ACCESSING REAL-TIME AUGMENTED REALITY SIMULATION USING IoT CLOUD CONNECTION ON MATERIAL HANDLING CONVEYOR AS A CASE STUDY

Siti Aisyah Jaafar[1,*], Sukarnur Che Abdullah[2], Ahmed Jaffar[3], Farok Hj. Azmat[4]

[1]*Faculty of Mechanical Engineering, Universiti Teknologi MARA, Shah Alam, Malaysia*

[2]*Faculty of Mechanical Engineering, Universiti Teknologi MARA, Shah Alam, Malaysia*

[3]*Faculty of Mechanical Engineering, Universiti Teknologi MARA, Shah Alam, Malaysia*

[4]*Faculty of Computer and Mathematical Science, Universiti Teknologi MARA, Shah Alam, Malaysia*

[*]For correspondence; Tel. + (60) 89111366, E-mail: sitiaisyahjaafar9@gmail.com

ABSTRACT: *Conventional AR-IoT applications only specialize in the localized area for the on-site application that visualizes data using textual environments which do not promote the user's sensorial perception and interaction. An experimental investigation was conducted to explore the ability of AR as support to off-site IoT monitoring data visualization, called "AR-Simulasi". The main purpose of this research aimed to develop a remote monitoring device using an AR visualization approach. The developed remote monitoring is applied to an algorithm to remote AR by using a monitoring application system and the efficiency of the AR-Simulasi Apps was analyze. The results reveal that the efficiency in data transfer is 10.4% based on the time delay comparison. Referring to multiple previous studies, any IoT devices with a time delay of less than 250ms would be considered to be successfully established and reliable for further utilization. Based on the results of the efficiency analysis, AR-Simulasi has been proven to be a reliable AR-IoT application.*

Keywords: Augmented Reality, Internet of Things, Cloud, Real-Time, Simulation

1. INTRODUCTION

The embodiment of Augmented Reality (AR) for the industrial revolution 4.0 (IR4.0) has got a lot of attention over the last decade as a medium that is scalable, flexible, and allows for more human-digital interactions [1–3]. Because of its versatility and intelligent interface, Virtual Reality, and the Internet of Things (IoT) have sparked interest in many industrial sectors in recent technology [4]. AR is also coming into view in terms of its applicability in assistive systems, thanks to growing scientific and commercial interest and regular launches of new products [5–7].

The development of AR-IoT incorporates Augmented Reality (AR) and Internet of Things (IoT) systems. This application of AR-IoT is partially a continuation of the previously adapted IoT system for temperature detection smartphone devices [8]. Researchers have used AR technology in a few environments, including education, farming, manufacturing, and more. AR has previously been used for entertainment purposes, such as social media filters, gaming, and commercials. With advances in ICT technology, modern devices with higher resolution and faster response times can be used to produce reliable results in an efficient space [1, 9].

For off-site IoT monitoring, the integration with AR technology is still minimum. One example is in "IoT-monitoring", where the use of textual components such as texts, numbers, and graphical information to visualize data [10]. IoT monitoring systems were also introduced by healthcare professionals to replace ward rounds and use smart sensors and smartphone devices to track patients in the hospital to save time and receive emergency alerts faster [3, 11–13]. Aside from that, IoT monitoring has become increasingly popular in recent years for a variety of purposes, including agriculture, healthcare, manufacturing, and education [10, 14–16]. Most AR-IoT applications depend on image processing. In this case, the user must scan using camera projection, and the information will be superimposed over the scanned object in their devices' AR area [17].

By developing an AR simulation, AR was integrated into off-site IoT monitoring in this report. AR simulation, by definition, is a real-time animation that is in harmony with the motions of a real machine. The contact between smart sensors, cloud communication, and the apps activated the simulation's movement engagements. This necessitates the use of MQTT to communicate between various devices [18].

The main aim of this paper is to see whether an off-site AR-IoT monitoring application can be developed instead of an on-site one. Aside from that, the development of the application shall be applied with the coding algorithm to analyze the efficiency of the AR Simulation application.

