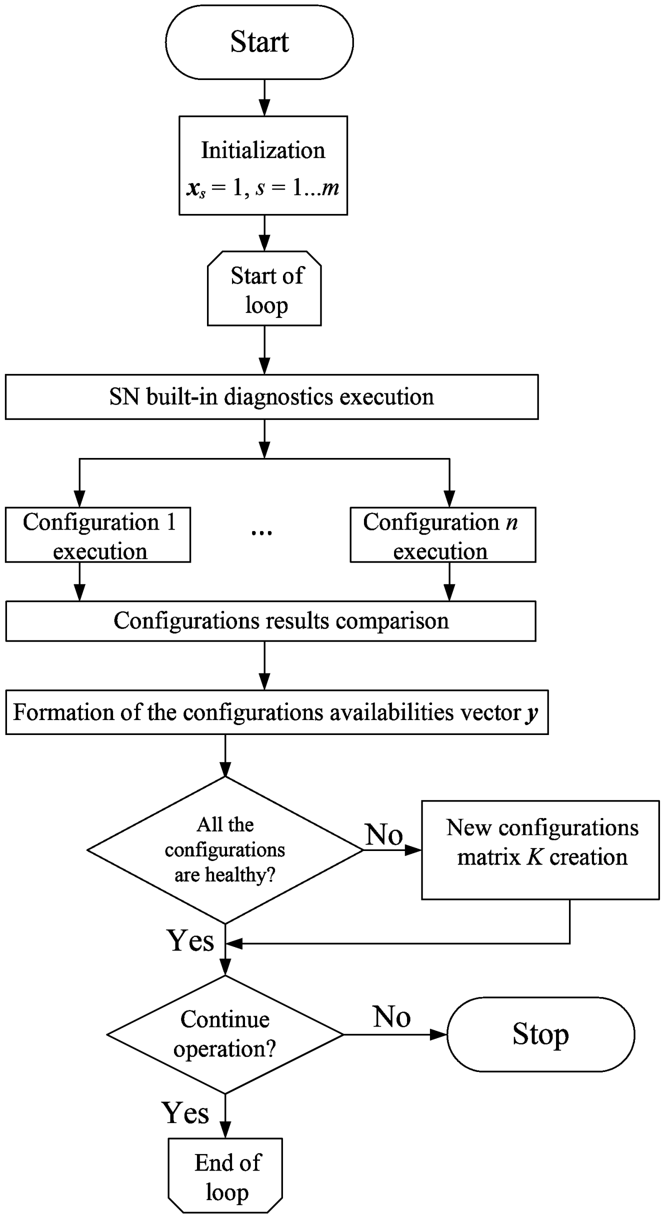

Fig. 3. Configuration management algorithm.

IV. Conclusion

Our approach based on feedback given by configurations results comparison gives additional tools to detect inoperable configurations and faulty components. It can be used along with components build-in diagnostics functions and other diagnostics methods (filters, observers etc.). No additional hardware is needed.

Both of configurations set and their component composition can be optimized according to various criteria, such as the DCS computing and communication resources load, results interpretation unambiguity. Optimization of the number and component composition of DCS configurations should be performed taking into account the depth and reliability of the built-in software and hardware self-diagnostics of DCS components. To build configurations presented in redundant SN components, the use of genetic algorithms is promising.

REFERENCES

[1] M. Wagih, A. Hamdy and W. Anis, “Efficient Wireless Sensor Networks in Centrifugal Compressors,” *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*, Coimbatore, India, 2019, pp. 163-167, DOI: 10.1109/ICACCS.2019.8728412

[2] V. Bukov, V. Kutahov, A. Bekkiev. “Avionics of zero maintenance equipment”. *27th Congress of the International council of the aeronautical sciences 2010, ICAS 2010*

[3] R. K. Yedavalli and R. K. Belapurkar, “Application of wireless sensor networks to aircraft control and health management systems,” *Journal of Control Theory and Applications*, vol. 9, no. 1, pp. 28–33, 2011.

[4] X.Dai, K.Sasloglou, R.Atkinson, et al. “Wireless Communication Networks for Gas Turbine Engine Testing”, *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 212876, 18 pages, DOI:10.1155/2012/212876.

[5] Ageev, A.M., Bronnikov, A.M., Bukov, V.N., Gamayunov, I.F. “Supervisory control method for redundant technical systems”, *Journal of computer and systems sciences international*, 2017, vol. 56, no 3, pp. 410-419. DOI: 10.1134/S1064230717030029.

[6] Zakharov N.A., Klepikov V.I. Podkhvatilin D.S. “Network redundant DCS configuration management”, *Advances in systems science and applications*, 2020, vol. 20, no 1, pp. 83-90, DOI: 10.25728/assa.2020.20.1.825

[7] Klepikov V.I. “Ensuring fault-tolerant computations in distributed control systems”, *Automation and Remote Control*, 2013, vol. 74(12), pp. 2112-2121.

[8] Osipova N.V. “Synthesys of asymptotic observer for magnetic separator control in iron ore processing”, *Mining informational and analytical bulletin (scientific and technical journal)*, 2018, no 6, pp. 153-160.

[9] Osipova N.V. “The use of Kalman filter in automatic control of indicators of iron ores magnetic concentration”, *Izvestiya. Ferrous Metallurgy*, 2018, vol. 61, no. 5, pp. 372–377.

[10] Klepikov V.I., Podkhvatilin D.S., Dudorov Y.N., Sharapov G.V., Zakharov N.A. “Information-measuring diagnostics complex for technical maintenance”, *Automation and Remote Control*, vol. 72, no. 5, pp. 1089-1094, DOI: 10.1134/S0005117911050171.