Three-Term Fault Diagnostic Algorithm for Optimized Incubator Control Syatem

[1]Iloh, P. I., [1]Nwabueze, C. A. and [2]Ohanaja, P.
[1]Department of Electrical/Electronic Engineering, Chukwuemeka Odumegwu Ojukwu University, Uli.
[2]Department of Works, Medical Instrumentation Unit, Nnamdi Azikiwe University Teaching Hospital (NAUTH), Nnewi.
Email: ca.nwabueze@coou.edu.ng.

ABSTRACT

Given the increasing need to optimize production time of incubators, a number of approaches have been used to develop incubator control system but none of these methods used fault diagnostic and tolerant algorithms integrated within Proportional-Integral-Derivative (PID) control algorithm. Besides, most of the existing fault diagnostic and tolerant algorithms were not developed with remote based applications in mind. This work developed and integrated a fault diagnostic and tolerant algorithm with remote reporting capability into the Existing PID Algorithm (EPIDA). Behavioral models of the priority components were used to develop an Intelligent Algorithm (IA) on embedded c language, a hybrid of c and c++ languages while arduino uno was used as a target controller for the developed IA. Validation of the IA for Fault Diagnosis and Tolerant Control (IAFDTC) was done using the EPIDA and the IAFDTC to control the process plant under the absence and presence of simulated fault. With 99.05% accuracy as the minimum benchmark, EPIDA had accuracy of 99.78% under normal conditions and dropped to 96.95% in the presence of simulated fault confirming that the EPIDA was not originally designed to diagnose and tolerate faults. However, the IAFDTC had accuracy of 99.96% under normal operation and 99.62% accuracy under the influence of fault. In addition, it sends remote reports via wireless network to control system custodians. This shows that IAFDTC achieved an improvement of 2.67% in the accuracy of the existing algorithm while maintaining the system's availability and integrity under the influence of fault condition.

Keywords: Proportional-Integral-Derivative Algorithm, Controller, Intelligent Algorithmn, Fault Diagnosis and Tolerant Control

1.0 INTRODUCTION

Worldwide every year, over 4 million infants die within a month of birth. Of this number, 3.9 million belong to the developing world. 25% of the deaths are caused due to complications of prematurity, most often heat and water loss that can be prevented by using an incubator [6]. Premature infants are babies born prior to the normal 36 or 37 weeks of gestation within the womb. As a result, their physiological systems are underdeveloped making the infant vulnerable to a number of health complications. This inadequate thermoregulation, wherein their physiology is not able to compensate for the heat these babies loose and the loss of water from the body are by far the leading causes of death in premature infants [7].

A neonatal incubator is a device consisting of a rigid box-like enclosure in which an infant may be kept in a controlled environment for medical care. The temperature range of a baby incubator is between 36.5–37.2 ^{o}C as reported by Victoria and Dale [9]. Therefore, optimal temperature set point for such incubator should be 36.85 ^{o}C. The incubator temperature should not be adjusted by no more or less than 0.5 ^{o}C at a time, implying that the fluctuations in the control system should fall within the range of 36.85 ± 0.5 ^{o}C at any given time [9].

This definitely requires precision control. On-Off control with its characteristic oscillation will not be able to achieve such control objective, hence Proportional-Integral-Derivative Algorithm (PIDA) is designed to handle such precision control. However, PIDA was not designed to handle self-diagnosis and fault tolerant. Given the criticality of baby incubators, there is need to incorporate proactive measures in the existing incubator control system such that the system will know when fault occurs in the system, take steps to maintain control objective while classifying and reporting the fault(s) to the maintenance personnel for quick correction. Moreover, with the advancements in Information and Communication Technology (ICT), there is need for position control of system for Internet of Things (IoT). The EPIDA was not designed for communication with external world since it is a close loop algorithm for precision control within the control loop. Therefore, there is need to improve the EPIDA so that it will be self-reporting, self-diagnostic and fault tolerant while maintaining control system's availability and integrity. This paper presents a fault diagnostic algorithm based on the behavioral model of the priority components, and embeds it on the existing PID control algorithm. In addition, a prototype fault tolerant and diagnostic control system was built and validated while incorporating remote reporting in the new algorithm.

