Performance Evaluation of Tilt-based Vehicle Accident Report System Using Triple Modular Redundancy

Rakhmadhany Primananda¹, Bayu Bagus Prabowo², Agung Setia Budi³, Mahardeka Tri Ananta⁴, Zainul Abidin⁵

¹,²,³,⁴,⁵ Brawijaya University, Malang

¹rakhmadhany, ³agungsetiabudi, ⁴deka, ⁵zainulabidin@ub.ac.id,
²bayubagus200898@gmail.com

*Corresponding Author

Received 29 August 2023; accepted 13 December 2023

Abstract. A vehicle accident report system was designed to detect accidents accurately and send message to related person. However, after accident, some components or modules can be damaged and it may cause failure of accident detection. Therefore, Triple Modular Redundancy (TMR) method was implemented in the system to mask faults within the modules and ensure the system to continue working. TMR is one of passive hardware redundancy methods. The system includes three MPU-6050 sensors functioning as vehicle tilt sensors to provide hardware redundancy, a TC9548A module serving as an I2C multiplexer, an Arduino Nano acting as a microcontroller and voter, and a NodeMCU ESP8266 responsible for transmitting message. Based on experiment results, the TMR system has a reliability value of 0.9928, a failure rate value of 0.0004/cycle, and accuracy of 3 sensor value of 85%. While, the non-redundancy system has a reliability value of 0.9286, a failure rate value of 0.0053/cycle, and accuracy of single sensor value of 57%. The results of the system process were sent to and displayed via an e-mail. It can be concluded that the TMR system has better performance than the non-redundancy one.

Keywords: failure rate, Triple Modular Redundancy, reliability, voter, vehicle accident report

1 Introduction

Nowadays, the frequency of vehicle accidents has quite high, due to excessive speed, malfunction of vehicle part, and the drivers not being able to control their vehicles properly [1], [2]. In addition, the number of vehicle drivers on the roadway also affects the occurrence of accidents. Therefore, a vehicle accident report system is designed to gather all relevant information about the driver from computing devices, including incident detection, the location of the incident, and the closest police station or hospital, so that the driver gets rescue immediately. The system will send an alert message to the closest people who relate to the vehicle owner too [3], [4], [5], [6].

Previous research about a vehicle accident report system have been published with several methods. Design of a vehicle accident report system to detect accidents and transmit the accident information to the related person using Arduino Uno as a microcontroller, sensor vibration as an accident detector, and GSM module as data sender [3]. However, there are several issues in this system. The research is just a design proposal of the system, there is no system realization report. Then, the Arduino Uno is
large enough to be placed on the vehicle. Other research uses GSM and GPS modules as data senders too, but with the addition of MQTT protocol to send the data over the cloud in real-time [7], [8] and also VANET compatibility [9].

However, the vehicle accident report system has the potential for failure or damage to one of the components like the sensor due to the accident situation, vibration, and collisions. Therefore, the system needs a method to make it still work even in non-ideal conditions. One of the methods is offered by fault tolerant vehicle system. Fault-tolerant vehicle systems can be attained by various measures. To prevent bottlenecks in system design, a comprehensive approach should be adopted. Early detection and diagnosis aim to deliver enough time for implementing counteractions, including maintenance, reconfiguration, repair, or other operations. There is a possibility of the vehicle losing its functionality despite the controller working properly and the fault being diagnosed. The fundamental aspect of fault-tolerant vehicle design lies in the cooperation of all subsystems. Maintaining system functionality is achieved with redundancy by adding backup hardware, software, information processes, and subsystems into the system that perform the same functions as the original units. This redundant design scheme commonly includes sensors, actuators, microcontrollers, and communication network [10], [11]. In the case of reporting accident events, the research of various transportation-related industrial sectors is investigated. The primary emphasis is on exploring functional safety standards and technical solutions. In relation to the optimization model, the level of expected redundancy is a concern. Because of cost constraints and limited packaging capacity, employing redundancy concepts beyond the 3rd level is impractical. Therefore, Triple Modular Redundancy (TMR) can be considered as a potential architectural subsystem. Consequently, in the context of an automotive RAP (Reliability, Availability, and Performance), the question is not how much redundancy should be implemented, but rather whether it is necessary and where it would be most effective [12].