2. METHODOLOGY

The development of the system involves several sections. The first section is the data transfer between the real process and the cloud. While the second section would be data transfer between the cloud and the AR virtual process, where the simulation of AR can be observed. To execute the algorithm of data transfer, it must first be tested on a relevant case study. The case study chosen for this AR-IoT simulation is a simple conveyor for the color sorting process. This case study was chosen due to the simplicity of this process to able to re-create it like a virtual simulation of the real process in AR projection in the next section of this paper. The setup of the conveyor machine for the case study is as in figure (1) below.

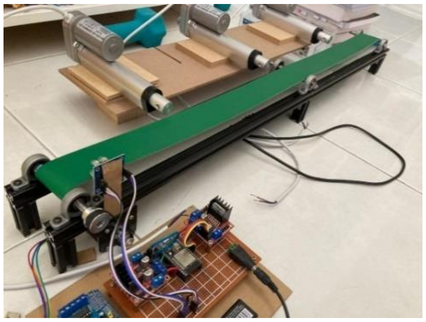


Fig (1) Material handling process as a case study

Figure (1) shows a conveyor machine that was used for the color sorting system. It has been programmed to sort out wooden colored cubes into three different color groups, which were red, yellow, blue, and unknown. The color sorting process was developed using the ESP32 itself, with the help of the TSC3200 color sensor.

Data transfer between real process and cloud was established initially by re-designing the real machine, which is as in figure (1), in CAD form. The CAD version of the machine is then converted to .obj format to enable it to be imported into Unity 3D software. Unity 3D software is the tool that was used to create the AR Simulation of the real process, herewith called the virtual process.

The proposed setup of the IoT devices is by using a TSC3200 color sensor for color sensing. The microcontroller used to collect data from the color sensor is ESP32. To integrate linear actuators, motor drivers needed to be included in the setup. The full setup is as in figure (2) below.

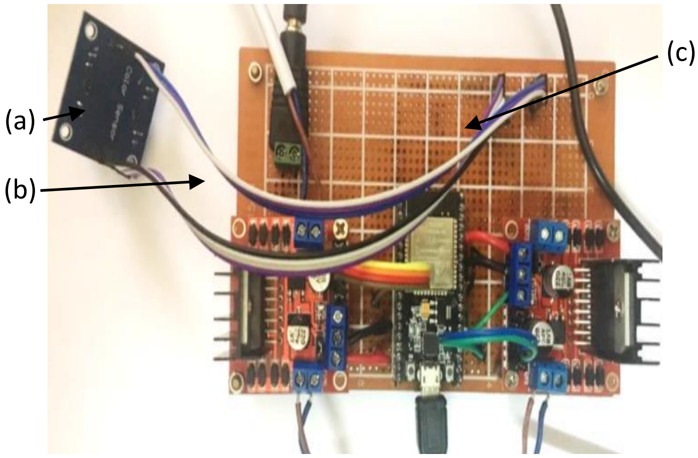


Fig (2) IoT devices setup on the machine

Figure (2) shows the IoT devices that include (c) ESP32 as a microcontroller, (a) TSC3200 color sensor, and (b) motor drivers were attached to the real process to obtain the real-time data of the process while it was running. This IoT setup was chosen due to the

While the framework for the connection between the real conveyor process and the cloud system was then developed. The algorithm was separated into two mediums, which is the real conveyor process and the second medium is the ESP32 with cloud data transfer. The first medium shows the algorithm for color sorting system in real conveyor process, and how it transmits its data of color detected to the external source. There were two exported data which are from the color sensor, and another is from the linear actuator signals. The setup can be seen as in figure (3). The information sent from the first medium was received and stored by the cloud, using the MQTT communication scheme as in figure (4) below.