2.0 THE PID CONTROL LOOP

The back-end algorithm used in control systems are Proportional-Integral-Derivative (PID) and are shown in equations 1 and 2:

$$m = K_p e(t) + K_i \int e\,(t)dt + K_d \frac{de(t)}{dt} \qquad (1)$$

$$\mathrm{e} = \mathrm{S} - \mathrm{b} \qquad (2)$$

where K_p, K_i, K_d are proportional, integral and derivative constants respectively, m is the output of the PID controller, e is the error signal while S is the set point of the process and b is the feedback signal. Every form of control system revolves around these equations be it robust control, fault tolerant control, on-off control, feedback control as well as feed forward control [4]. PID control equation has enjoyed universal acceptance in automation industry because the past, present and future states of the controlled variables were considered in Eqn. 1. The proportional component takes care of the present error in the controlled variable while the integral component takes care of the past errors just as the derivative component is in charge of the future errors [1, 2]. The pictorial representation of PID control equation is shown in figure 1 [3]. PID algorithm has a number of advantages in control applications. It was primarily designed to control systems whose transfer functions are not known [5]. It is very good for dynamic systems as well as systems whose mathematical models are uncertain [8]. A typical PID control set up is made up of three basic components: the PID controller, the process plant and the sensors. PID controller is a hardware device used to implement Eqns. 1 and 2, while the sensors convert the measured process variable to a form that can be read by the PID controller.

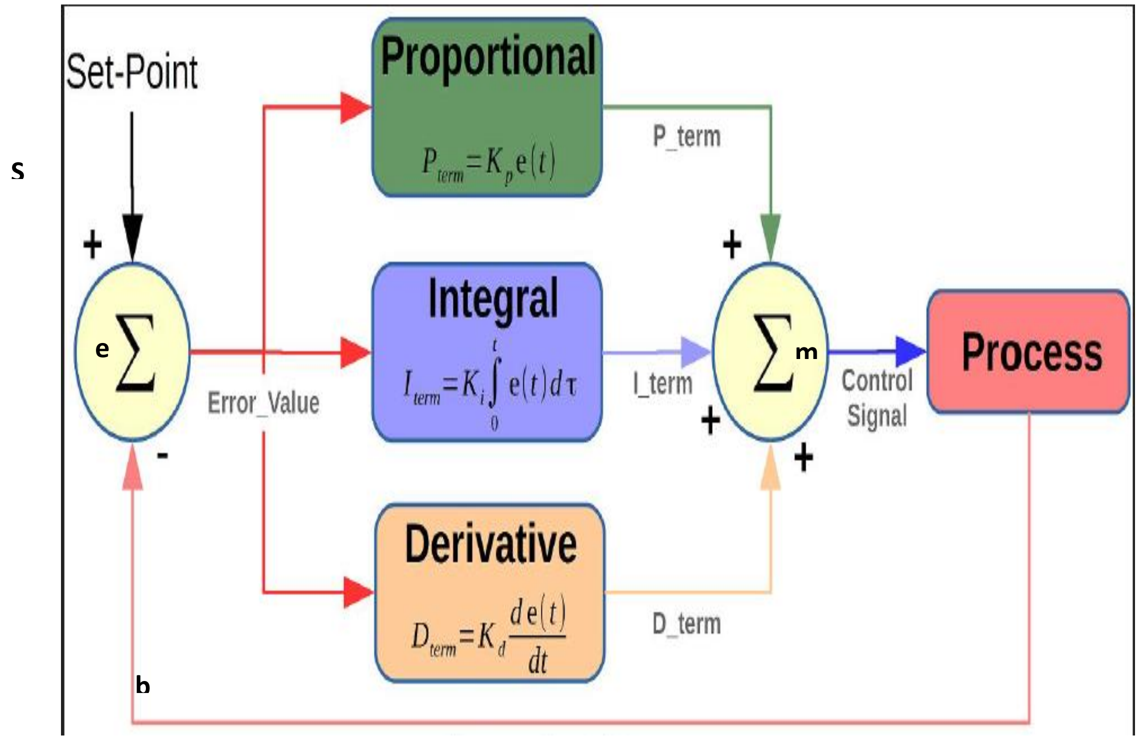

Figure 1: PID Algorithm in a Control Loop [3].

3.0 MATERIALS AND METHOD

The proposed Self-diagnostic and Fault Tolerant Control System with remote monitoring (SDFTCSRM) was realised using bottom-up approach by describing the overview, followed by identification and design of the Priority Components (PC) of the Base End System (BES). Development of Intelligent Algorithm for the Fault Tolerant Control System (IAFTCS) was done using the behavioral model of the PC. The software implementation of the developed IAFTCS was done using embedded c language, a hybrid of c and c++ languages while the flow chart for remote monitoring of the BES was developed and implemented on the front end system leveraging on the Graphical User Interface (GUI) capability of the arduino uno Integrated Development Environment (IDE). Figure 2 is the block representation of the proposed SDFTCSRM. The system mainly consists of three major sections: the front end, back end and the third party maintenance.

