Fault Tolerance for Complex System

Chan Shi Jing[1, a)], Dwi Pebrianti[1, b)], Mohd Fadhil Abas[1,c)]
[1]*Faculty of Electrical & Electronics Engineering,*
University Malaysia Pahang Campus Pekan, 26600 Pekan,
Pahang, Malaysia.
E-MAIL: [a)]chanshijing@gmail.com, [b)]dwipebrianti@ump.edu.my, [c)]mfadhil@ump.edu.my

Luhur Bayuaji[2. d)]
[2]*Faculty of Computer Science and Software Engineering,*
University Malaysia Pahang, 26600 Pekan,
Pahang Darul Makmur, Malaysia.
E-MAIL : [d)] luhurbayuaji@ump.edu.my

Abstract—Fault Tolerance (FT) enables system to continue operating despite event of failures. Therefore, FT is served as a backup component or procedure that can immediately play its role without incurring any service lost. FT exists in many form, where it can either be the software or hardware or both the hardware and software. Fault Tolerance is an umbrella term for fault detection, fault isolation, fault identification and fault solving for all types of fault. For the purpose of visualizing the fault detection and isolation process, a two wheel robot is used in this study as a representation of a complex system. The aim of this research is to construct and design a new Fault Tolerance algorithm. The proposed method for the fault tolerant is aimed to solve the fault in actuator or in the sensor by resetting and adjusting it to the correct position. The technique used is able to recognize the different fault from its data condition through the system sensors or actuator. This combined method is served as an innovative solution to identify a fault and solve it immediately.

Keywords—Kalman filter; Artificial Neural Network; Fault tolerant; Fault detection; Fault Isolation

1. Introduction

Fault tolerance is a subsystem of the control system, which serve as a condition solution to solve the fault or failure (hardware or software) in the system. Most of the papers have been researching on how to identify the system fault, including failure of the actuator and sensor as well as the error of the system. Some methods proposed to solve the situation either through the software or the hardware. The software solutions are [1] the FSM method to identify the type of fault, [2] the GLRT mathematical method to identify the fault, [3] artificial neural network, [4] fault decision method, and [5] Fault-Aware Code Transformation For Sensor Network (FACT). On the other hand, hardware method is [6] the Advanced Avionics Fault Isolation System (AAFIS) concept utilizes bit (build in test) and etc. Most of fault systems focus on identifying the fault of the system and thereby the solutions stop upon fault identification. The fault that has been identified would require human to take action to fix the system. Therefore, the question here is why don't algorithms generate self-identification and self-correction on system fault.

Most researchers are using the FSM method for the system communication including, [8] Independent observer to detect fault in communication system as well as [9] multiple observers into the one of fault tolerant to identify if it is capable of detecting the execution error of an ISO transport protocol even when a subset of the observers is faulty.

Another method proposed is utilizing the Petri nets including rollback recovery with check pointing, recovery blocks, N-version programming, and conversations [10]. This algorithm automatically constructs the Petri net models using parameterized subnet primitives. The numerical results clearly illustrated the applicability of the proposed models, where [11] dynamic fault-tree modeling techniques is capable for handling high levels of reliability frequently employ high levels of redundancy, dynamic redundancy management, and complex fault & error recovery techniques. This is also known as a fault-tolerant parallel processor, [12] a Markov chain of the fault-tolerant software system.

In this paper, a self fix algorithm is proposed to generate the solution in faulty condition which will be tested using a Two Wheel Mobile Robot (TWMR). It should be noted that it can only work well in the condition of non-severity of failure. This

TWMR is mimicking the working principle of ground vehicles such as car and bus. Therefore, the study will support the safety matters in transportation industry [7]. Furthermore, the result obtained from this study will be able to lead for better performance of complex system. Once the decision making process of the FT system obtains a precise value of fault parameter, it will undergo faster processing tie and reveal result regarding the failure position as well as the severity of failure precisely.

The concept of a state FSM as an observer, which is capable of detecting specified type of faults in minimum time.[7] In this paper we will focus on a very simple technique on the solution solving for the faulty condition in the system. We transform the FSM from an observer based technique towards a solution based technique. In this condition the FSM subsystem is capable to order or instruct the system to work under the possible fault in the system.

