Distributed Cooperative Fault Diagnosis Method for Internal Components of Robot Systems

Hitoshi Kono *kono@robot.t.u-tokyo.ac.jp*
Department of Precision Engineering
The University of Tokyo
Tokyo 113-8656, Japan

Musab Obaid Alhammadi *musab.alhammadi@kustar.ac.ae*
Department of Electrical and Computer Engineering
Khalifa University
Abu Dhabi 127788, United Arab Emirates (UAE)

Yusuke Tamura *tamura@robot.t.u-tokyo.ac.jp*
Department of Precision Engineering
The University of Tokyo
Tokyo 113-8656, Japan

Atsushi Yamashita *yamashita@robot.t.u-tokyo.ac.jp*
Department of Precision Engineering
The University of Tokyo
Tokyo 113-8656, Japan

Hajime Asama *asama@robot.t.u-tokyo.ac.jp*
Department of Precision Engineering
The University of Tokyo
Tokyo 113-8656, Japan

Abstract

Robot systems have recently been studied for real world situations such as space exploration, underwater inspection, and disaster response. In extreme environments, a robot system has a probability of failure. Therefore, considering fault tolerance is important for mission success. In this study, we proposed a distributed cooperative fault diagnosis method for internal components of robot systems. This method uses diagnostic devices called diagnosers to observe the state of an electrical component. These diagnosers execute each diagnosis independently and in parallel with one another, and it is assumed that they are interconnected through wireless communication. A fault diagnosis technique was proposed that involves gathering the diagnosis results. Further, computer simulations confirmed that the distributed cooperative fault diagnosis method could detect component faults in simplified fault situations.

Keywords: Fault Detection, Distributed Cooperative System, Internal Component, Robot System.

1. INTRODUCTION

Robot systems have recently been deployed in many real world situations. Among other applications, mobile robot inspection systems for disaster-stricken areas, underwater inspection, and space satellites have been developed. The benefits of robot systems, particularly rescue robots, have been demonstrated in extreme environments [1], [2]. Herein, robot system refers to a mobile robot, such as rescue robots. These systems can decrease the risk associated with dangerous work and increase work efficiency. However, it is difficult to understand the state of an extreme environment. A robot system has a probability of failure because extreme environments are dangerous not only for humans but also for robot systems [3].

In this context, discussion of fault tolerance is important to understand the reliability of robot systems. To date, many types of fault tolerance methodologies have been considered. For example, model-based, signal-based, and model-free methods have been proposed [4], [5]. These approaches assumed that fault tolerance was considered when the robot system was developed. However, many robot systems are already in operation worldwide, and it is not easy to implement fault tolerance capabilities for these robot systems. Discussion regarding attachable fault tolerances, specifically fault detection method and diagnosis for currently operational robot systems, is important.

Previous research on general fault detection and diagnosis methods have focused on centralized architecture, and the systems required a large number of sensors [6]–[8]. Moreover, in the model-based method, there is a need to design dynamic and environmental models of robot systems before operation, and real-time calculation is necessary. Therefore, previous work has the following drawbacks:

- System architecture is centralized. If the main computer stops working, the fault diagnosis function becomes invalid as well.
- Systems need environmental models, robot kinematic models, and real-time calculation.
- It is difficult to apply to currently-operated robot systems.

To increase the reliability of a robot system, a novel fault detection and diagnosis methodology that considers the above drawbacks must be developed. To do so, our research focused on the following features:

- Decentralized system architecture
- Model-free fault detection and diagnosis methodology
- Attachable system for currently operating robot systems

In this study, a distributed cooperative fault diagnosis method was proposed. This method focused on fault diagnosis of electrical components in the robot system's body because robot systems typically consist of various electrical components. Although robot systems also have mechanical components, this research considered only electrical components, such as embedded computers, motor controllers, and motor drivers. To realize distributed fault diagnosis, the implementation of a small diagnostic device, called a *diagnoser,* in every component was proposed.

The rest of the paper is organized as follows. Section 2 describes the previous approaches. Section 3 is an overview of the proposed distributed cooperative fault diagnosis method. Section 4 provides details about the conditions of the computer simulation experiments and results. Section 5 contains concluding remarks.