3.1 Hardware Implementation

Sensors 1 and 2 measure the temperature of the process plant. The output of the sensors serves as analog inputs A0 and A1 of the on-board analog to digital converters of the controller. The controller through its outputs displays the state of the system on a Liquid Crystal Display (LCD) and controls the process plant through the triac driver moc3021 via the triac BT139. The controller also sends remote report both to the system custodians and third party maintenance manager when fault is detected in the back end system while alerting the system's operator via buzzer alarm. The triac driver moc3021 is instrumental to the switching of the load (heater). The GSM transmitter is a modem based on sim900 chip which operates within the frequency band of 900MHz. It is selected in this work because it is compatible with arduino uno. To detect faults arising from the actuator, an opto coupler is needed, which works in such a manner that when there is ac input to it, the VCC (5v dc) at pin 5 of the opto coupler will appear at pin 4 of the coupler. Thus pin 4 of the coupler is pulled down so that 0 volt will always be at pin 4 when there is no input to the opto coupler as shown in Figure 3.

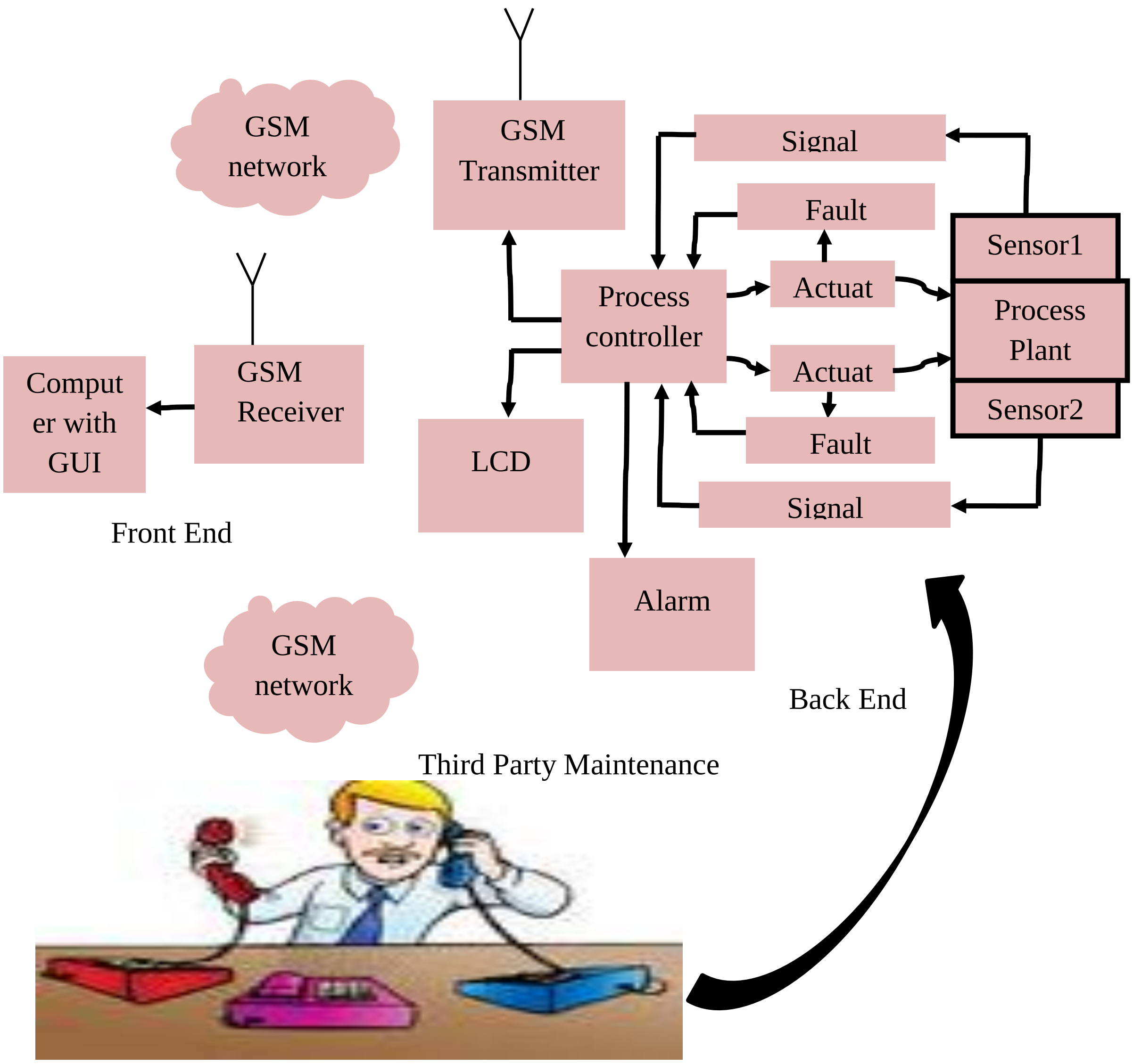

Figure 2: Block diagram of the Proposed Self-diagnostic and Fault Tolerant Control System with Remote Monitoring.