In the next section we will review the concept of FSM to show how FSM model serve as the solution solving and indicate our desired target to decompose the observer fault class from the Adaptive Neural Network (ANN) into a set of smaller FSM solution capable of jointly solving faults. We also provide a virtual alternate procedure capable of operating in real time that eliminates synchronization problems arising in the original method.

2. MODEL APPROACH

Our research objective is to develop an algorithm that is able to solve the fault system automatically during their operation life. The modeling fault tolerant software also considers the impact of the performance on the system. Therefore, only relevant result for a given period of time is obtained for the stable and reliable software condition. [13] This section briefly describes the various fault-tolerant techniques in use.

There are several condition faults tolerant approvals used in this paper to avoid the happening of non-responding error occurred in the system. The 4 main type of responding approaches taken by the system are:

1. *Recovery* type: to encapsulate each critical function of a process state [9]. In the event of a failure, a checkpoint allows the robot to reverse its motion to the previous condition. Therefore, storing checkpoints of a process are same nodes where the executing process can be conducted to recover from the failures.
2. *Reset* type: this scheme is activated when there is a critical functional fault in the system and it is unable to be solved by the other approach.
3. *Replace* type: to replace the available sensor that still working well in the robot for restoring the system function.
4. *Avoid* type: to avoid the fault that may be happening in the actuator by resetting the actuator. The following section will illustrate how the voiding type failure happens in the complex system.

Our approach of modeling the reliability of the fault tolerant software is more general than previous one as it combines information on the software structure as well as the reliability growth of the components. An important property of the FSM model is the transformation of a stable and reliable FSM toward the system solution solving needed model to handle reliability growth. This is particularly relevant as it allows the reliability growth of fault tolerant software system to be modeled from the reliability growth of its components. The transformation is therefore based on the interpretation of the FSM model.

Based on our research, this approach is one of the easiest in the process of modeling of fault tolerant software system because it is systematic as electronic FSM methods.

3. METHODOLOGY

In this research, the residual information was generated by the comparing the value of the actual and the filtering value. This information will then be fed into the ANN to identify the type of fault that might exist. If fault was detected, the system will do further adjustment using the FSM solution solving technique.

In ANN techniques, some models of the system are used to decide regarding the type of fault occurred on the system. The system model may simulate and allow the user to obtain valid source information about the fault behaviors. It also involves the use of simplifying approximations and assumption within the simulation, hence allowing user to verify the fidelity and validity of the simulation outcome.

This algorithm detects potentially faulty sensor or actuator in the Two Wheel Mobile Robot and automatically generates the solution based of each faulty condition. During the faulty of the TWMR, the type of fault behavior generated will compare the solution that have been set or stated in the system, giving instruction to the robot to adjust its angle to reach the destination.

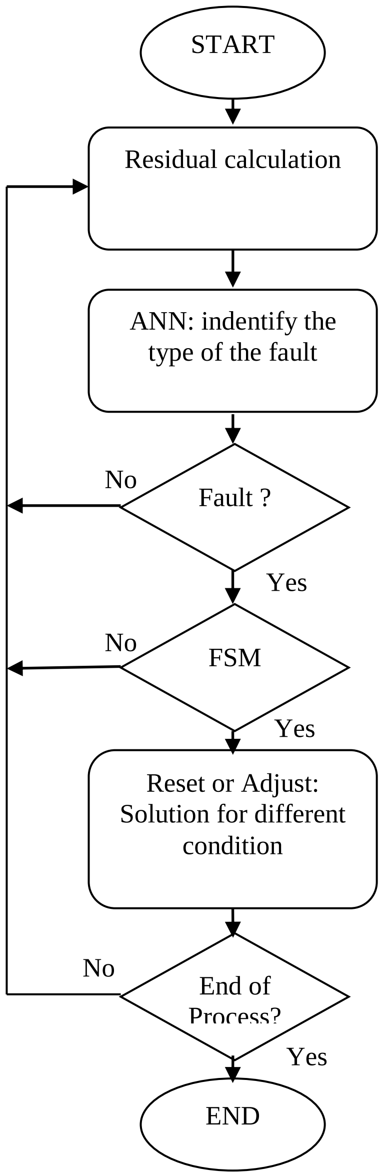

FIGURE 1: Flow planning of FT

4. ARTIFICIAL NEURAL NETWORK (ANN)

In the system we assume that the fault classification of the system is fed by the ANN (Artificial Neural Network). The fault diagnosis scheme is based on a group of observation where every residual is exclusively associated to with sensor. Each residual has a 3 input, the input signal u of the system and the output of its corresponding sensor [10].