2. PREVIOUS APPROACH TO FAULT TOLERANCE

Until now, numerous fault detection and diagnosis methods have been proposed. However, fault detection and diagnosis are elements of fault tolerance technology. It is difficult to define each technology independently, and the techniques overlap one another. To utilize previous research, attention was not only paid to fault detection and diagnosis but also to related research on the fault tolerance of a system.

2.1 Faults in Robot Systems

Okina et al. defined the fault of computer systems as the "difference between realizable and required function" [6]. A robot system has outputs, such as action and behavior, based on inputs, such as tele-operated commands and environmental information. In this study, based on the definition of the fault by Okina et al., the fault of the robot system was interpreted as that occurring when the robot system gives an unexpected response to the input commands or information.

On the other hand, a robot system is constructed from many types of electrical and mechanical components; the robot system cannot be down in all components' broken cause with exception of explortion and radiation effect. Therefore the fault of a robot system is defined such as broken of the some components. In detail, based on the fault definition by Okina, the fault state of a robot system emerges when there are unexpected outputs from the internal components in the robot system.

In addition, as the basic approach, it was assumed that the faulty component was the electrical component in the robot body without sensors, such as a camera, encoder, or measurement sensor for environmental information, because input information from the environment was unknown or unexpected based on the above fault definition.

2.2 Typical Approach

As mentioned above, many kinds of fault detection and diagnosis methods for robot systems have been discussed, such as model-based, signal–based, and model-free approaches. There are traditional techniques for fault detection and fault diagnosis [4], [5]. For example, outside the robot system domains, Lefebvre proposed an on-line fault diagnosis method using partially observed petri nets [9]. A petri net is effective for the representation of the state of a system; however, this method requires a model of the system, as is the case with model-based approach.

Okina et al. proposed a signal-based fault diagnosis method [6]. However, this type of system is centralized and requires voltage and current values when the robot system is fault-free. Moreover, this system needs invasive sensors, such as current sensors, to observe the current state.

In the model-free approach, fault tolerance utilizing learning algorithms is a representative technique. Liu et al. proposed a system using the credit assignment fuzzy cerebellar model articulation controller (FCA-CMAC) neural network for unmanned underwater vehicles [10]. The model-free approach does not require designing environmental and robot models. However, the learner is constructed with a centralized architecture. If the main computer equipped with learning mechanisms is broken, the fault diagnosis function is also inoperative.

2.3 Multi-agent Approach

In the fault tolerant domain, Parker proposed ALLIANCE, which is a multi-robot approach for redundant robot systems [11]. ALLIANCE realizes redundancy using cooperative control of teams of robots. It can detect the faults of one robot, and allow other robots take over the lost function to complete the required task. However, it cannot diagnose the cause of each fault. Christensen et al. proposed fault-tolerant swarms of robots [12]. This research involved a multi-agent approach and focused on the methodology and protocols for the recovery of robot swarms. However, it also cannot diagnose the cause of the robot fault.

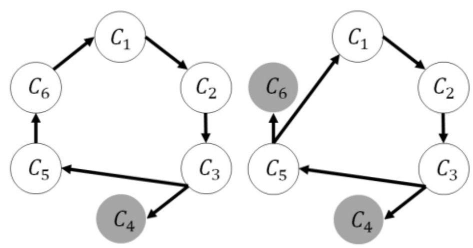
2.4 Computer Simulation Approach

In recent years, run-time fault detection methods based on the comparison of simulated and real robot behavior have been proposed [13][14]. This is similar to the model-based approach because environmental and robot models are designed before running the system. This method's approach to the run-time mobile robot fault detection leverages high performance computing resources. However, this method needs highly accurate environmental models, robot kinematics, and high accuracy observation of robot motion. It is difficult to implement in complicated real-world robot systems.

2.5 Distributed Approach To Fault Diagnosis

In the fault diagnosis of computer domains, distributed approaches have been proposed. For example, adaptive distributed system-level diagnosis (Adaptive-DSD) was proposed by Bianchini et al. [15]. This methodology has still been discussed and applied in recent years [16]. Figure 1 (a) and (b) show the concept of Adaptive-DSD. For example, a computer network system that has six computers connected by communication is shown in Figure 1 (a). Black arrows indicate corresponding relationship of diagnosis. In this case, in a fault-free situation, computer C_n works as an autonomous distributed network. If C_4 is broken, C_3 can detect the fault of C_4. Then, obtained information about the fault of C_4 is shared using the communication among the computers, and C_3 changes the diagnosis target and communication path. In doing so, the system can retain the communication network and diagnostic function. After that, if C_6 is broken (Figure 1 (b)), C_5 can also detect the fault of C_6, and C_5 changes the communication path to C_1.