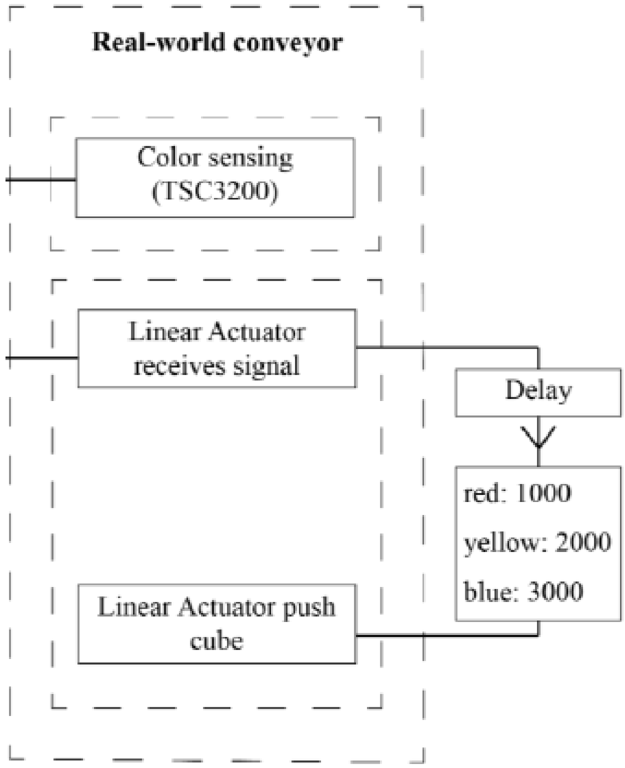


Fig (3) Schematic framework of real-world conveyor

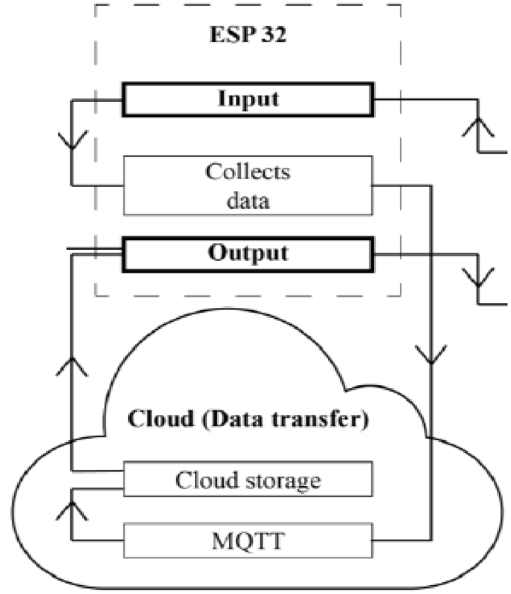


Fig (4) Schematic framework of ESP32 and cloud storage when data were received from real word conveyor.

The framework was used as a benchmark to create the programming algorithm. The programming algorithm is needed to establish the system proposed by the framework. The programming uses the C++ language in Arduino IDE software. Both programming for the color sorting process and external cloud data transfer were programmed using this method. Therefore, the algorithm that initiates the establishment of the data transfer is as in figure (5) below.

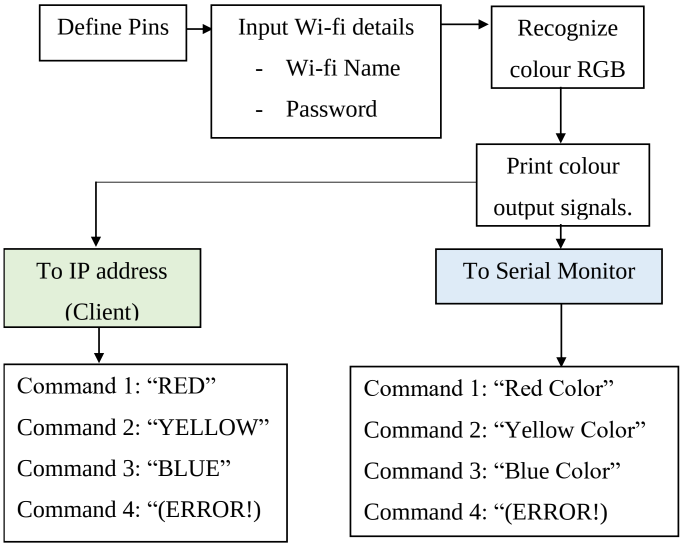


Fig (5) Algorithm of the real process to cloud storage data transfer

Figure (5) shows the flow of data transfer from the real world conveyor process, to the cloud. It begins with the definition of input pins, where the color sensor and microcontroller were plugged into those recognized pins. Then, the internet connection details were needed to be inserted into the code. The development process proceeds with the testing and analysis of the connection. To do so, the connection was tested during the data transfer, to see if the data were sent from the real process to the cloud system, in the AWS-IoT cloud server.

Fig (6) Algorithm of the real process to cloud storage data transfer

The diagram in figure (6) shows the observable connectivity from the AWS-IoT server. It shows the established connection between ESP32 and the cloud AWS system, by using MQTT as the means of communication between the two mediums.