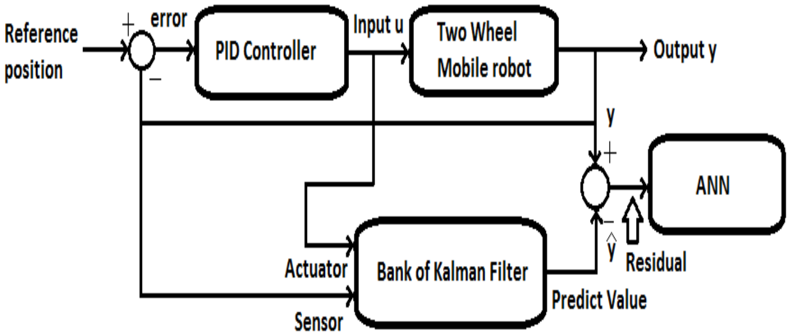

FIGURE 2: Block Diagram of the Complete System

In this system, the reference position is set such that TWMR work to the destination. The reference position is compared to the position produced due to an error and will be fed to the PD controller to generate an input u to the system. The Kalman filter is used as an observer to predict the value of the output value of the sensor and the actuator. It will then be compared to the output y to generate the residuals.

The fault classification is defined as shown in the Table I below:

Table I: Fault Definition

Output of the ANN system	Fault classification
[1 0 0 0 0 0]	Fault 1 (actuator 1)
[0 1 0 0 0 0]	Fault 2 (actuator 2)
[0 0 1 0 0 0]	Fault 3 (Sensor 1)
[0 0 0 1 0 0]	Fault 4 (Sensor 2)
[0 0 0 0 1 0]	Fault 5 (Sensor 3)
[0 0 0 0 0 1]	Fault 6 (Sensor 4)

The ANN structure and the learning process are defined by the following parameters;

Table II: The Parameters of the ANN system

Parameter	Number
Input vector	: 18
Output vector	: 06
Number of the layer	: 02
Number of the neurons in the first layer	: 18
Number of the neurons in the second layer	: 12
Number of the neurons in the last layer	: 06
Momentum	: 0.5
Learning rate	: 0.9

In this work, our scheme uses BP to design the network identification fault system. It consists of three layers in the structure, as shown in Figure 3.