This concept may contribute to this research; however, a concrete diagnosis method is not defined, and Adaptive-DSD is assumed for usage in computers.



(a) (b)

FIGURE 1: A concept of Adaptive-DSD. (a) shows that one computer is broken. (b) shows that two computer nodes are broken, and the computer prior to the broken computer changes the diagnosis target to maintain the communication loop.

3. PROPOSED METHOD

To determine the fault tolerance of robot systems, one must go beyond previous research and the three keywords indicated in Section 1, which are “decentralized,” “model-free,” and “attachable.” Hence, in this study, a distributed cooperative fault diagnosis method (DCFD) is proposed.

For this method, the following assumptions were made. A diagnostic device called a diagnoser observes the state of a component, such as its input-output signals, to detect a fault. Based on Adaptive-DSD, the result of the diagnosis can be shared using the wireless communication capabilities in each diagnoser. Finally, all diagnosers can obtain the overall diagnosis results.

3.1 Model of Component

In this study, for basic research, we assumed a simplified model for each internal electrical component, as shown in Figure 2. Component C_n has an input and output (Figure 2), and C_n outputs the same input value as in the following equation:

$$C_n(u) = \begin{cases} 1 & (u = 1) \\ 0 & (u = 0) \end{cases} \tag{1}$$

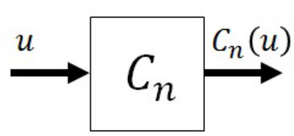


FIGURE 2: Simplified model of internal electrical component for simulation.

3.2 Faults in Robot Systems

Diagnosers can observe the input and output signals of their corresponding components (Figure 2). Before diagnosis, diagnosers learn the input and output signals of the corresponding components using a learning algorithm. When diagnoser D_n diagnoses C_n, D_n observes the signals of its corresponding component and compares the observed signal with its knowledge of the expected signal, as shown in the following equation.

$$D_n(u, C_n(u)) = \begin{cases} \text{Fault\_free} & (C_n'(u) = C_n(u)) \\ \text{Fault} & (C_n'(u) \neq C_n(u)) \end{cases} \tag{2}$$

Here, u is the input signal of the component. $C_n(u)$ is the output signal of component. $C_n'(u)$ is the expected output of the component, which is calculated from u by the diagnoser using the learned input-output signal pair data. If the expected output value, $C_n'(u)$, is different from the observed output value, $C_n(u)$, the component is in a fault condition. In contrast, if $C_n'(u)$ is equal to $C_n(u)$, the component is fault-free.

3.3 Distributed Cooperative Fault Diagnosis

The component architecture shown in Figure 3 was assumed. In this figure, three components and three diagnosers are in the robot body, and the components are connected in series. Diagnoser D_n is connected to component C_n. To detect the fault in a component, a diagnoser compares the observed signal and expected output signal, which was pre-calculated by learned data. If the observed signal and learned data are different, the component is faulty. When D_n is broken, other diagnosers (e.g. D_{n-1}, D_{n+1}) observe the state of D_n using an Adaptive-DSD technique [14]. To diagnose the fault of the diagnoser, neighboring diagnosers use signals, such as the heartbeat of the broken diagnoser. Moreover, the corresponding component, C_n, is referred to as a hidden component when the diagnoser, D_n, is broken. This is called a hidden component fault diagnosis problem (Figure 4). The DCFD method can estimate the state of hidden components using communication and cooperation with neighboring diagnosers. This method contributes to the detection of multiple faults, i.e. malfunction of components and diagnosers at the same time.

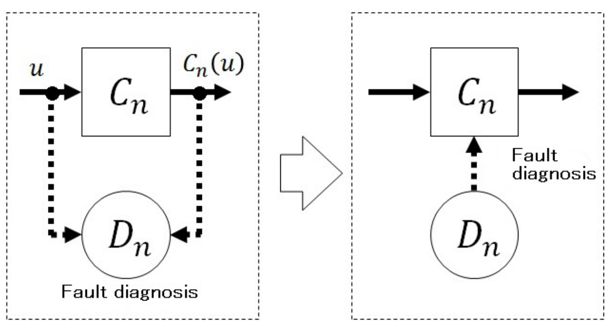


FIGURE 3: Left side figure shows the specified model of a diagnoser with a component, and right side figure shows the simplified model.