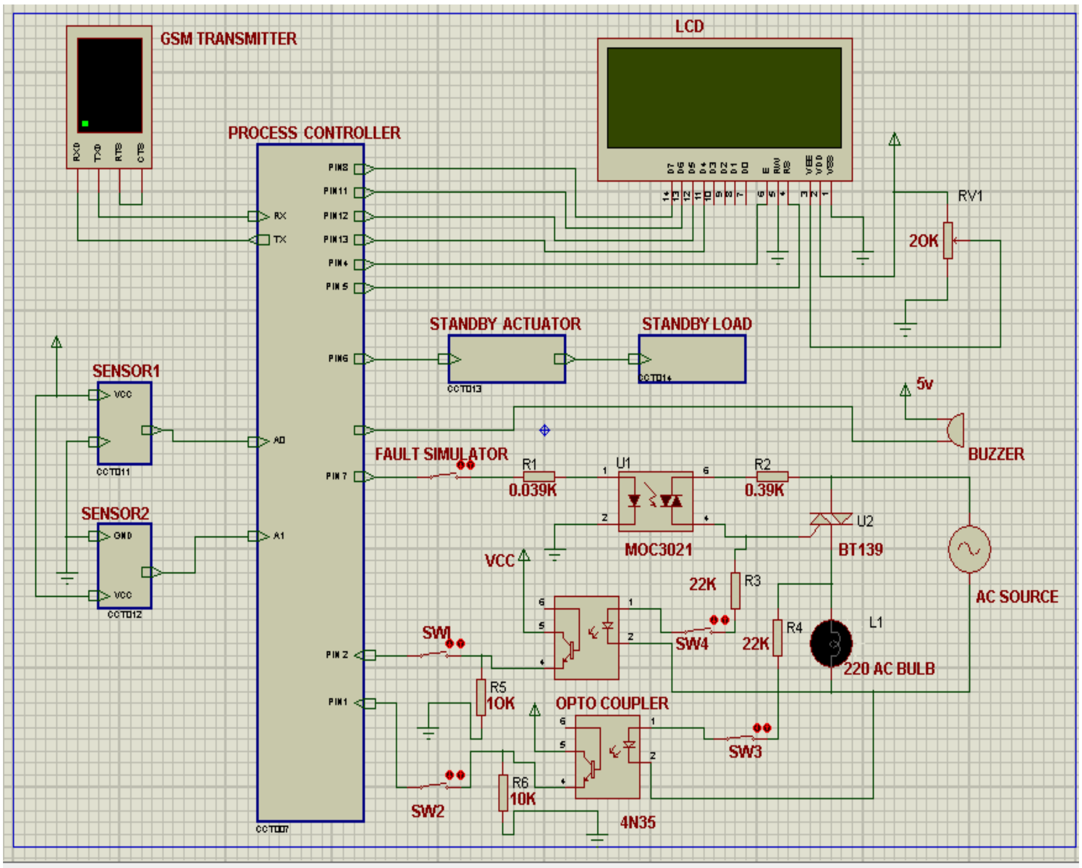

Figure 3: Circuit Diagram of the Hardware Components in the SDFTICS

Figure 4 shows the complete hardware implementation of SDFTICS with remote reporting. The opto couplers, sensors, triac and its driver, LCD and GSM modem were first of all mounted on a circuit board before being interfaced to arduino board with on-board microcontroller. The electronics that controls the plant was interfaced to the GUI of arduino IDE so that automated real time data can be collected from the process plant.

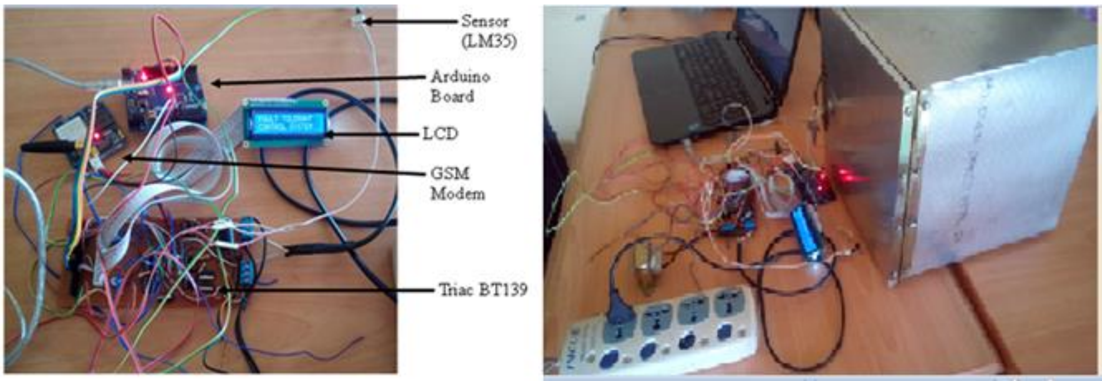

Figure 4: Hardware Implementation of the SDFTCS of Neonatal Incubator.