Based on the presented background, this paper proposes vehicle accident report system using one of the redundancy methods, Triple Modular Redundancy (TMR), based on tilt of vehicle. Then, the data of tilt is sent via e-mail to a related person. The use of the TMR method on this research is to create a system can work normally even if there is damage or error to components or data sent, so it can still operate when an accident occurs. The system uses three MPU-6050 sensors as tilt sensor a TC9548A module as a multiplexer I2C, an Arduino Nano as a microcontroller and act as a voter, and a NodeMCU ESP8266 as a message sender. The implementation of this system will be implanted directly onto the body or frame of the vehicle. It is essential for the system capable to capture data from sensor, analyze it, and generate output, such as a warning message that can be sent to a specified e-mail address.

TMR is one of the passive hardware redundancy methods that has triplicated modules in parallel fashion, in this proposed system using 3 tilt sensors. Majority voting is employed to determine the correct result. If one of the modules encounters a failure, the majority voter masks the fault by recognizing as a correct result of the remaining two fault-free modules. TMR can mask only one module failure. If a failure occurs in one of the remaining modules, the voter may produce an incorrect result [13], [14], [15].

The selection of the appropriate sensor for detecting accidents is a consideration in the design of vehicle accident report system in order to provide an optimal solution. The tilt sensor is used to monitor the tilt angle ensuring that the vehicle is in a safe or unsafe angle [16], [17], [18]. This sensor can be utilized to detect crashes or rollovers that occur during or after an accident [7], [19].

The system will evaluate of performance using 3 parameters, including sensor
accuracy, reliability, and failure rate. This 3 parameters will provide an overview of the TMR system, indicating that the system is considered reliable if it has high sensor accuracy, high reliability, and low failure rate.

2 Method

2.1 Hardware Design

The system consists of components as shown in Figure 1. There are three sensors of MPU-6050 which have same I2C bus. So an I2C multiplexer TC9548A is required as gatekeeper of communication between MPU-6050 and Arduino Nano. The TCA9548A allows a microcontroller to communicate with up to 8 I2C devices with the same I2C bus using the I2C communication protocol. Then, the microcontroller can select which I2C bus on the multiplexer to address. So the Arduino Nano can access the sensing result of all three sensors simultaneously. Then the Arduino Nano will process the recorded data and perform voting based on the results of the recorded data. The voting results will be changed to be token before sent to ESP8266 using serial communication. By going through the webhook feature from IFTTT site via WiFi [20], the token will be forwarded and converted to e-mail message which will be sent to user.

In order to realize the communication, some pins must be connected between available components. Table 1 describes the pairing of MPU-6050 and TC9458A. Table 2 describes the pins configuration between TC9548A and Arduino Nano. Table 3 describes pins configuration between Arduino Nano and NodeMCU (ESP8266). The NodeMCU is also important part of the Vehicle Accident Report System to send emergency message via internet. After placed together in a box, the hardware realization can be seen in Figure 2.

![Figure 1. Block diagram of Vehicle Accident Report System](image)

The MPU-6050 sensor has 3 axis, namely X-axis, Y-axis, and Z-axis. The X-axis represents the horizontal axis that can detect tilts in the left and right direction. The Y-axis represents the vertical axis that can detect tilts in the up and down direction. While, Z-axis represents the perpendicular axis that can detect rotations along vertical axis [21], [22], [23]. However, in this system is only taken in the X-axis because to observe the tilt of the vehicle to the left or right. The measurement of the tilt direction towards the horizontal plane (X-axis) has been attempted using a sensitivity range parameter of 8g in terms of g-force [24]. A positive value on the X-axis more than +8
indicate a tilt to the right. Conversely, a negative value on the X-axis less than -8 indicate a tilt to the left.

Table 1. The Pin configuration between MPU-6050 and TC9548A

<table>
<thead>
<tr>
<th>Pin MPU-6050</th>
<th>Pin TC9548A</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPU-6050 1</td>
<td>SCL</td>
</tr>
<tr>
<td></td>
<td>SDA</td>
</tr>
<tr>
<td>MPU-6050 2</td>
<td>SCL</td>
</tr>
<tr>
<td></td>
<td>SDA</td>
</tr>
<tr>
<td>MPU-6050 3</td>
<td>SCL</td>
</tr>
<tr>
<td></td>
<td>SDA</td>
</tr>
</tbody>
</table>

Table 2. Pin configuration between TC9548A and Arduino Nano

<table>
<thead>
<tr>
<th>Pin TC9548A</th>
<th>Pin Arduino Nano</th>
</tr>
</thead>
<tbody>
<tr>
<td>VCC</td>
<td>5V+</td>
</tr>
<tr>
<td>GND</td>
<td>GND</td>
</tr>
<tr>
<td>A0</td>
<td>GND</td>
</tr>
<tr>
<td>A1</td>
<td>GND</td>
</tr>
<tr>
<td>A2</td>
<td>GND</td>
</tr>
<tr>
<td>SCL</td>
<td>A5</td>
</tr>
<tr>
<td>SDA</td>
<td>A4</td>
</tr>
</tbody>
</table>