In Figure 5, Equation 2 is modified as follows to detect the C_2 fault using observed input signal, v, and output signal, w.

$$D_3\ (v, w) = \begin{cases} \text{Fault\_free} & (C_2'(v) = w) \\ \text{Fault} & (C_2'(v) \neq w) \end{cases} \tag{3}$$

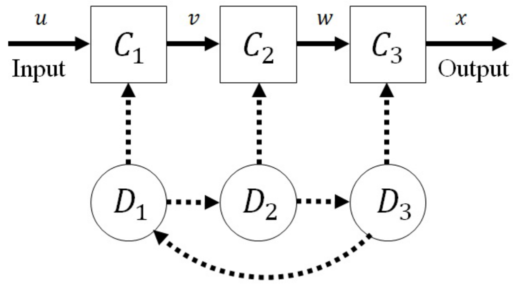


FIGURE 4: An example architecture of electrical components and diagnosers in a robot system. C_n indicates a component, and D_n is a diagnoser. An arrow indicates a correspondence relationship of fault diagnosis.

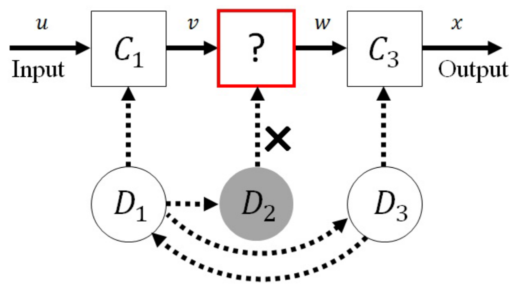


FIGURE 5: Hidden component fault diagnosis problem.

4. COMPUTER SIMULATION

Simulation experiments were carried out to confirm the effectiveness of DCFD in a computer-simulated environment. In this computer simulation, the three simple components and diagnoser model were examined under the following fault conditions.

4.1 Experimental Conditions

4.1.1 Multiple Fault Situations

The components, diagnosers, and their connections were set as shown in Figure 4. In this experiment, it was assumed that each diagnoser obtained the knowledge of the input-output signal pairs from learning before the experiment. In the initial state of the experiment, all components and diagnosers were fault-free. Experimental procedures were as follows:

1) All components and diagnosers run with DCFD.
2) After 60 seconds, D_2 breaks down.
3) After the breakdown of D_2 , confirm that D_1 can detect the fault of D_2 using DCFD
4) After 120 seconds, C_2 breaks down resulting in a hidden component fault diagnosis problem.
5) Confirm that D_3 can estimate the fault of C_2.
6) Finally, confirm that the DCFD method can estimate multiple faults.

The proposed system executed the diagnosis and presented the results at the end of experimental operation. A component outputs the same value when it obtains an input signal. The input signal, $u = \{0,1\}$, alternated randomly every 0.5 seconds. Each diagnoser diagnosed the state of the component once every second.

In this experimental setup, diagnosis results were not deterministic with one diagnosis because when the input signal of the component was '0', the output signal was '0' regardless of whether the component was broken or not. In other words, a diagnoser could detect the fault of components only when the input signal was '1'. In response to this problem, the likelihood of faults was adopted to increase the accuracy of fault diagnosis. Fault likelihood is given by n/N. n

is the number of observed faults, and N is the number of diagnoses performed. Here, N is set to 50.

4.1.2 Three Types of Faults

In the above experimental condition, it was confirmed that DCFD could detect both component and diagnoser faults at the same time. However, various fault behaviors exist in the actual components. In these experimental conditions, three fault types were set as the component fault states (Figure 6). The following fault types were adopted:

- Fault condition A: stable fault with output '0'
- Fault condition B: stable fault with output '1'
- Fault condition C: intermittent fault

Here, a "stable fault" means that the component output for any input is permanently 1 or 0. On the other hand, an "intermittent fault" alternates between the fault state and the fault-free state randomly. Note that fault condition A is the same fault condition as the above experiment.