3.2 Software Implementation

The software for implementation of the intelligent algorithm for fault diagnosis and tolerant control was developed using embedded c language and compiled using the editor and compiler in arduino integrated development environmentas shown in Figure 5. This section leveraged on the behavioral models of the priority components to develop the intelligent algorithm for fault diagnosis and tolerant control in the incubator system. The flowchart of the intelligent algorithm, X1-X8 is the links which control the sub-flowchart to the main flow chart.

X1 represent the detailed description of software initialization,
X2 checks whether fault has occurred in the control system,
X3 is for fault classification in the control system,
X4 represents the main control of the process plant,
X5
X6 represents the slave control of the process plant.
X7
X8

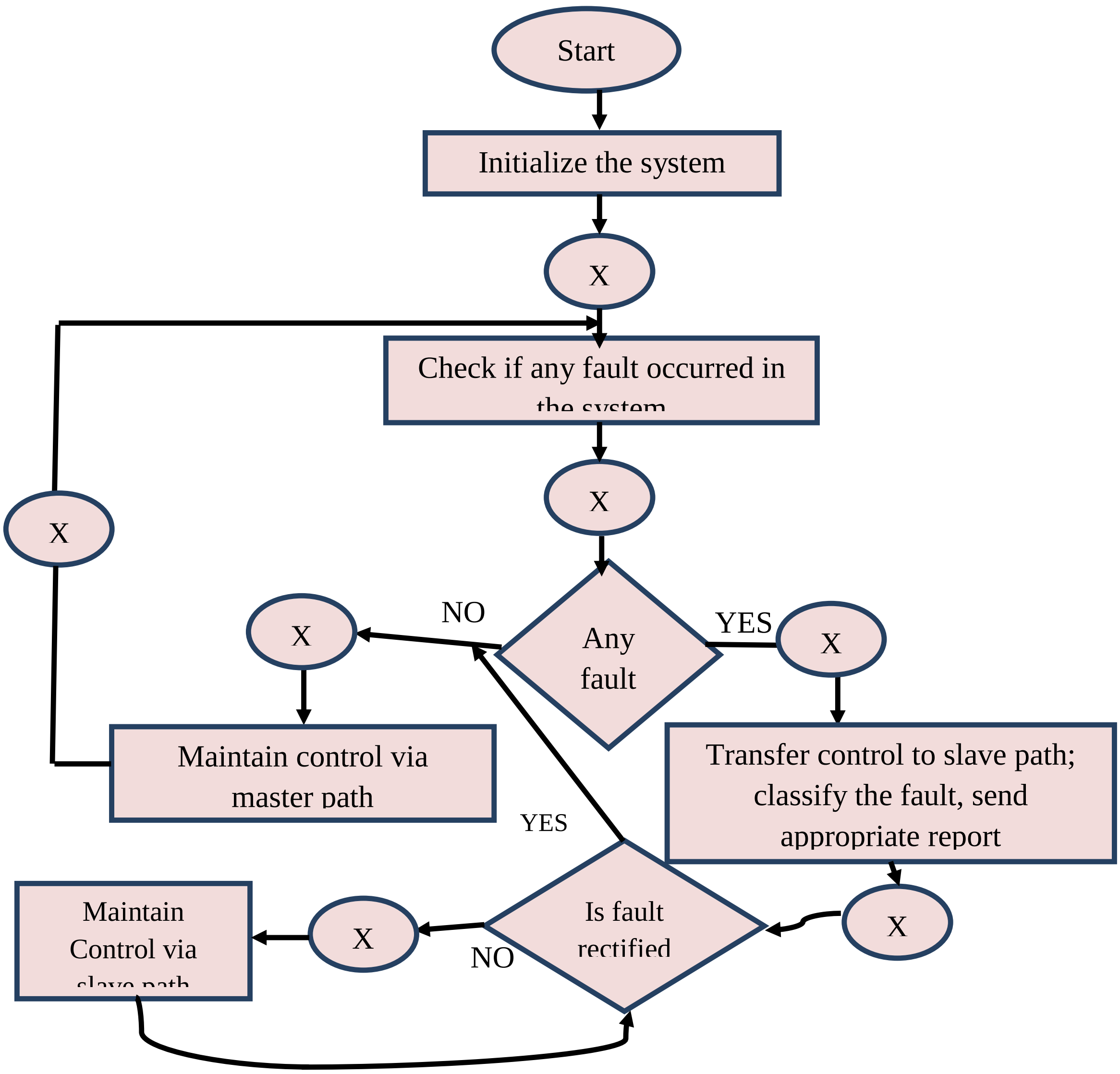

Figure 5: The Intelligent Algorithm for Fault Diagnosis and Tolerant Control (IAFDTC).

