

Fault Detection and Fault Tolerant System for Smart Factories

Prajakta Jadhav^{*†}, In-Geol Chun^{*†}

^{*}Korea University of Science and Technology (UST), Yuseong-gu, Daejeon, South Korea.

[†]Embedded Software Department, Electronics and Telecommunications Research Institute (ETRI),

Yuseong-gu, Daejeon, South Korea.

{prajakta65, igchun}@etri.re.kr

Abstract— Cyber-Physical Systems plays an important role in smart factories. Nowadays, factories are adopting highly automated environment for more efficient and reliable industrial operations. The industrial systems should operate continuously in this highly automated environment. Therefore, fault detection and tolerant systems has become an essential part of smart factories. Fault detection and tolerance is a challenging task as it needs to deal with uncertain and unpredictable nature of industrial systems. We have developed a pick and place robot using LEGO mindstorms NXT kit, for experimental purpose. Faults occurring in the pick and place robot are detected. In order to remove these faults, we have used neural network methodology. From the results we can conclude that, the proposed system can detect the faults as well as it has capability to reduce the faults up to great extent.

Keywords— Cyber-Physical Systems, fault detection, fault tolerance, neural network.

I. INTRODUCTION

Cyber-Physical Systems unite the cyber world of computing and communication with physical components. Cyber-Physical Systems are advantageous because of its characteristics such as safety, security, adaptability and usability [1]. Cyber-Physical Systems are an important part of the fourth industrial revolution named as 'Industrie 4.0' [2]. The use of Cyber-Physical Systems with production systems gives rise to the 'smart factories'. Smart factories are fully automated cyber-physical production systems which will automatically manage the industrial processes. These smart factories are adopting fully automated environment for reliable and efficient operations. Therefore, requirement of smart industrial systems is increasing day by day [3].

The physical environment in which the industrial system operates is fluctuating therefore, one cannot determine its behavior. On the other hand, the cyber world of computing and communication is not flexible. Therefore, failures can occur in industrial systems, which decrease the productivity of the system. Hence, it is necessary to detect faults in the system to avoid further losses. Designing a reliable, intelligent and efficient system can help to overcome the drawbacks of physical systems. Author In-Geol Chun et. al. have proposed faults and adaptation policy modeling method, to develop a self-adaptive robot [4].

Various types of robots are used for manufacturing and production processes in industrial systems. Some of the examples of robots are pick and place robot, welding robot, painting robots, object sorting robots and so on. Robots with independent decision making capabilities are suitable to operate in smart industrial environment. These robots can work efficiently to increase the speed of manufacturing process and also helps to reduce human errors, resulting in high productivity. Therefore, pick and place robot developed using LEGO mindstorms NXT kit is used, for experimental purpose.

In this paper, we have proposed a fault detection and fault tolerant system for smart factories. By using the proposed technique, we can deal with uncertain and unpredictable nature of industrial systems. A fault analysis model is developed to organize all the possible faults in robotic system. Residuals are generated to detect the faults. Some faults exists in sensors and actuators of robot, which are unsynchronized in nature. But the system should operate continuously hence a fault tolerant system is required. Fault tolerance method using neural network's feed forward and backpropagation techniques is used. We have trained the faulty data using neural network methodology to remove the faults occurring in pick and place robot. Neural network is a self-adaptive, data driven technique with high degree of accuracy.

The rest of the paper is organized as follows; Related work is presented in Section II. In Section III, system modeling, fault analysis model is described. Also fault detection and fault tolerance methodology is discussed, in this section. Case study and results are included in section IV. And, we have concluded this paper along with future work in Section V.

II. RELATED WORK

A rapid development of industries is increasing the complexity of industrial systems. Complex systems may lead to failure and losses. Hence, a lot of research is going on, in the area of fault detection and fault tolerant systems. The purpose of fault detection system is to detect the faults quickly to avoid the system failure. An adaptive threshold generation method for fault detection in mobile robots is proposed by Farzad B. et. al. in [5]. A fault detection mechanism monitors