The observation targets for the result are C_2 and D_2. Other experimental settings are the same as those described in the previous experiment.

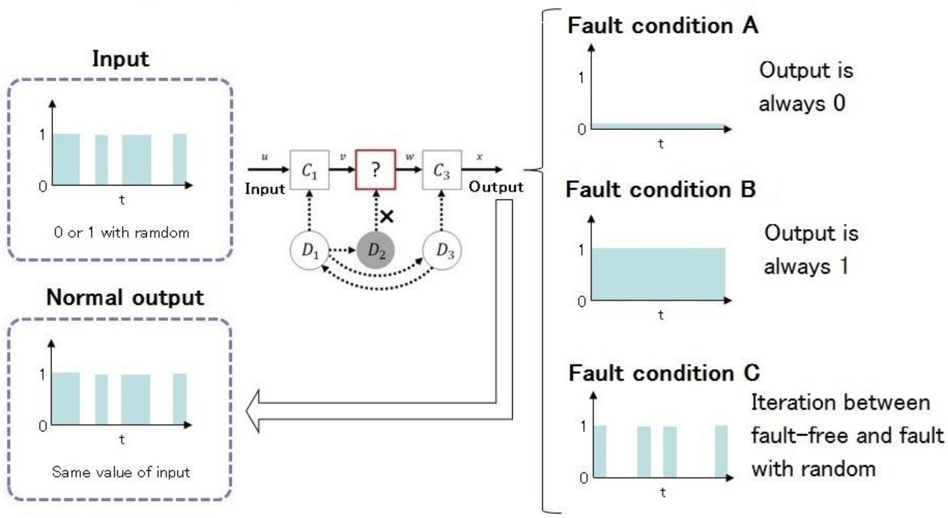


FIGURE 6: Expected output and three types of faults.

4.2 Results

4.2.1 Multiple Faults

The results for the diagnosis are shown in Figure 7. The fault likelihood of D_2 increases and converges to '1' in 60 seconds. This result means that D_1 can detect faults of D_3 when D_2 is broken. In Figure 7, the fault likelihood of C_2 increases after 120 seconds; however, this result of C_2 does not exhibit convergence to one value compared with the likelihood of D_2. The main cause of this phenomenon is that sometimes a diagnoser can detect a fault-free state of C_2 regardless of whether C_2 is broken or not. However, these results indicate effectiveness in solving the hidden component diagnosis problem by using a suitable threshold value.

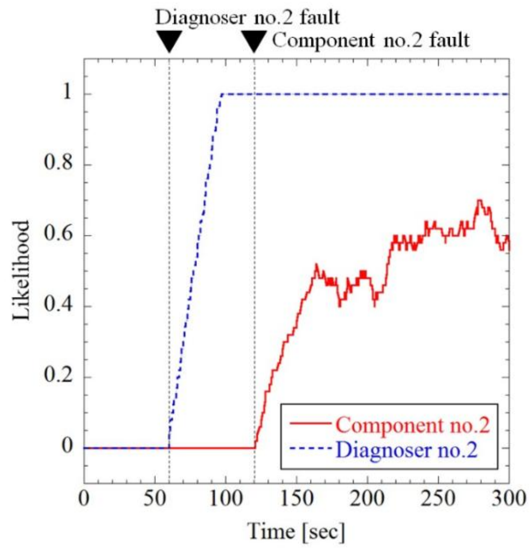


FIGURE 7: Results of experimental condition ''Multiple faults''. Comparison of change of fault likelihood of diagnoser no.2 and component no.2.

4.2.2 Three Type of Faults

The results for the likelihood transition of the three fault types are shown in Figure 8. Note that the result of fault condition A is the same as that of the previous experimental condition. The likelihood of all types of fault conditions increases at 120 seconds, and the likelihood of fault conditions A and B becomes greater than 0.5. However, in fault condition C, the likelihood increases more gradually, and the value does not reach 0.5.