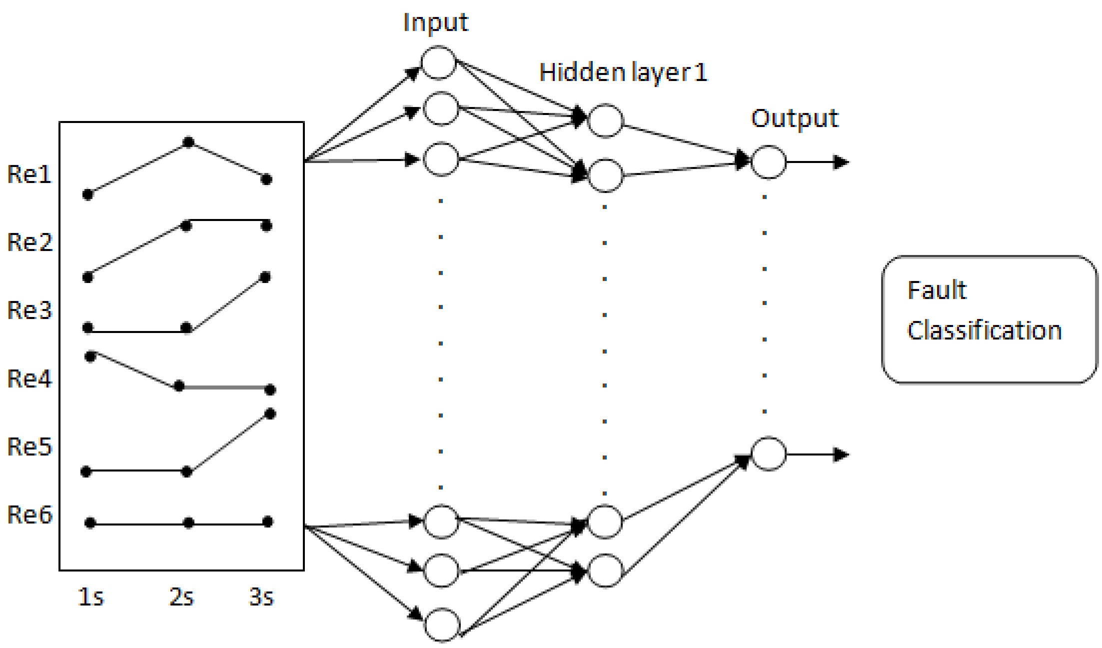

FIGURE 3: The ANN model

The ANN used in this study is the back-propagation training process to find the appropriate weight for each network. $Re1$, $Re2$, $Re3$, $Re4$, $Re5$ and $Re6$ are the residual of the system, while Fault class is the fault identification of the actuator and sensor.

5. FSM SUBSYSTEM

The system is observed over a time interval and the relevant solution case that is generated by the system is recorded. The set of the relevant solution is treated and prepared by the system.

The model of the system is shown below.

Fault occur in the system

Solution for the system

	Fault 1	Fault 2	Fault 3	Fault 4	Fault 5	Fault 6
S0	0	0	0	0	0	0
S1	1	0	0	0	0	0
S2	0	1	0	0	0	0
S3	0	0	1	0	0	0
S4	0	0	1	0	1	0
S5	0	0	1	1	1	0
S6	0	0	1	1	1	1
S7	1	1	1	1	1	1

FIGURE 4: FSM Solution and Fault

The value 1 inside the box table is responding as Yes condition showing fault occurred, while value 0 represent no that fault in the condition.

The solution for the system is stated in the form of the FSM. The rows of the Fault occur in the system correspond to the faults occur in the system. Its columns represent the state solution; the element (i, j) is 1 if the fault i forces a transition to the state. The i is the solution state of the system while the j is the fault that might happen in the system.

The number of the resulting state FSM is depended on the number of the size on the probability of the combination faulty that might be occurred in the system. The computation complexity of the error solution finding is considerably reduced. [4]

In this section to find the number of combination, we recall the basic of the FSM algorithm for N states (number of fault), with $N = 2^{m-1}$, when N is not a power of 2, a probability of the fault m combination number in order to make a power of 2. The decomposition algorithm can be generated reversely by $m = log_2 N$. The elements of state s_N are the order $\{s_1, \dots, s_{n+1}\}$. The algorithm generates partitions that are symmetric.

Example:
Algorithm 1 (only 2 faults can be detected)

$$s_N = \{s_1, \dots, s_{n+1}\}.; \quad s_N = \{s_1\}$$
$$\text{For } N = 2^{m-1}$$

End

A formal composition rule of FSM model in now derived under the following main signature:

Signature 1: the faults can occur in any second in the system while its diagnostic procedure is completed running.
Signature 2: the faults can occur or replace instantly in any second once a fault has solved, the diagnostic procedure is completed.

Technically, this signature is useful to limit the dimension of the FSM generated by the composition of elementary FSM model. Once a fault has been detected, the system is usually reconfigured in the state solution given in the system to maintain in safety standards. In the new system configuration, any further fault has to be viewed as non simultaneous fault occurring to a safe configuration.

6. RESET AND ADJUST PROCEDURE

The solution condition is sequential looped by solution state of the system, and the solution state to solution state is passed through by the fault class that is assigned by the ANN. We can assume that the higher the types of solution condition in the system is the higher the accuracy of the efficient work done by the robot correctness.