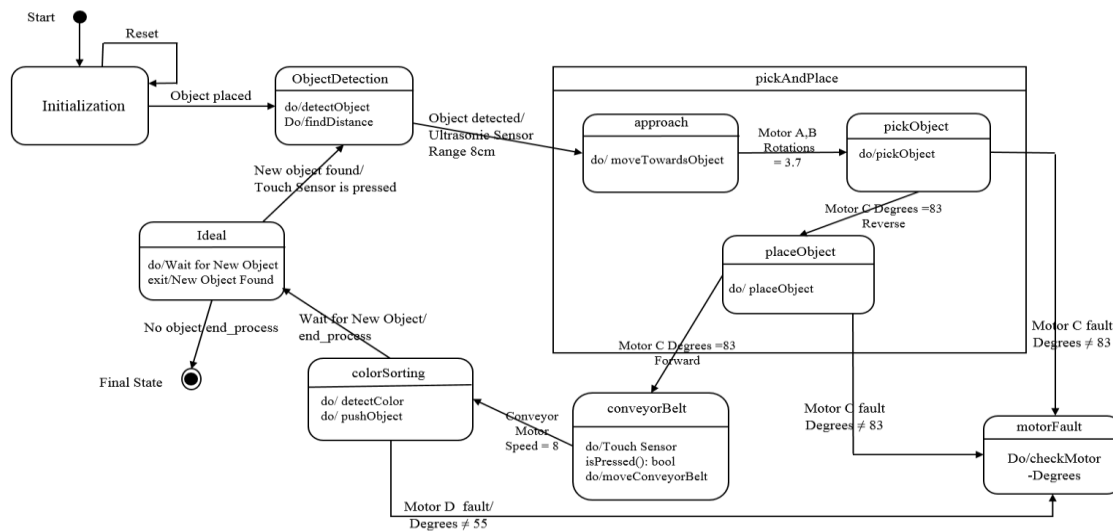


Figure 1: State-chart diagram for pick and place robot using UML

the systems input and output, which will notify the fault detection by using alarm [6].

Actuator faults are considered most challenging, as it affects the system performance by improper actuator movements [7]. Most of the research on actuator fault tolerant systems is based on linear systems [8], [9], and [10]. Authors Duan et. al. discussed fault detection and diagnosis along with fault tolerant control techniques [11].

There are several neural network based fault tolerant techniques developed for mobile robots. A path planning algorithm developed using neural network and sensor data is presented in papers [12], [13], and [14]. They have trained the robot intelligently, to avoid obstacles and find path. Jasmin Velagic et. al, designed neural network based controller for motion control of robot using backpropagation optimization algorithm [15]. Michael Hofbaur et. al., used model based reasoning to improve the robustness of robotic systems [16]. Steinbauer Gerald et. al., mentions that model-based approach not only formalizes the fault diagnosis problems but also ensures that the results are accurate [17]. A fault detection technique is experimented on welding robots, which shows that to predict the faults in robotic systems neural network technique can be used, in [18].

Self-adaptive systems must have ability to change its operation and adapt the new policies automatically [19]. A conventional feedback control systems results in poor performance as well as malicious behavior of actuator and sensors [20]. To overcome these drawbacks new approaches are required, which can tolerate the system faults along with its performance stability. Therefore, we have used neural network approach for fault tolerance in this paper.

This paper is focused on development of fault detection and fault tolerant system, for robotic systems. For case study purpose, pick and place robot is used. The faults in this robot are detected and tolerated. The results demonstrate the

effectiveness of fault tolerant approach using neural network on pick and place robot.

III. FAULT DETECTION AND FAULT TOLERANT SYSTEM

Even today, the most efficient robot system may not perform its operations perfectly, that anyone could imagine. An occurrence of fault drastically changes the robots performance, which may lead to some disaster, or human injuries. This is because of lack of reliability of a robotic system to perform its operations in the presence of faults. Therefore, a fault detection and fault tolerance methodology needs to be developed. In this section we will have a detailed look at system modeling, fault analysis model and fault detection and fault tolerance methodology.

A. System Modeling

Designing an intelligent robotic system is a complex process; therefore a powerful model is required. We have used Unified Modeling Language (UML) to design a robotic system.

To present the dynamic behavior of pick and place robot, state-chart diagram is used. In pick and place robot, there are six main states viz. initialization, object detection, pick and place, conveyor belt, color sorting and ideal state. The robot firstly initializes the system and all its components. If the system is not initialized properly then, system resets. Later, it goes in object detection state where, robotic system detect the object using ultrasonic sensor. The ultrasonic sensor checks if the object is within the given range. After the object has been detected, the robot goes in pick and place state.

The state-chart diagram is shown in figure 1. In pick and place state, there are three sub-states; approach, pick object, place object. In approach state, pick and place robot approaches towards an object which is detected by ultrasonic sensor. The arm of the robot grabs an object. After getting a