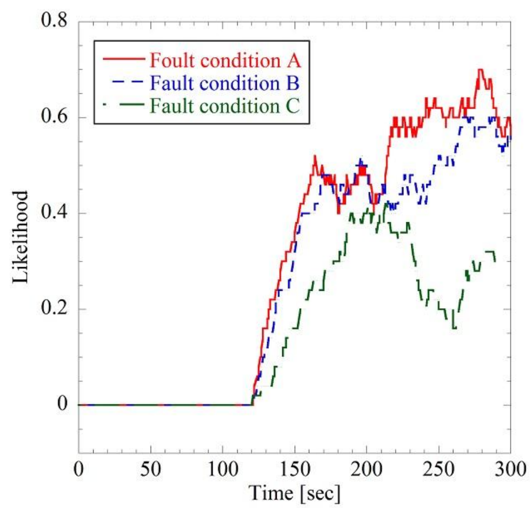


FIGURE 8: Results of experimental condition “Three type of faults”. Comparison of change of fault likelihood in three types of faults.

4.3 Discussion

4.3.1 Accuracy of Fault Estimations

The results for the incorrect estimation case are shown in Figure 9. In Figure 9, the likelihood of fault in the fault-free component is increased in C_1 and C_3. The main cause of this incorrect estimation is a delay between the input and output. In this experiment, a propagation delay of 0.1 seconds was designed between the input and output signals of the component.

In this case, D_1 and D_3 detect the difference between the input and output signals of their respective corresponding components. Therefore, the threshold value, T_h, should be set as $T_h > 0$.

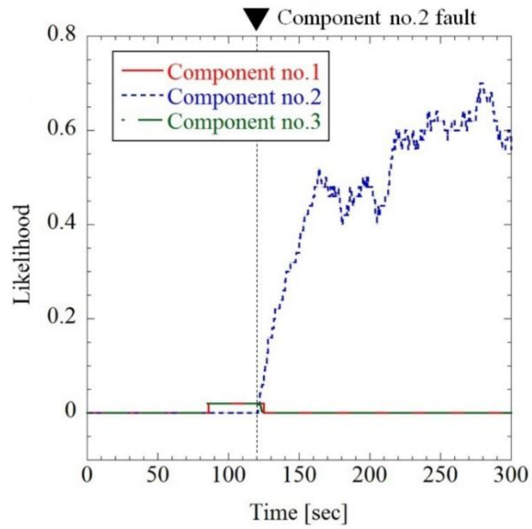


FIGURE 9: Results of experimental condition ''Multiple faults''. Comparison of change of fault likelihood of diagnoser no.2 and component no.2.

4.3.2 Threshold for Fault Detection

The experimental results indicate that the DCFD can estimate the faults of D_2 and C_2. However, the transition of fault likelihood from the DCFD can also be determined, and it is necessary to define a suitable threshold value to detect the fault of a component.

The fault likelihood of the diagnoser becomes ‘1’, and it is easy to estimate the fault because diagnosers estimate the state of other diagnosers based on direct communication, such as heart beat. On the other hand, the change in likelihood of components is unstable. Therefore, a low threshold value should be set to estimate the fault. In fault conditions A and B, fault behavior is simpler than that in fault condition C. Therefore, the threshold value can be set between 0.4 and 0.6. However, in fault condition C, transition of likelihood is lower than other fault conditions. In this case, to estimate the fault of fault condition C, the threshold value should be set at 0.1.

The threshold depends on the system architecture and signal pattern. To apply this threshold to an actual robot system requires a discussion on determining threshold values on a case by case basis.

4.3.3 Limitations of DCFD

Experimental results indicate that DCFD can estimate the fault of C_2 as a hidden component when D_2 is broken. Moreover, if D_1, D_2, or D_3 is broken, DCFD can detect the fault using cooperation among surviving diagnosers. However, DCFD cannot estimate the fault state of a component located at the outer edges of a system when there are multiple faults. For example, in Figure 4, the fault states of C_1 and C_3 cannot be determined when the diagnosers D_1 and D_3 are broken because the surviving diagnoser cannot observe the signals u or x.

5. CONCLUSION

A DCFD method was proposed to diagnose the state of the internal electrical components in robot systems. Simulation experiments were performed under simplified conditions. The experimental results suggest that the proposed method can detect both diagnoser and hidden component malfunction. This result indicates that the DCFD method has the potential to solve the hidden component fault diagnosis problem.

In future, as described in Section 4.3.3, investigation of an estimation method for faults occurring in components located at the outer edge of a system is required. Once complete, the effectiveness of the DCFD method will be demonstrated in more complex and dynamic situations. In the current simulations, components and diagnosers are represented by simplified experimental models. Future experiments will involve extending this method to actual robot systems.