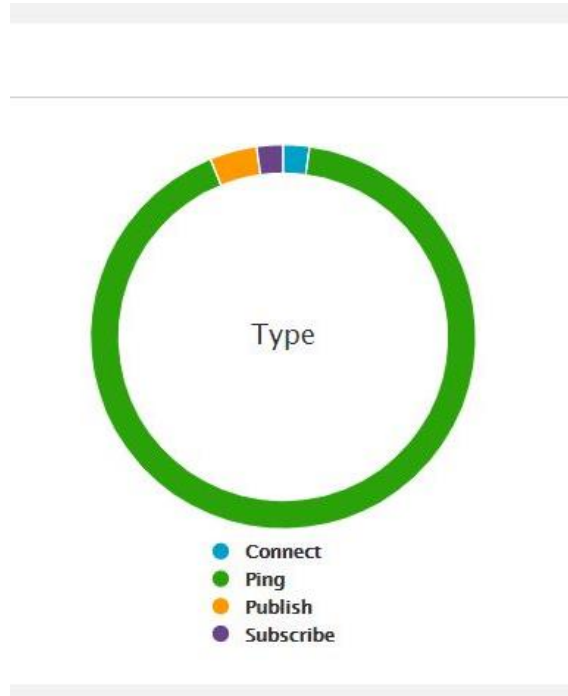


Fig (7) Algorithm of the real process to cloud storage data transfer

Diagram in figure (7) above shows that the data transfer was established, by being able to see the "Connect" and "Ping" as in the figure. The yellow dot stands for the "Publish" which means the data from ESP32 were published in the cloud AWS system, and thus, the purple dot suggests that the two mediums were successfully subscribed to each other for further data transfer.

Once the connection between the first two mediums was successful, the development of the 2nd and 3rd mediums was established. In which, the data transfer between the cloud service, to the AR Simulation process.

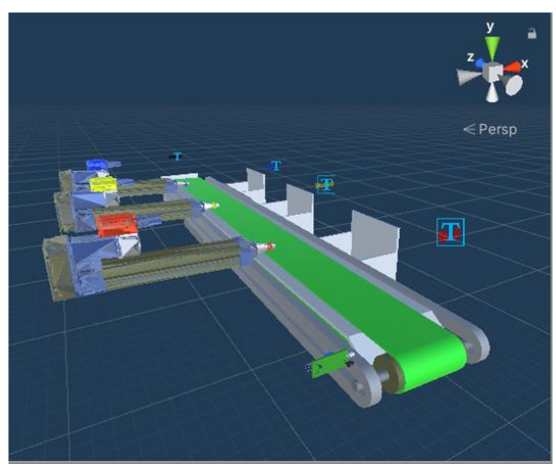


Fig (8) Scene preview of virtual conveyor in Unity with vector points visualization

Figure (8) above shows the imported CAD design of the material handling machine, into the Unity scene for simulation development. The simulation development involves the integration between simple animation for movements, and also programming algorithms for receiving real-time data and performing the animations, which is the most crucial part of the AR simulation.