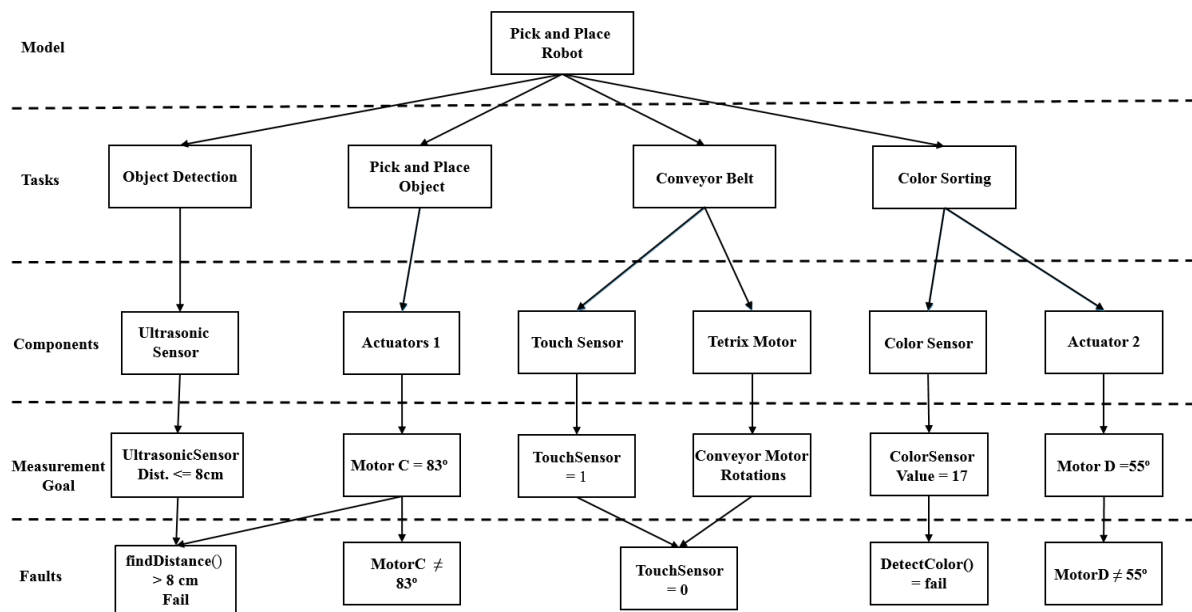


Figure 2: Fault Analysis Model (FAM)

transit instruction to move the motor, the arm places an object on conveyor belt. The transition occurs from pick and place state to conveyor belt state. Conveyor belt state, checks if touch sensor is pressed or not. If touch sensor is pressed, the conveyor motor starts to move. Next, the system transits to color sorting state. In the color sorting state, color sensor detects an object using given color code and sorts it. This is one complete cycle of robot. After completing all these steps, the robot goes in ideal state, where it waits for new object. If new object is detected, the transition occurs from ideal to object detection state. If there is no object then the robot ends the process. This is the main functionality of pick and place robot.

B. Fault Analysis Model

To develop a fault detection and fault tolerant system, it is necessary to organize the possible faults in proper manner. In this section, Fault Analysis Model (FAM) is explained, which determines various faults occurring in pick and place robot. It is necessary to understand how and where the system can go wrong. This helps to improve the operation of the system. Also, this is necessary for robots safety.

There are various types of faults, such as faults in the hardware, software bugs, or some abrupt situations, human errors and so on. It is very difficult to differentiate between the faults. There are several tools such as Fault Tree Analysis (FTA) which analyses the faults in the system using fault tree and graphical representation of the events [21].

We have developed a Fault Analysis Model (FAM), which describes overall system structure including fault conditions occurring in a system. It is a generalized model which can be used for robotic system. Figure 2 shows the Fault Analysis Model for pick and place robot. It is represented using a tree structure. The top event shows model name, for example, in this case, the top event is pick and place robot. The next level

represents the tasks of the system. Hence, it is easy to sort the faults in the system related with particular tasks. Components used in each task such as sensors or actuators are noted below each tasks. Every component is designed to perform some specific operation. Measurement goal represents the goal of every component. In the final stage, the fault conditions are given. If the system cannot reach the given measurement goal then faults can occur in a system. For example, the measurement goal for motor C (as shown in figure 2) is 83 degrees but, if this condition is not achieved then fault can occur in the system.

C. Fault Detection and Fault Tolerance Methodology

In this sub-section, fault detection and fault tolerance methodology is explained in detail. Our designed, fault detection and fault tolerant system, consists of two main modules which are, fault detection unit and fault tolerant system as shown in figure 3. The output data from robotic system's sensors or actuators is given to fault detection unit for detecting anomalies. After detecting the faults, this faulty data is given as an input to fault tolerant systems, where the faults are minimized using neural network technique. The details of fault detection and fault tolerant system are given below.

1) Fault Detection System: Fault detection determines system behavior and its performance. There are several methods of fault detection. A model-based fault detection method is used for detecting actuator faults in pick and place robot. It consists of detection of faults in sensors, actuators by calculating its input and output signals [22].

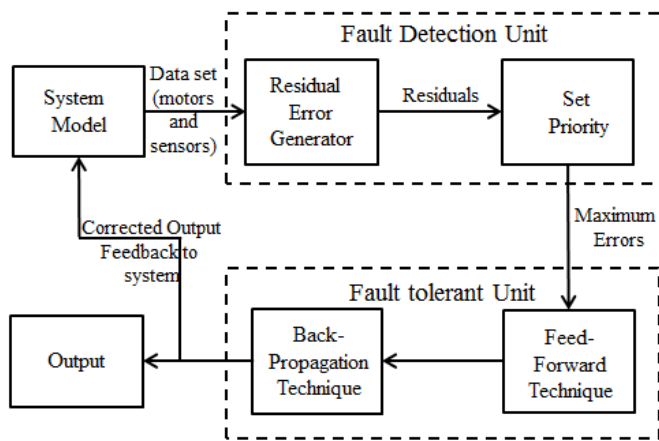


Figure 3: Block diagram of fault detection and fault tolerant system

The fault detection unit, shown in figure 3 measures the inconsistencies between actual output and target output. To determine the faults occurred in the system, residual error is calculated as shown in equation 4.1.

$$r = y - \hat{y} \quad (4.1)$$

Where, r = Residual error
 y = Target output
 $\hat{y}$ = Actual output

Also, total number of errors in the actuators are calculated. And highest priority is set to the actuator with maximum number of errors. The residual error output of the actuator with maximum number of errors is given to fault tolerant system to minimize the faults. Therefore, the actuator with maximum errors will get fault recovery fast. But, if the number of error in the actuators are same, then first come first serve technique is applied to the actuators.