The method to develop the solution is based on each faulty condition:

Solution 0: running in the no fault condition the actuator 1 and actuator 2 are in the forward condition
Solution 1: faults in the actuator 1 of the mobile robot are common actuator fault. The changes in speed or the turning of the actuator have been extensively studied.
Solution 2: running during the actuator 2 is in fault condition. The changes in speed or the turning of the actuator have been extensively studied to avoid the non-linear moving of the robot.
Solution 3: running in the fault of the sensor 1 by replacing the sensor 3 as the front side detect distance sensor and the running mode is in the reverse condition.
Solution 4: running in the fault of the sensor 1 and sensor 3 by replacing it with sensor 2 to detect the front side distance condition. In this condition, there is no detection working in the moving condition. Detecting in the time every moving 1 to 1 cm, and also depend on the distance that is detected by the sensor 2.
Solution 5: running in the fault of the sensor 1, sensor 3 and sensor 2 by replacing it with sensor 4 to detect the front side distance condition. In this condition, there is no detection working in the moving condition. Detecting time every moving 1 to 1 cm, and also depend on the distance that detected by the sensor 4.
Solution 6: free sensor mode to ensure the actuator 1 and actuator 2 are running properly.
Solution 7: critical condition stops working for the system.

In this section we have considered 8 types of solution to solve the 6 type of possible fault that might occur in the TWMR. In the next section, we will describe the scenario of the voiding method fault elimination method that has been applied to the actuator where there is 10 percent of the fault overshoot error on the system.

7. RESULT AND DISCUSSION

In this study, an actuator fault is implemented abruptly after the 4th second of the simulation time by 10 percent to the wheel 1 motor. This fault can be due to deformations, flatten of the tires, broken, etc. We are using the fault voiding method in the system for the failure that occurs in time 4s. The voiding method is normally used in the actuator fault condition. This is because of no replacement actuator available in the TWMR. Other than that, we also need to consider the percentage and the frequency of the fault occurrence in the robot.