3.3 Data Collection

Four sets of data were collected from the process plant. The first data, Data 1 is the plant's response to the existing PID algorithm. The second set of data, Data 2, is the plant's response to the existing PID algorithm when the algorithm was subjected to simulated fault condition. The third data, Data 3 is the plant's response to the intelligent fault diagnostic and tolerant PID algorithm under normal operation while the final data Data 4 is plant's response to the intelligent fault diagnostic and tolerant PID algorithm when the algorithm was subjected to the simulated fault condition.

4.0 RESULTS AND ANALYSIS

Figure 6a is the graph of the process plant response to the EPID algorithm under normal condition while figure 6b is the graph of the plant's response to the FDTPID algorithm under normal condition. Figure 6c shows the comparison of responses of the process plant to the FDTPID algorithms and the EPID algorithms under normal conditions. Result from the Figures shows that the EPID algorithm maintained the control objective in the considered process plant (incubator) with an accuracy of 99.78% while the FDTPID control algorithm improved the process plant availability and intergrity with an accuracy of 99.96% and their performance is acceptable under normal condition since the minimum control accuracy required to achieve the control objective in the considered process plant is 99.05%. Therefore the percentage improvement is $((99.96 – 99.78)/99.78)/) * 100 = 0.18\ \%$.

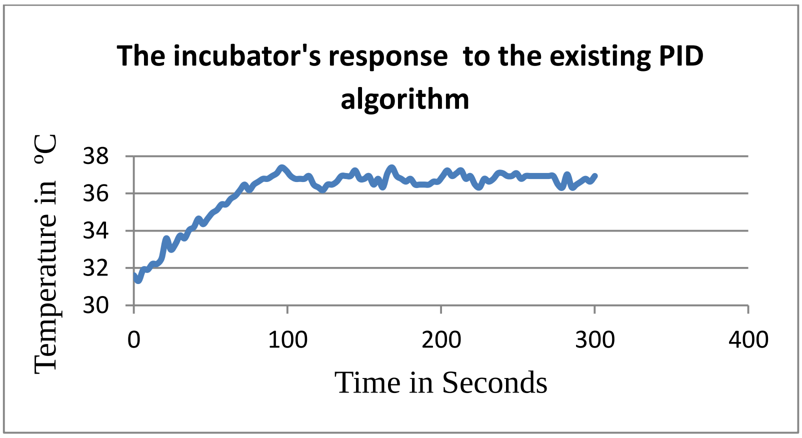

Figure 6a**:** Plant's Response to the Existing PID Algorithm under normal Condition

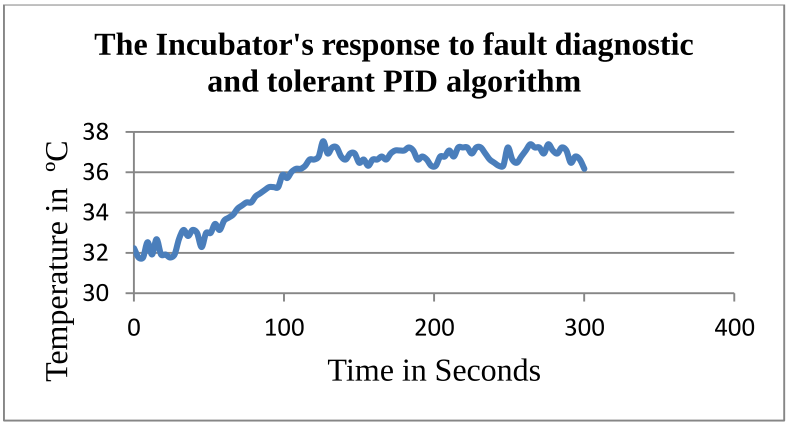

Figure 6b: Plant's Response to the Fault Diagnostic and Tolerant PID Algorithm under normal Condition