Table 3. Pin configuration between NodeMCU and Arduino Nano

<table>
<thead>
<tr>
<th>Pin NodeMCU</th>
<th>Pin Arduino Nano</th>
</tr>
</thead>
<tbody>
<tr>
<td>D5</td>
<td>D5</td>
</tr>
<tr>
<td>D6</td>
<td>D6</td>
</tr>
</tbody>
</table>

Figure 2. Hardware realization
Figure 3. Flowchart of TMR voter
2.2 Software Design

Software of the system as a TMR voter was designed following the flowchart mentioned in Figure 3. Program starts with setting the value of each existing token becomes 0, then Arduino Nano will read the recorded value X-axis from three MPU-6050 sensors X1, X2, and X3. The recorded results determine the value of each tokens according to the conditions, more than +8 or less than -8. The next process is voting based on the existing token value, “is there any token value which is worth more than one or voting majority?” otherwise the system will start process from the beginning. If the majority voting conditions are met then the next process is creating an emergency message. In addition, each sensor will also display the tilt direction based on the sensor output with the latest token value. If r>1 will display the direction to the right, and if l>1 will display the direction to the left, whereas if not both will return to the zero value again. The emergency message then be sent to the user in the form of email via IFTTT. After that, the program will restart the process from the beginning.

3 Results and Discussion

In general, this testing is conducted by manually pushing and then releasing the vehicle without a driver for a distance of 100 meters until an accident occurs. After the accident, the system will provide a notification in the form of tilt results to the e-mail. The following is the display in the e-mail for the tilt data on non-redundancy as shown in Figure 4(a) and e-mail notification of TMR as shown in Figure 4(b).

![Image](a)

![Image](b)

Figure 4. Display in the e-mail notification on (a) non-redundancy system and (b) TMR system

In this section, various tests are discussed, including sensor accuracy, reliability, and failure rate. Moreover, the ability of system to mask faults when utilizing TMR is an additional test. Firstly, sensor accuracy testing was conducted by comparing the tilt
direction data from the e-mail with the tilt direction data from the real location, then find the same tilt data. The measurement of sensor accuracy in (1) was obtained by calculation:

\[
\text{Accuracy} = \frac{\sum \text{correct data (e-mail & real)}}{\text{total data}} \times 100\% \quad (1)
\]

Secondly, failure rate refers to the rate at which failures occur in a system per unit of time, in this case, cycle. The test will be stopped when there was a failure in reading the data from the e-mail, even if accident occurs. The measurement of the failure rate test was obtained from the system reliability value approach first. The formulas for measuring the reliability of a system differ between non-redundancy and TMR. First, find the probability that the sensor will fail, \( F(t) \) in (2) [25]. The sensor is declared a failure if it does not send tilt direction data to the e-mail.

\[
F(t) = \text{Prob}\{T \leq t\} \quad (2)
\]

From (2), it can be assumed to calculate probability of failure in (3) by:

\[
F(t) = \frac{\text{number of failure}}{\text{total data}} \quad (3)
\]

After obtaining the probability of failure, the reliability \( R(t) \) that probability of single sensor or non-redundancy will survive before fail [25] can be calculated in (4) as follows:

\[
R(t) = 1 - F(t) \quad (4)
\]

However, the measurement of system reliability on TMR is different from single sensor or non-redundancy. The reliability of TMR system is determined by the assumption that at least two modules or sensors are functioning correctly. This assumption act as a perfect voter and the module failures occur independently of each other. Thus, the reliability of a TMR system can be expressed in (5) as follows:

\[
R_{TMR} = R_1 R_2 R_3 + (1 - R_1) R_2 R_3 + R_1 (1 - R_2) R_3 + R_1 R_2 (1 - R_3) \quad (5)
\]

\( R_1 R_2 R_3 \) represents the probability that all three modules operate correctly. \( (1-R_1)R_2R_3 \) represents the probability that the first module has failed, while the second and third module operate correctly. \( R_1(1-R_2)R_3 \) represents the probability that the first and third module operate correctly, while the second module has failed. \( R_1R_2(1-R_3) \) represents the probability that the first and second module operate correctly, while the third module has failed. After getting the reliability value, then the final measurement is the failure rate denoted by \( \lambda \) (lambda) [13] with formula in (6):

\[
\lambda = -\frac{\ln R(t)}{t} \quad (6)
\]

where \( t \) represents the time or cycle unit.