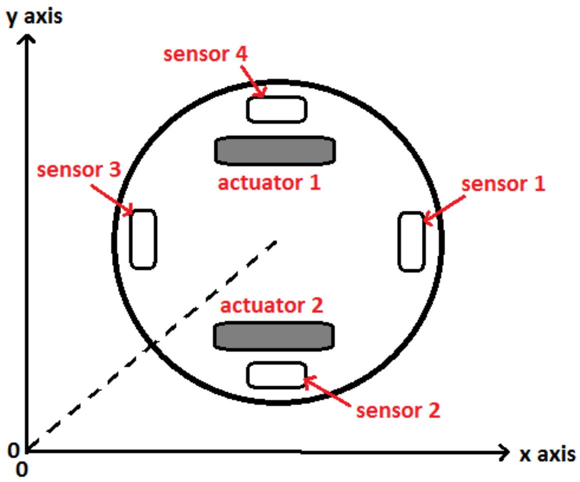

FIGURE 5: Schematic model of Two Wheel Mobile Robot

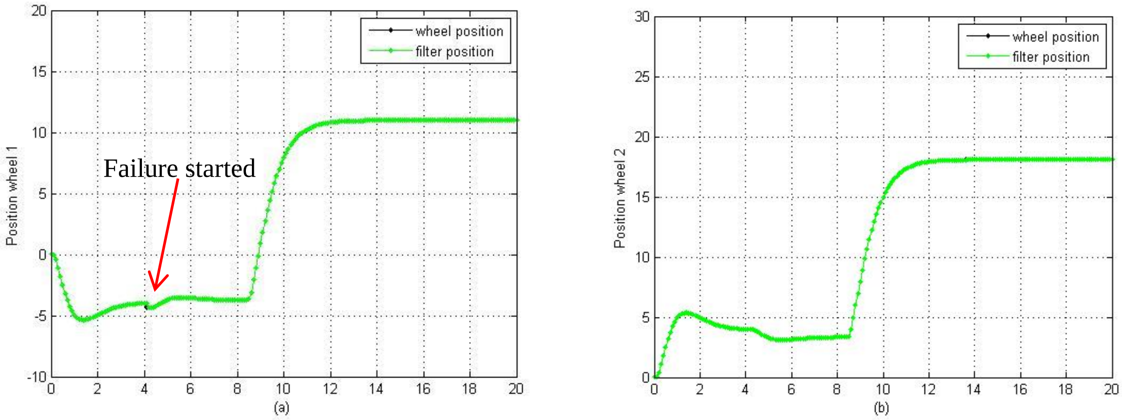

FIGURE 6: The Actuator 1 Fault in 4th Second

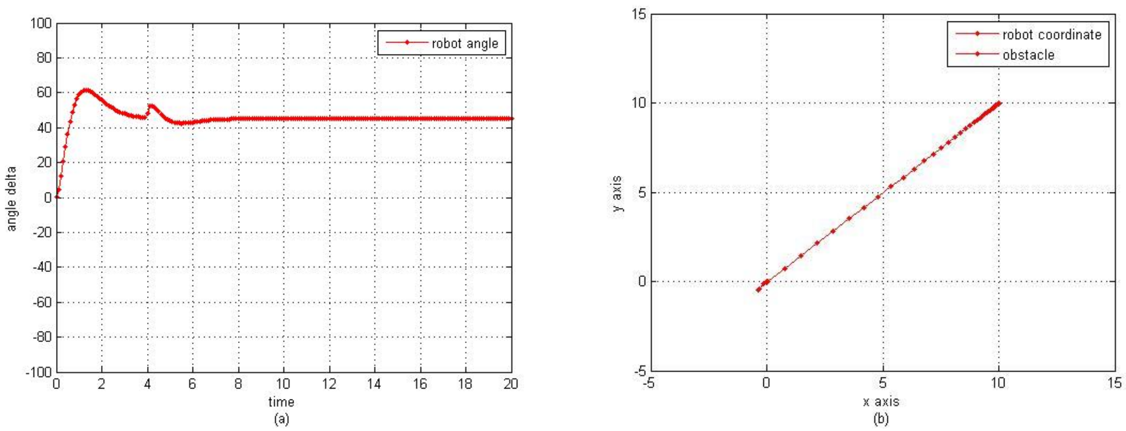

FIGURE 7: The Angle and Map of the Two Wheel Mobile Robot

The entire actuator faults have been tested. At each time t the result of hypotheses would update the status each time. The fault appears on t time moment $t = 4s$ and the out u_t of the sensor is 30% up of its nominal value. Figure 6 represents both the real position (black dot line) and the predicted position (green line).

Figure 7 (a) shows that the angle rotation of the TWMR, angle against time graphs in first 10 second. Figure 7 (b) shows the location of the robot in every 0.1 second, and the initial value started from (0, 0) and end with (10, 10).

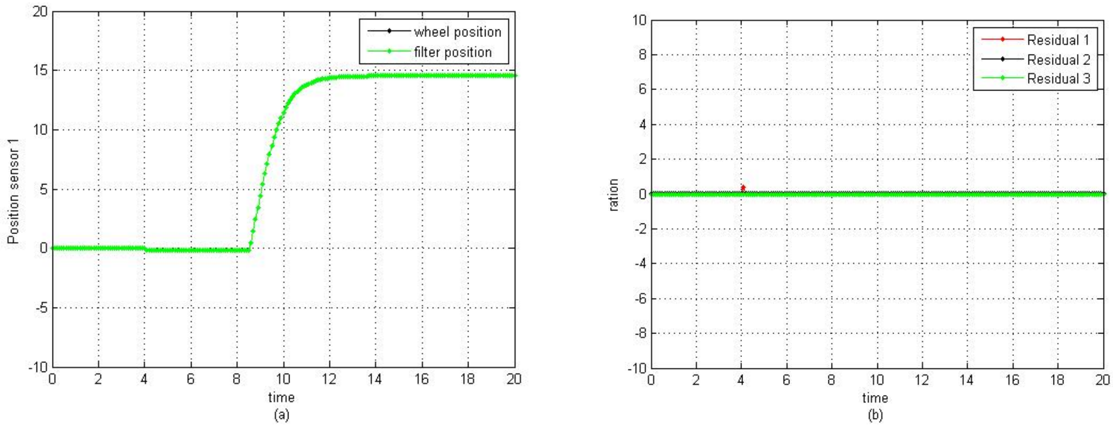

FIGURE 8: Sensor 1 Position Signal and Residual of The Two Wheel Mobile Robot

Figure 8 (a) shows the front sensor range distance against time in first 10 second. Figure 8 (b) shows the three difference types of the resident that have been generated ($Re1$, $Re2$, and ($Re3$/ $Re5$) ($Re4$, $Re6$ can be ignored in this condition)), while fault detected at the 4 second to the reverse condition and at the same time the $Re3$ will replay by $Re5$, and $Re3$ is ignored, .