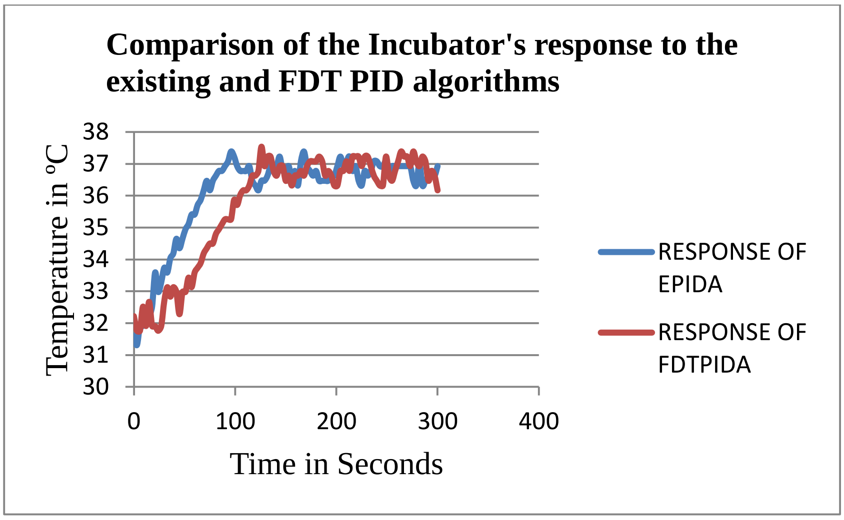

Figure 6c: Response of the Process Plant to the FDT and the Existing PID Algorithms under normal Condition

Figure 7a shows the response of the process plant to the EPID algorithm under fault condition while figure 7b is the response of the process plant to the FDTPID algorithm under the same fault condition. Figure 7c is the comparison of the plant's response to the EPID and FDTPID algorithms under the same fault condition. It is seen from figures 7c that the FDTPID algorithm was able to achieve the control objective even in the presence of the simulated fault condition with an accuracy of 99.62%, while an accuracy of 96.95% was achieved in the case of the EPIDA. It is very clear that the EPIDA could not achieve its control objective and not suitable for fault diagnosis and tolerant control. The practical impact on the incubation is that the control objective will not be achieved and that the affected baby may die or would be deformed.

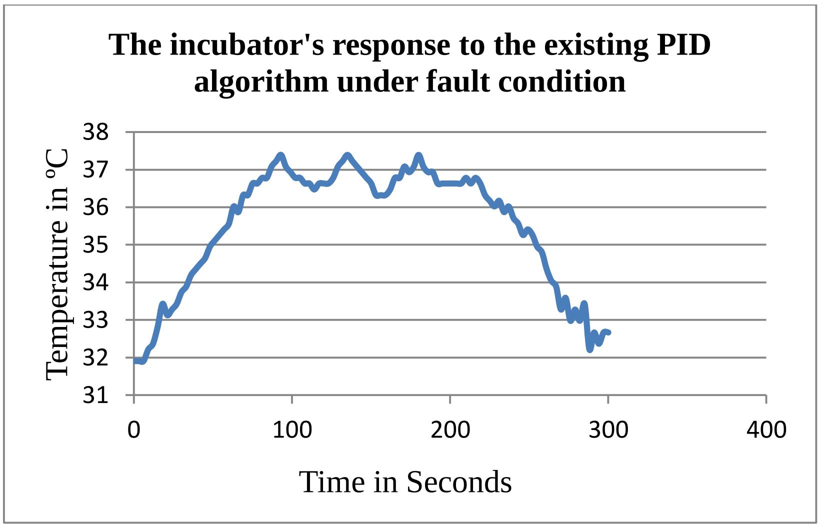

Figure 7a: Response of the Process Plant with the existing PID Algorithm Fault Condition

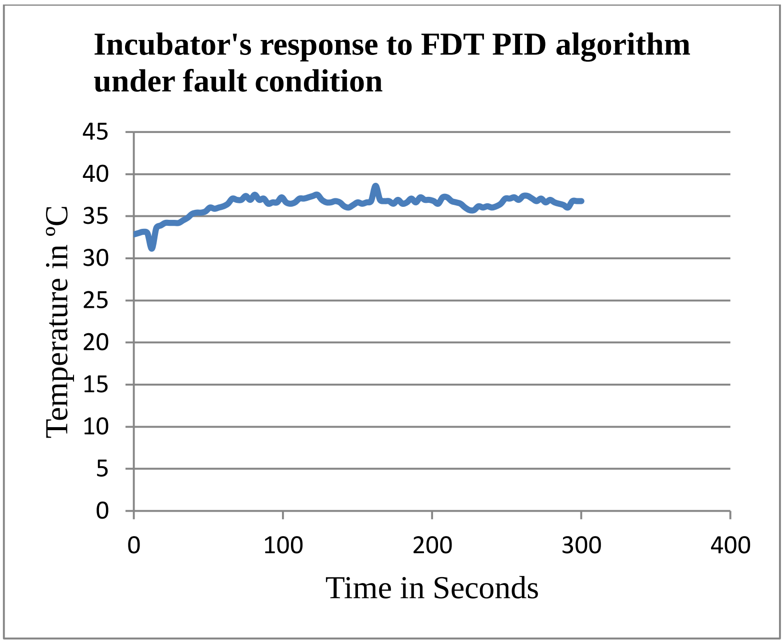

Figure 7b: Response of the Process Plant with the FDTPID Algorithm under Fault Condition