Thirdly, the testing of TMR method ability to mask faults aims to determine
whether this method can work well. This can be achieved by comparing the tilt direction data from the e-mail with the tilt direction data from the real location. The results of the sensor accuracy, reliability with failure rate tests are compared for both non-redundancy and TMR for further analysis. While, the result of ability of system to mask faults result test only for TMR.

### 3.1 Non-Redundancy Testing Result and Analysis

Table 4. Non-Redundancy result

<table>
<thead>
<tr>
<th>No. of test</th>
<th>X value</th>
<th>Tilt direction (e-mail)</th>
<th>Tilt direction (real location)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9.69</td>
<td>Right</td>
<td>Right</td>
</tr>
<tr>
<td>2</td>
<td>10.46</td>
<td>Right</td>
<td>Left</td>
</tr>
<tr>
<td>3</td>
<td>8.36</td>
<td>Right</td>
<td>Right</td>
</tr>
<tr>
<td>4</td>
<td>19.6</td>
<td>Right</td>
<td>Left</td>
</tr>
<tr>
<td>5</td>
<td>-8.56</td>
<td>Left</td>
<td>Left</td>
</tr>
<tr>
<td>6</td>
<td>-19.6</td>
<td>Left</td>
<td>Left</td>
</tr>
<tr>
<td>7</td>
<td>-12.14</td>
<td>Left</td>
<td>Left</td>
</tr>
<tr>
<td>8</td>
<td>9.48</td>
<td>Right</td>
<td>Right</td>
</tr>
<tr>
<td>9</td>
<td>-8.77</td>
<td>Left</td>
<td>Right</td>
</tr>
<tr>
<td>10</td>
<td>-19.6</td>
<td>Left</td>
<td>Left</td>
</tr>
<tr>
<td>11</td>
<td>9.38</td>
<td>Right</td>
<td>Left</td>
</tr>
<tr>
<td>12</td>
<td>9.68</td>
<td>Right</td>
<td>Left</td>
</tr>
<tr>
<td>13</td>
<td>9.40</td>
<td>Right</td>
<td>Right</td>
</tr>
<tr>
<td>14</td>
<td>-</td>
<td>-</td>
<td>Right</td>
</tr>
</tbody>
</table>

All parameter of the test results on non-redundancy are based on Table 4. In non-redundancy testing, the number of tests obtained 14 cycles. This is because data can no longer be sent to email on the 14th cycle, despite the fact that there was an accident.

The measurement of sensor accuracy in non-redundancy test was obtained by calculation that refers to (1), yielding:

\[
\text{Accuracy} = \frac{8}{14} \times 100\% = 57\%
\]

In addition, to obtain a measurement of the failure rate in non-redundancy test (6), it is essential to calculate probability of failure (3) and single sensor reliability (4) as follows:

\[
F(t) = \frac{1}{14} = 0.0714
\]

\[
R(t) = 1 - 0.0714 = 0.9286
\]

\[
\lambda = -\frac{\ln(0.9286)}{14} = 0.0053/\text{cycle}
\]
3.2 Triple Modular Redundancy Testing Result and Analysis