2) Fault Tolerant System: Fault tolerance is the ability of the system to perform its tasks continuously in the presence of faults.

We have used neural network technique to reduce the anomalies present in the robotic system. Neural network is complex system with an adaptive learning technique. It has input, hidden and output neurons. Each neuron is assigned with some random weights. These weights are adjusted accordingly, to achieve desired output. This neural network technique has some advantages, such as adaptive learning, fault tolerance or anomaly detection, real-time operation etc.

First of all, we have trained the neural network with a correct pattern or target output. Neural networks are good at recognizing the patterns. If the pattern is incorrect, the neural network can detect the errors. Also it can be trained with the desired output pattern. By using this technique errors are reduced. Feed forward technique is used, in which input is passed through neurons to get an output. This input is assigned with randomly generated weights. Then all the inputs are summed together with weights assigned to it. Later, it is passed

through activation function and we get the output. The output is calculated by using the equation 4.2. Hyperbolic tangent is used as an activation function (equation 4.3).

$$Output = f(\sum_i i_i w_i) \quad (4.2)$$

Where, f = Activation function
 i = Input
 w = Weights

$$f = \tanh x \left(\frac{e^x - e^{-x}}{e^x + e^{-x}} \right) \quad (4.3)$$

The neural network will check if the output is similar to the target output or not. If the output is similar, then there is no error. But if the output is not same, as target output then there is an error. The backpropagation technique is used to minimize the error. This is the training process. It will update the weights which were previously assigned to the neurons. As the weights changes, the error will either increase or decrease. Neural network will search the point where the error is minimum. For this it will minimize the root mean square of errors in training process (equation 4.4). The output with minimized errors is obtained after this backpropagation technique. This output is fed back as an input to the actuators of pick and place robot.

$$RMS = \sqrt{\frac{1}{n} \sum_i^n (target_i - actual_i)^2} \quad (4.4)$$

Gradient Calculation (equation 4.5) is used to distribute the errors across neural network structure. The random weights which are applied to each neurons needs to be adjusted to minimize the errors and obtain the desired output. Here, η is the overall learning rate, which is set to 0.15 and α is the momentum, set to 0.5.

$$\Delta W_t = \eta \frac{\partial \epsilon}{\partial W_t} + \alpha \Delta W_t - 1 \quad (4.5)$$

Using this fault detection and fault tolerant system, we can minimize the faults occurring in pick and place robot.

IV. CASE STUDY

In this section, we have described the details about implementation of pick and place robot using LEGO Minstorms NXT kit. Also advantages of using neural network are mentioned. Results of fault detection and fault tolerance method are also mentioned in this section.

A. System Implementation

LEGO Mindstorms NXT is widely used for educational purpose. Programmable robots can be developed by using LEGO kit. LEGO NXT brick uses a 32-bit ARM processor with a LCD Display powered by battery. The 32bit ARM processor comes with 256Kilobytes of flash memory for program storage

and 64 kilobytes of RAM for storing data during program execution. The brick consists of four input ports for connecting sensors and three input/output ports for servo motors. One more processor is used to deal with the data obtained from sensors. This processor has 4 kilobytes of flash memory and 512 bytes of RAM. To communicate with a computer Bluetooth or Universal Serial Bus (USB) facility is provided. It comes with NXT-G programming software.

TABLE I

LIST OF LEGO MINDSTORMS COMPONENTS USED FOR PICK AND PLACE ROBOT

Tasks	LEGO Components
Object Detection	Ultrasonic Sensor
Pick and Place	Servo Motors
Conveyor Belt	Touch Sensor, Tetrix Servo Motor
Color Sorting	Color Sensor, Servo Motor

LEGO Mindstorms NXT kit consists of various sensors like Ultrasonic sensor, Touch sensor, Color Sensor, Light sensor, Sound Sensor etc. Ultrasonic sensor can be used for object recognition/detection. It can operate between the ranges of 0-255 cm. Color sensor is used for detection various colors. Color sensor V2 can detect up to 17 different colors.

For designing pick and place robot using LEGO Mindstorms NXT several components are used which are shown in Table I. Also the model of pick and place robot is shown in figure 3.

We have implemented this pick and place robot using LEGO Mindstorms NXT kit. There are four main tasks, viz. object detection, pick and place, conveyor belt and color sorting. Components used in each task, for developing this robot, are shown in Table I. Also the model of pick and place robot is shown in figure 4.