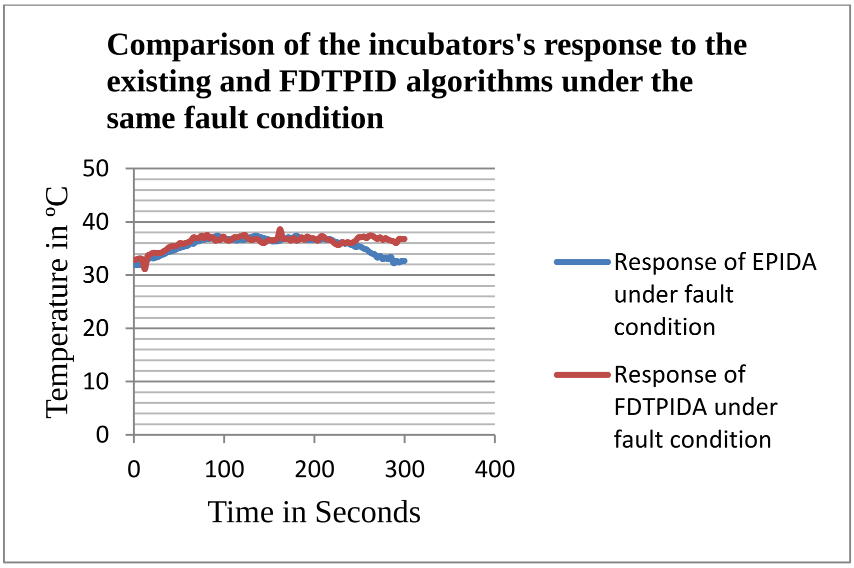

Figure 7c: Response of the Process Plant to the FDT and the Existing PID

4.1 Validation of the Fault Diagnostic and Tolerant PID Algorithm

The result shows that under fault, the EPID algorithm had the Mean Steady State Error (MSSE) of 1.1231147 while that of FDTPID algorithm was 0.1390164. It means that the FDTPID algorithm was able to improve the accuracy of the EPID algorithm from 1.123 to 0.139 while maintaining the system's availability and integrity under the influence of fault condition. In terms of percentage, the improvement is 99.62% - 96.95% = 2.67%.

CONCLUSION

Fault diagnostic and tolerant algorithm with remote reporting capability integrated into the EPID control algorithm of incubator control system reduced the down time of incubator machines using its self-diagnostic ability with an improvement of 2.67% in the accuracy of the existing algorithm while maintaining the system's availability. This self diagonistic neonatal incubator design satisfied the standard requirements of industrial control systems.

REFERENCES

1. Abdelgeliel, M., Qaud, F. and Ashour, H. (2014), Realization of Adaptable PID Controller within an Industrial Automated System, *IEEE 11th International Conference on Control and Automation*, pp. 965-970.
2. Bela, G. L. (2006), Process Control and Optimization, Boca Raton Publication, London.
3. Bolton, W. (2006), *Programmable Logic Controllers*, Elsevier Press, Armsterdan.
4. Dale, R. P. and Stephen, W. F. (2009), Industrial Process Control System, Fairmont Press, . India.
5. Ikhlef, A., Kihel, M., Boukhezzar, B., Mansouri, N. and Hobar, F. (2015), Remote PID Control of Tank Level System, IEEE *International Conference on Interactive Collaborative Learning*, pp. 20-24.
6. Lokuge, P., Maguire, Y. and Wu, A. (2002), Design of a Passive Incubator for Premature Infants in the Developing World, Massachusetts Institute of Technology, www.stanford.edu/~cbauburn/basecamp/dschool/nepalstudio/MIT% 20Premature. pdf.
7. Madden, S. (2000) The Preemie Parents' Companion, The Harvard Common Press, Boston.
8. Stan, Ż. (2013), An Introduction to Proportional-Integral-Derivative (PID) Controller, Purdue University, Lafayette USA.
9. Victoria, B., and Dale, G. (2018), Newborn Services Clinical Guideline, Care ofthebabyinanIncubator. www.adhb.govt.nz/newborn/Guidelines/Admission/BabyInIncubator.htm.