Table 5. Triple Modular Redundancy result

<table>
<thead>
<tr>
<th>No. of test</th>
<th>X1 value</th>
<th>X2 value</th>
<th>X3 value</th>
<th>Tilt direction (e-mail)</th>
<th>Tilt direction (real location)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-19.6</td>
<td>-19.6</td>
<td>-19.6</td>
<td>Left</td>
<td>Left</td>
</tr>
<tr>
<td>2</td>
<td>19.6</td>
<td>9.7</td>
<td>9.7</td>
<td>Right</td>
<td>Right</td>
</tr>
<tr>
<td>3</td>
<td>2.99</td>
<td>11.39</td>
<td>11.39</td>
<td>Right</td>
<td>Right</td>
</tr>
<tr>
<td>4</td>
<td>10.15</td>
<td>9.39</td>
<td>9.39</td>
<td>Right</td>
<td>Right</td>
</tr>
<tr>
<td>5</td>
<td>-19.6</td>
<td>7.73</td>
<td>7.73</td>
<td>Left</td>
<td>Left</td>
</tr>
<tr>
<td>6</td>
<td>10.07</td>
<td>9.33</td>
<td>9.33</td>
<td>Right</td>
<td>Right</td>
</tr>
<tr>
<td>7</td>
<td>19.6</td>
<td>9.9</td>
<td>9.9</td>
<td>Right</td>
<td>Right</td>
</tr>
<tr>
<td>8</td>
<td>-9.36</td>
<td>-9.97</td>
<td>-9.97</td>
<td>Left</td>
<td>Left</td>
</tr>
<tr>
<td>9</td>
<td>-19.6</td>
<td>4.13</td>
<td>4.13</td>
<td>Left</td>
<td>Left</td>
</tr>
<tr>
<td>10</td>
<td>10.08</td>
<td>9.35</td>
<td>9.35</td>
<td>Right</td>
<td>Right</td>
</tr>
<tr>
<td>11</td>
<td>12.97</td>
<td>13.59</td>
<td>13.59</td>
<td>Right</td>
<td>Left</td>
</tr>
<tr>
<td>12</td>
<td>11.69</td>
<td>13.95</td>
<td>13.95</td>
<td>Right</td>
<td>Right</td>
</tr>
<tr>
<td>13</td>
<td>10.26</td>
<td>9.47</td>
<td>9.47</td>
<td>Right</td>
<td>Right</td>
</tr>
<tr>
<td>14</td>
<td>19.6</td>
<td>15.17</td>
<td>15.17</td>
<td>Right</td>
<td>Right</td>
</tr>
<tr>
<td>15</td>
<td>-9.16</td>
<td>-9.8</td>
<td>-9.8</td>
<td>Left</td>
<td>Right</td>
</tr>
<tr>
<td>16</td>
<td>-19.6</td>
<td>-10.16</td>
<td>-10.16</td>
<td>Left</td>
<td>Right</td>
</tr>
<tr>
<td>17</td>
<td>12.73</td>
<td>9.63</td>
<td>9.63</td>
<td>Right</td>
<td>Right</td>
</tr>
<tr>
<td>18</td>
<td>13.36</td>
<td>7.8</td>
<td>7.8</td>
<td>Right</td>
<td>Right</td>
</tr>
<tr>
<td>19</td>
<td>-9.24</td>
<td>-10.01</td>
<td>-10.01</td>
<td>Right</td>
<td>Right</td>
</tr>
<tr>
<td>20</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Left</td>
</tr>
</tbody>
</table>

All parameter of the test results on TMR are based on Table 5. In TMR testing, the number of tests obtained 20 cycles. This is because data can no longer be sent to email on the 20th cycle, despite the fact that there was an accident.

The measurement of sensor accuracy in TMR test was obtained by calculation that refers to (1), yielding:

\[
Accuracy = \frac{16}{20} \times 100\% = 85\%
\]

In addition, to obtain a measurement of the failure rate in TMR test (6), it is essential to calculate probability of failure (3), single sensor reliability (4), and also TMR reliability (5) as follows:
\[ F(t) = \frac{1}{20} = 0.05 \]
\[ R(t) = 1 - 0.05 = 0.95 \]

Because of all modules have the same values \( R_1 = R_2 = R_3 = R \) with the value 0.95, then the (5) will derived as follows:

\[
R_{TMR} = R^3 + (1-R)R^2 + R(1-R)R + R^2(1-R)
\]
\[
= R^3 + R^2 - R^3 + R^2 - R^3 + R^2 - R^3
\]
\[
= 3R^2 - 2R^3
\]
\[
= 3(0.95)^2 - 2(0.95)^3
\]
\[
= 0.9928
\]

Then,
\[
\lambda = -\frac{\ln(0.9928)}{20} = 0.0004 / \text{cycle}
\]

Lastly, the testing of the TMR method ability to mask faults can be seen in the anomalies data that occurred in the 3rd, 5th, 9th, and 18th test cycles. These are considered anomalies because the value on one of the sensors was outside the specified range, specifically, values less than -8 or less than +8. However, after tilt data matching between the e-mail and the real location was conducted, the results were the same.

4 Conclusion

A vehicle accident report system with TMR method implementation was designed and examined. Experiment results confirm that implementation of TMR method can increase sensor accuracy, reduce failure rate, and increase reliability. The TMR system also can send a warning message to a determined email address both in normal and anomaly situations.

In the future, additional features, such as notification for android smartphone and location coordinate sending will be developed. Development of the system using other type of sensors will also be considered as future work.

Acknowledgments. This work is supported by Faculty of Computer Science, Universitas Brawijaya through Program Kegiatan Penelitian Dana DIPA tahun 2023.

References