Object detection task uses ultrasonic sensor for detecting the object. We have set the object detection range of 4-10 cm. If the object is within the given range, the robot will detect the object. Next task of the robot is to pick up the object. 2 LEGO NXT servo motors (motor A and B) are used for movement of the arm. Another servo motor is used to grab the object (Motor C). To grab the object the motor C will move in 83 degrees in forward and reverse directions. After grabbing the object, motors A and B makes 3.5 rotations and places the object on conveyor belt. The touch sensor is used as a switch for starting the rotations of conveyor belt. If touch sensor is pressed the conveyor belt starts to rotate. As mentioned in the Table I, tetrix motor is used for conveyor belt. It stops by the color sensor, which senses the color of object. We have used white and green color objects. Motor D is a servo motor used to push the object, it is white in color else the conveyor motor will move forward with a speed of 8. Motor D moves in 55 degrees to push the object.

B. Advantages of Neural Network Technique

Model-based fault diagnosis approaches depend upon analytical mathematical models. The accuracy of these models

has direct impact on system performance and reliability.

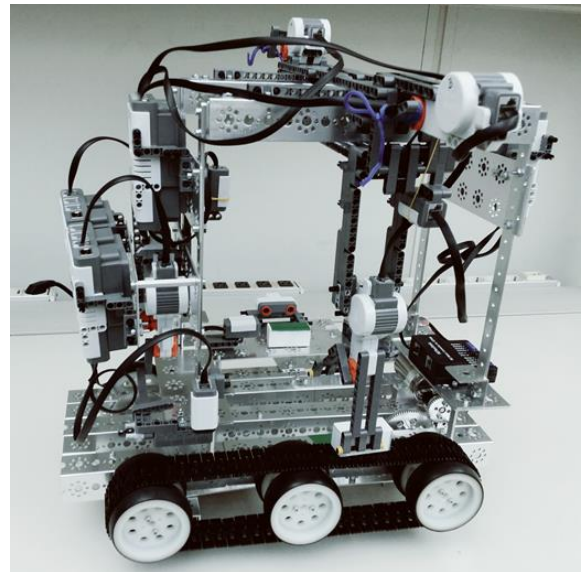


Figure 4: Pick and place robot using LEGO Mindstorms NXT kit

For complex and uncertain systems, use of mathematical models can become very complicated as well as time consuming. To overcome this drawback computational intelligence and learning based mathematical approach can be more promising. In last few decades, neural networks, fuzzy logic, genetic algorithms is extensively used and successfully applied to engineering systems.

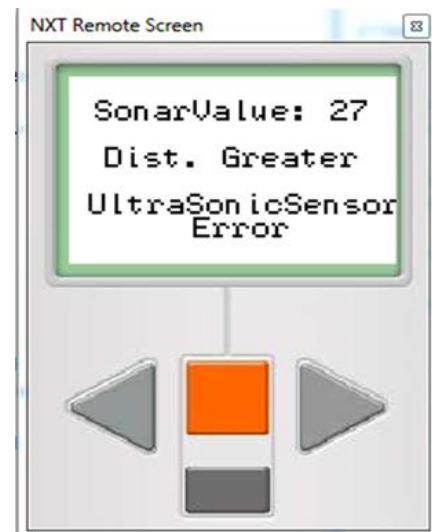


Figure 5: Object detection using ultrasonic sensor (with error)

The fuzzy if-then rules or boolean methods requires expert knowledge of the system, which can be a drawback of this method. To handle the large quantitative datasets, neural networks can be beneficial. Neural networks also provides an excellent frameworks for identification of nonlinear systems.

All these properties make neural network a promising tool for various applications such as pattern recognition, clustering, information integration, classification. And as a result, neural network has been extensively applied to fault diagnosis. Neural network based classifier is used for decision making, which classifies health as well as faulty data from the system. But it has a disadvantage that it requires data from all possible fault situations. In our proposed system, it uses the data from only a particular component which is passed through fault tolerant systems. Neural network based prediction methods, automatically learns from the nonlinear data from systems. The historical input and output data can be used to train the neural network. Therefore, neural network based prediction methods, have potential advantages such as self-learning, self-adaptability, parallel processing capabilities.

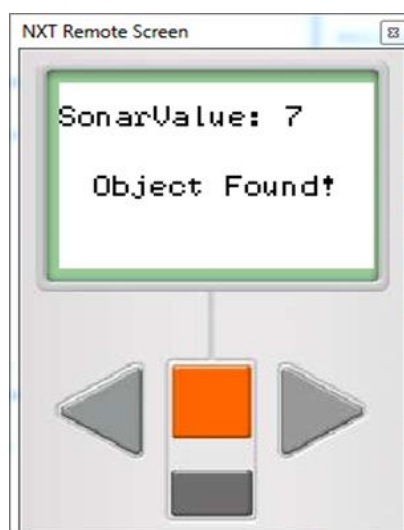


Figure 6: Object detection using ultrasonic sensor (without error)

C. Results

Object detection is one of the task in pick and place robot. We have checked the error in object detection by using fault detection unit. Figure 5 shows the output of ultrasonic sensor with error. Figure 6 shows the output without error. The LEGO NXT controller brick displays the status of error.