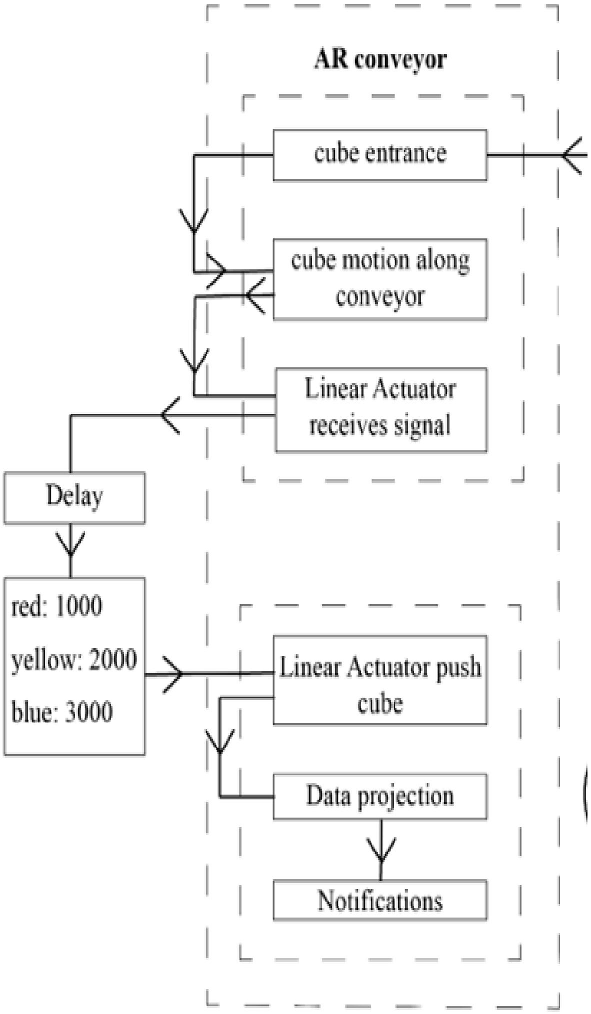


Fig (9) Algorithm of the real process to cloud storage data transfer

The programming algorithm is as in figure (9) above. Which is the continuation of the framework in figure (4) from the cloud data transfer section. Therefore, it explains that the real-time data obtained in the cloud were then transferred into Unity sketch, where the AR simulation was built.
Marker is the target image for the projection of the AR virtual conveyor in Unity [19]. AR marker is a two-dimensional image that is used to trigger augmented reality experiences. It has been used to monitor, orient, and position augmented reality projections, enabling users to place AR objects and contents in the real world. Instead of the greyish stuff, multiple colors can now be set as the AR marker in the latest version of AR Base.

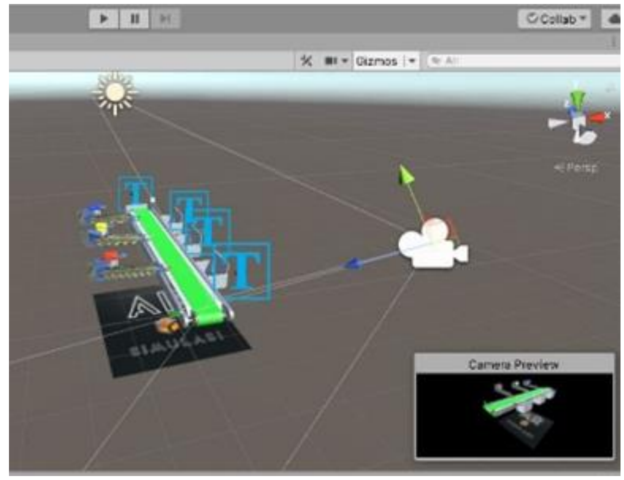
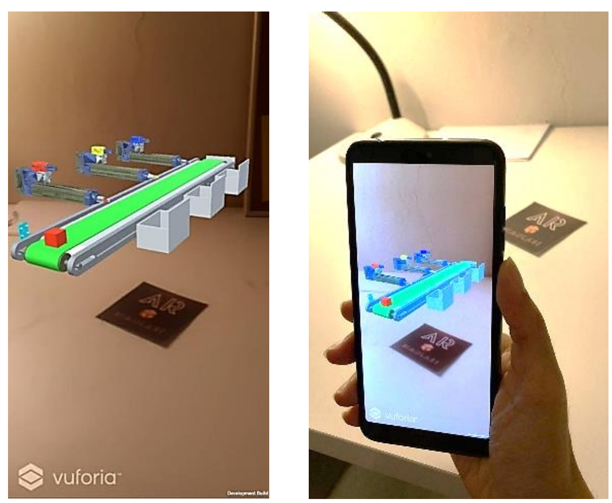


Fig (10) AR marker imported in Unity Scene

Figure (10) above shows the AR marker used for the projection of AR simulation. The AR marker was created in the vuforia website, and then the unique code was copied into the build section in Unity 3D software to retrieve the marker in the scene as shown above.
The final stage of the development is then by exporting the project as a smartphone application. The OS used for this paper is android, due to the developmental environment which is windows OS, therefore it is easier to be exported to android. However, it can also be exported for IOS and other OS as well.

3. RESULTS AND DISCUSSION

The development of real-time AR Simulation was initiated by the development of the AR Simulation Apps as in figure (11) below.