6. ACKNOWLEDGEMENT

This work was funded by Tough Robotics Challenge, ImPACT Program of the Council for Science, Technology and Innovation (Cabinet Office, Government of Japan), and was supported by JSPS KAKENHI Grant-in-Aid for Research Activity Start-up (15H06102).

7. REFERENCES

[1] R.R. Murphy. “Trial by fire,” IEEE Robotics & Automation Magazine. vol. 11, pp50-61, 2004.

[2] K. Nagatani, S. Kiribayashi, Y. Okada, S. Tadokoro, T. Nishimura, T. Yoshida, E. Koyanagi and Y. Hada. “Redesign of rescue mobile robot Quince,” in Proc. 2011 IEEE International Symposium on Security, and Rescue Robotics (SSRR), 2011, pp. 13-18.

[3] J. Carlson and R.R. Murphy. “How UGVs physically fail in the field,” IEEE Transactions on Robotics, vol. 21, pp. 423-437, 2005.

[4] Z. Gao, C. Cecati and S.X. Ding. “A survey of fault diagnosis and fault–tolerant techniques– Part I: Fault diagnosis with model-based and signal-based approaches,” IEEE Transactions on Industrial Electronics, vol. 62, pp. 3757-3767, 2015.

[5] Z. Gao, C. Cecati and S.X. Ding. “A survey of fault diagnosis and fault tolerant techniques - Part II: Fault diagnosis with knowledge–based and hybrid/active approaches,” IEEE Transactions on Industrial Electronics, vol. 62, pp. 3768-3774, 2015.

[6] S. Okina, K. Kawabata, T. Fujii, Y. Kunii, H. Asama and I. Endo. “Self-diagnosis system of an autonomous mobile robot using sensory information,” Journal of Robotics and Mechatronics, vol. 12, pp. 72-77, 2000.

[7] I. Eski, S. Erkaya, S. Savas and S. Yildirim. “Fault detection on robot manipulators using artificial neural networks,” Robotics and Computer-Integrated Manufacturing, vol. 27, pp. 115-123, 2011.

[8] M. Schlechtingen and I.F. Santos. “Comparative analysis of neural network and regression based condition monitoring approaches for wind turbine fault detection,” Mechanical systems and signal processing, vol. 25, pp. 1849-1875, 2011.

[9] D. Lefebvre. “On-line fault diagnosis with partially observed petri nets,” IEEE Transactions on Automatic Control, vol. 59, no. 1919-1924, 2014.

[10] Q. Liu, D. Zhu and S.X. Yang. “Unmanned underwater vehicles fault identification and fault-tolerant control method based on FCA-CMAC neural networks, applied on an actuated vehicle,” Journal of Intelligent & Robotic Systems, vol. 66, pp. 463-475, 2012.

[11] L.E. Parker. “ALLIANCE: An architecture for fault tolerant multirobot cooperation,” IEEE Transactions on Robotics and Automation, vol. 14, pp. 220-240, 1998.

[12] A.L. Christensen, R.O. Grady and M. Dorigo. “From fireflies to fault-tolerant swarms of robots,” IEEE Transactions on Evolutionary Computation, vol. 13, pp. 754-766, 2009.

[13] A.G. Millard, J. Timmis and A.F. Winfield. “Run-time detection of faults in autonomous mobile robots based on the comparison of simulated and real robot behavior,” in Proc. 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014), 2014, pp. 3720-3725.

[14] Z. Gao, J. Jia, J. Xie, W. D. Toh, P. Lin, H.Lyu, D. Julyanto, C.S. Chin and W.L. Woo, “Modelling and simulation of a 12-cell battery power system with fault control for underwater robot, ” in Proc. of 2015 IEEE 7th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM) , 2015, pp.261-267.

[15] RP. Bianchini Jr and RW. Buskens. “Implementation of online distributed system-level diagnosis theory,” IEEE Transactions on Computers, vol. 41, pp.616-626, 1992.

[16] S. Kelkar and R. Kamal. “Adaptive fault diagnosis algorithm for controller area network” IEEE Transactions on Industrial Electronics, vol. 61, pp. 5527-5537, 2014.