The next task is to detect errors occurring in the actuators, used in pick and place robot. We have set input in degrees, for the movement of these actuators. After the movement of actuator, the actual output movement in degrees is measured. This target value and actual output is passed through the fault detection unit to calculate faults.

We have calculated the residual errors of two actuators, motor C and motor D. The target output and actual output is plotted on graph.

Figures 7 and 8 shows fault detection of motor C for forward and reverse direction respectively. For motor C, the target input is set to 83 degrees which is shown by blue line. But, the actual movement performed by motor C varies from target input. The actual input is shown by dashed gray line in the figure.

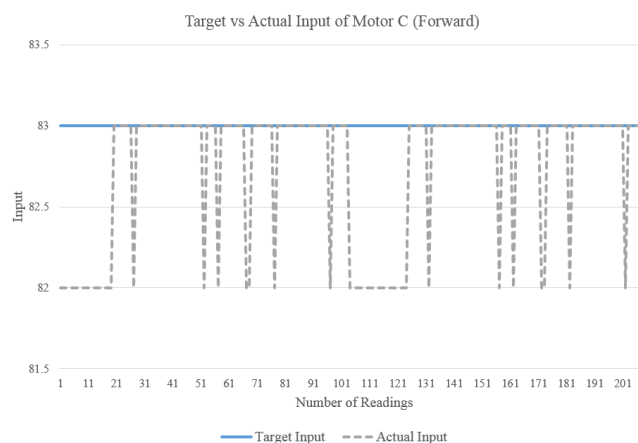


Figure 7: Fault detection of Motor C (forward direction)

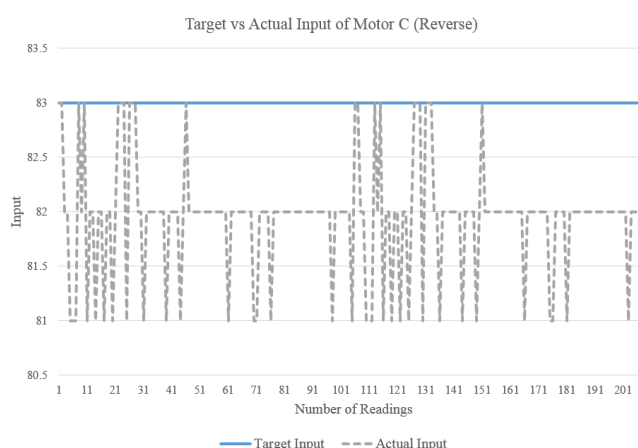


Figure 8: Fault detection of Motor C (reverse direction)

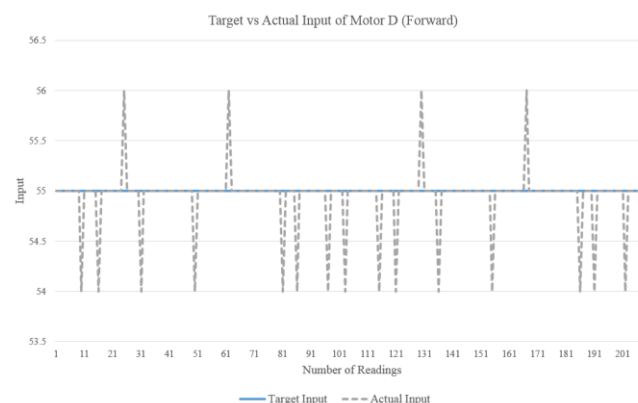


Figure 9: Fault detection of Motor D (forward direction)

Motor D is used in color sorting module. Target input of motor D is 55 degrees. That means the motor should move in 55 degrees in ideal condition with no faults in the motor. But the actual input of the motor differs from target input. This fault is detected by fault detection unit. Figures 9 and 10 shows fault

detection of motor D in forward and reverse direction respectively.



Figure 10: Fault detection of Motor D (reverse direction)

The set of monitored output of motor C and D has been given to fault tolerant system (shown in fig. 3) for minimizing the faults occurring in the two motors.

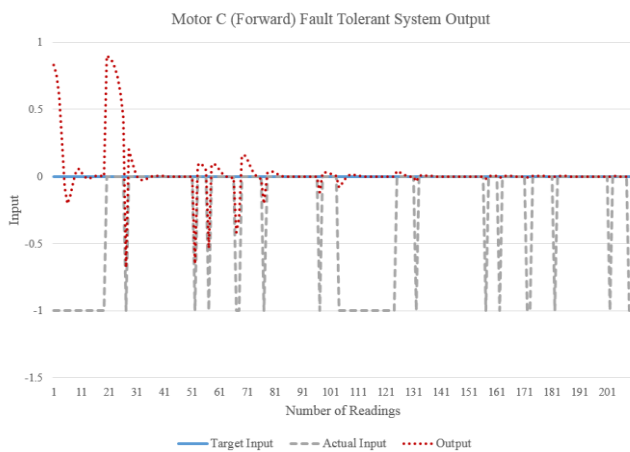


Figure 11: Fault tolerant system output (Motor C forward direction)

In this paper, we have considered only actuator faults for fault tolerance. The fault tolerant system trains the faulty data from actuators using neural network technique, to minimize the faults occurring in motors. Figures 11 and 12 shows results of angular movement of motor C for forward and reverse direction.