The experimental result above have been verified where it shows how the algorithm implement the solution and identify the fault of the system. The result reveals that it has been succeed to solve the fault during the time of fault occurrence to in the system. ANN model produced an output value in the $4s$ and has given an instruction to the robot to adjust its angle to pointing to destination.

8. CONCLUSION

We have presented a procedure to construct a set of a fault solution (subsystem) by decomposition of the fault tolerant algorithm which is intended to be integrated into real-time environment. The system is capable to voiding any possible fault occurred in the system. However, some faults of the system cannot be solved perfectly such as correcting the sensor to its initial state. In addition, faults involving timing and operational statistic cannot be handled by the FSM approach discussed here. The research work is currently under simulation condition only using the FSM model approach.

The future research should investigate the solvability of the proposed system on the different type of faults as well as the implementation of this method towards fault accommodation.

ACKNOWLEDGMENT

This works is supported by Ministry of Higher Education Malaysia, under the Fundamental Research Grant Scheme, FRGS **RDU140137**.

REFERENCES

[1] L. Magni; R. Scattolini; C. Rossi " A Fault Detection and Isolation Method for Complex Industrial Systems" *IEEE Transactions on Control Systems Technology,* 2000, vol.30, pp. 860 -865.

[2] D. W. Apley; Jainjun Shi"A fault detection, isolation, and identification technique for complex MISO linear systems" American Control Conference, 1994, vol. 3, pp. 2633-2637.

[3] S. Bouhouche, M. Lahreche S. Ziani and J. Bast "Fault Detection and Monitoring of Length Loop Control System in Pickling Process", 2005, pp. 642 - 647.

[4] C. Lee; R. L. Alena; P. Robinson"Migrating Fault Tree To Decision Tree For Real Timr Fault Detection On International Space Station" Aerospace Conference, 2005 IEEE, pp. 1-6.

[5] L. S. Bai; R. P. Dick; P. A. Dinda; P. H. Chou "Simplified Programming Of Faulty Sensor Networks Via Code Transformation and Run-time Interval Computation" Design, Automation & Test in Europe Conference & Exhibition (DATE), 2011, pp. 1-6.

[6] N. Benowitz; D. F. Calhoun; G. E. Alderson; J. E. Bauer; C. T. Joeckel "An Advanced Fault Isolation System for Digital Logic Advanced Avoinics Fault Isolation System" IEEE Transaction on Computers, 1975, vol. C-24, pp. 489-497.
[7] Puneet Goel, Goksel Dedeoglu, Stergios I. Roumeliotis, Gaurav S. Sukhatme "Fault Detection and identification in a mobile robot using multiple model estimation and neural network"
[8] G. Noubir ; Dept. of Comput. Sci., Swiss Federal Inst. of Technol., Lausanne, Switzerland ; B. Y. Choueiry ; H. J. Nussbaumer" Fault tolerant multiple observers using error control codes" 29 Oct-1 Nov 1996, pp. 84-91.
[9] George K. Fourlas, George C. Karras, Kostas J. Kyriakopoulos, "Sensors Fault Diagnosis in Autonomous Mobile Robots Using Observer – Based Technique" 2015, pp. 49 – 54.
[10] Y. B. Shieh ; D. Ghosal ; P. R. Chintamaneni ; S. K. Tripathi "Application of Petri Net Models for the Evaluation of Fault - Tolerant Techniques in Distributed Systems" 5-9 Jun 1989, pp. 151-159.
[11] J. B. Dugan; S. J. Bavuso; M. A. Boyd "Dynamic Fault-Tree Models for Fault-Tolerant Computer Systems" *IEEE Transactions on Reliability*, 1992, vol. 42, pp. 363-377.
[12] K. Kanoun; M. Kaaniche; C. Beounes; J. C. Laprie; J. Arlat "Reliability Growth of Fault-Tolerant Software" *IEEE Transactions on Reliability,* 1993, col. 42, pp. 205-219.
[13] C. Wang ; Columbia Univ., New York, NY, USA ; M. Schwartz "Fault detection with multiple observers" 4-8 May 1992, vol. 3, pp. 2187-2196.