The angular movement of motor D is shown in figures 13 and 14 respectively. Target input is shown by solid blue line. But, as there are errors in the system, actual input is different than target input which is represented by dashed gray color line. After applying neural network technique, we have successfully reduced the errors in the system. The dotted red line shows the fault tolerant systems output, which shows that the error can be reduced up to great extent. The performance of pick and place robot can be improved by providing this fault tolerant systems output, as an input to the robot.

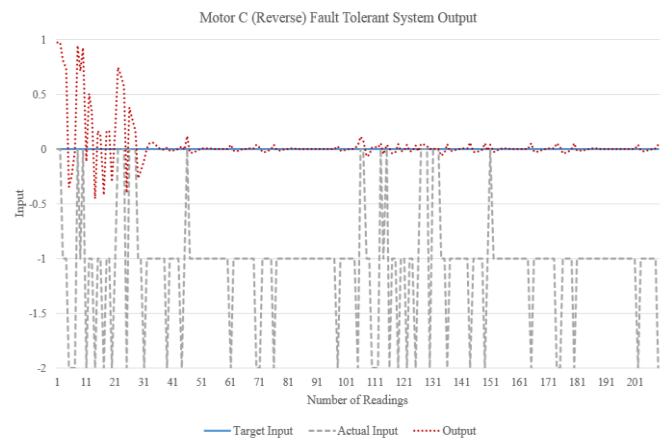


Figure 12: Fault tolerant system output (Motor C reverse direction)

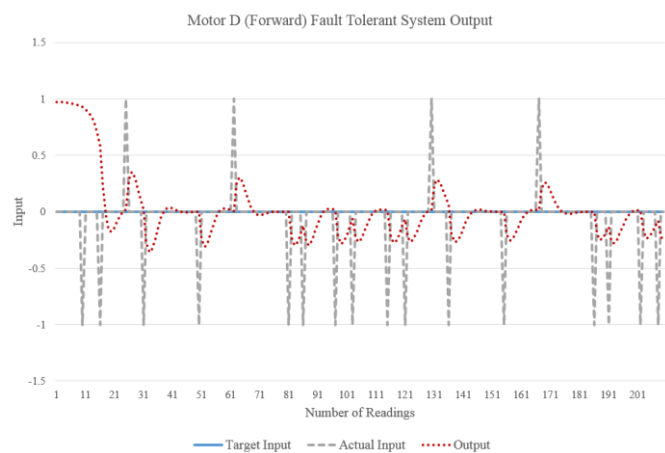


Figure 13: Fault tolerant system output (Motor D forward direction)

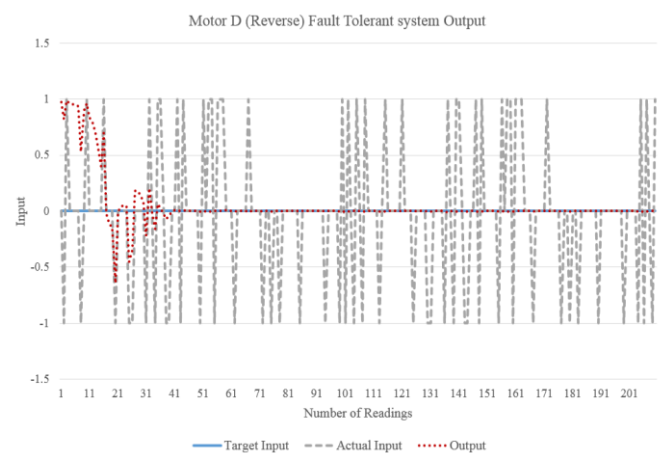


Figure 14: Fault tolerant system output (Motor D reverse direction)

V. CONCLUSION

Now days, it is necessary to have fault detection and fault tolerance techniques for smart industrial systems to increase the productivity. In order to achieve this goal, we have proposed neural network based fault detection and fault tolerance technique. For case study, we have developed LEGO based pick and place robot to implement our proposed system. After developing the robot, we have measured the errors of sensors and actuators. Residual generation method is used to compute the difference between actual output and targeted output. By using this error data, we have trained the neural network to reduce the actual output. The results of this gives to the pick and place robot to reduce the errors. The results shows that, the actuator faults are minimized using fault tolerant system. Hence, we can conclude that, the fault detection and fault tolerant systems makes it possible, to identify the faults and react to these faults, which can affect the operation of autonomous robotic systems.

ACKNOWLEDGMENT

This work was supported by the ICT R and D program of MSIP/IITP. [B0101-16-0663, Safety-critical Distributed Modular SW Platform]

REFERENCES

- [1] CyberphysicalSystems, <http://ieeecs.org/sites/ieeecs.org/files/documents/loCT-Part3-02CyberphysicalSystems.pdf>, June 2016.
- [2] Lee, Jay, Behrad Bagheri, and Hung-An Kao. "A cyber-physical systems architecture for industry 4.0-based manufacturing systems." *Manufacturing Letters* 3 (2015): 18-23.
- [3] Industry 4.0, <http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1293&context=bise>, May 2016
- [4] Chun, Ingeol, et al. "Faults and adaptation policy modeling method for self-adaptive robots." *International Conference on Ubiquitous Computing and Multimedia Applications*. Springer Berlin Heidelberg, 2011.
- [5] Baghermezhad, Farzad, and Kash Khorasani. "A robust fault detection scheme with an application to mobile robots by using adaptive thresholds generated with locally linear models." 2013 IEEE Symposium on Computational Intelligence in Control and Automation (CICA). IEEE, 2013.
- [6] Blanke, Mogens, and Jesper Sandberg Thomsen. "Electrical steering of vehicles-fault-tolerant analysis and design." *Microelectronics reliability* 46.9 (2006): 1421-1432.
- [7] Tao, Gang, and Suresh M. Joshi. "Adaptive control of systems with actuator failures." *Adaptive control of nonsmooth dynamic systems*. Springer London, 2001. 111-156.
- [8] Tang, Xidong, Gang Tao, and Suresh M. Joshi. "Adaptive actuator failure compensation for nonlinear MIMO systems with an aircraft control application." *Automatica* 43.11 (2007): 1869-1883.
- [9] Tao, Gang, Shuhao Chen, and Suresh M. Joshi. "An adaptive control scheme for systems with unknown actuator failures." *Automatica* 38.6 (2002): 1027-1034.
- [10] Yang, Zhenyu, Mogens Blanke, and Michel Verhaegen. "Robust control mixer method for reconfigurable control design using model matching." *IET Control Theory and Applications* 1.1 (2007): 349-357.
- [11] Zhuo-hua, Duan, C. A. I. Zi-xing, and Y. U. Jin-xia. "Fault diagnosis and fault tolerant control for wheeled mobile robots under unknown environments: A survey." *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2005.
- [12] Janglová, Danica. "Neural networks in mobile robot motion." *Cutting Edge Robotics* 1.1 (2005): 243.
- [13] Tripathi, G. N., and V. Rihani. "Motion planning of an autonomous mobile robot using artificial neural network." *arXiv preprint arXiv:1207.4931* (2012).
- [14] Gavrilov, Andrey V., and Artem Lenskiy. "Mobile robot navigation using reinforcement learning based on neural network with short term memory." *International Conference on Intelligent Computing*. Springer Berlin Heidelberg, 2011.
- [15] Velagic, Jasmin, Nedim Osmic, and Bakir Lacevic. "Neural network controller for mobile robot motion control." *International journal of intelligent systems and technologies* 3.2 (2008): 127-133.
- [16] Hofbaur, Michael, et al. "Improving robustness of mobile robots using model-based reasoning." *Journal of Intelligent and Robotic Systems* 48.1 (2007): 37-54.
- [17] Steinbauer, Gerald, and Franz Wotawa. "Model-Based Reasoning for Self-Adaptive Systems—Theory and Practice." *Assurances for Self-Adaptive Systems*. Springer Berlin Heidelberg, 2013. 187-213.
- [18] Eski, Ikbal, et al. "Fault detection on robot manipulators using artificial neural networks." *Robotics and Computer-Integrated Manufacturing* 27.1 (2011): 115-123.
- [19] Macías-Escrivá, Frank D., et al. "Self-adaptive systems: A survey of current approaches, research challenges and applications." *Expert Systems with Applications* 40.18 (2013): 7267-7279.
- [20] Zhang, Youmin, and Jin Jiang. "Bibliographical review on reconfigurable fault-tolerant control systems." *Annual reviews in control* 32.2 (2008): 229-252.
- [21] www.hq.nasa.gov/office/codeq/risk/docs/ftacourse.pdf
- [22] Isermann, Rolf. "Model-based fault-detection and diagnosis—status and applications." *Annual Reviews in control* 29.1 (2005): 71-85.



Prajakta Jadhav is currently Master's student in University of Science and Technology (UST), Daejeon South Korea. She received bachelor's degree in Electronics and Telecommunication Engineering in 2011 from University of Pune, India. She is a student researcher in Cyber-Physical Systems Research Lab in Embedded SW Research Division of Electronics and Telecommunications Research Institute



In-Geol Chun received his Ph.D. and M.S. degrees in Electrical and Computer Engineering from SungKyunKwan University, Korea, in 2010 and 1997 respectively. He is currently the director of CPS research section in Electronics and Telecommunications Research Institute (ETRI) from 1998 and also an adjunct professor in University of Science and Technology (UST) from 2012. His research interest are Cyber-Physical Systems (CPS), Smart Factory, Autonomic Computing Systems, Embedded Systems and Software Engineering