(a) (b)

Fig (11) (a) First-person and (b) second-person point of view of AR Simulation Apps from an Android phone

The screenshot in Figure (11) is from the Android phone that was used to run the AR Simulation Apps. The device's camera was used to scan the AR marker. Then, on top of the AR marker, the virtual conveyor appeared. When using the apps, the AR appearance was projected on top of the AR marker as desired. On the phone's screen, the AR conveyor was clearly visible.

For each operation that occurs during the color sorting process, a collection of timestamps was obtained. In the Unity console and the IDE serial monitor, timestamps were taken for each event. The data is then compared to real-world conveyor timing that was captured when watching the mechanism in action. The virtual conveyor's Apps data is represented by the Unity console, while the rawest IoT monitoring data is represented by the data from the IDE serial display. Due to its direct link to ESP32, which controls the sensors and linear actuators, IDE monitor data is the baseline IoT monitoring data. As a result, commands from the IDE monitor data will be received by any IoT monitoring program.

Table (1) Table of real-time timestamps data recorded

Qty	IDE Console	Unity Console	Delay (ms)
1	14:16:18:712	14:16:18:920	208
2	14:16:25:520	14:16:25:750	230
3	14:16:35:689	14:16:35:956	230
4	14:16:41:685	14:16:41:885	200
5	14:16:50:520	14:16:50:720	200
6	14:16:57:720	14:16:57:948	228
7	14:17:07:625	14:17:07:958	240
8	14:17:14:723	14:17:14:925	202
9	14:17:40:712	14:17:40:925	213
10	14:17:45:625	14:17:45:850	225
11	14:17:53:698	14:17:53:902	204
12	14:18:00:723	14:18:00:925	202
13	14:18:07:625	14:18:07:852	227
14	14:18:15:520	14:18:15:820	230
15	14:18:29:720	14:18:29:923	203
16	14:18:36:625	14:18:36:855	230
17	14:18:43:720	14:18:43:951	231
18	14:18:50:723	14:18:50:956	233
19	14:18:57:712	14:18:57:927	215
20	14:19:15:698	14:19:15:987	237
Average Delay (ms)			**219**

Table (1) shows the recorded real-time data during the color sorting process in the real process of the material handling machine. There were two sets of timestamps that were recorded which are from the real process and from the virtual AR simulation process. The data from the IDE console represents the real process, while the data from the Unity console represents the virtual AR simulation process. Therefore, to investigate the delay of data transfer from real to virtual process using this Apps, the time delay was calculated by deducting real process time from virtual process time. From the resulting timestamps that were collected during the sorting of colored cubes, the time delay was calculated.

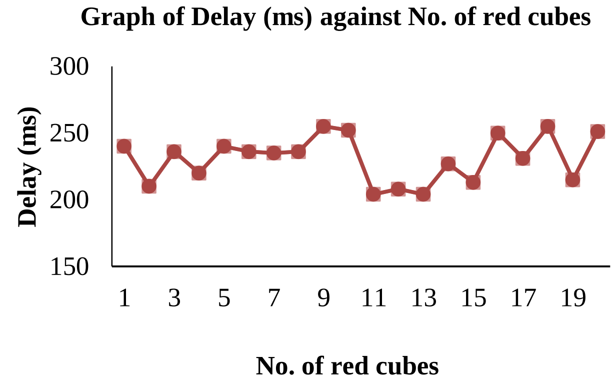


Fig (12) Graph of Delay (ms) against no. of red cubes

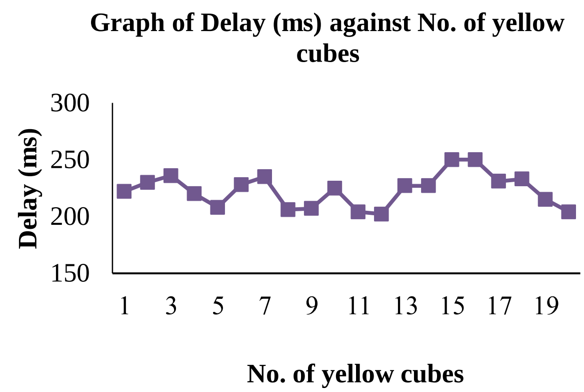


Fig (13) Graph of Delay (ms) against no. of yellow cubes

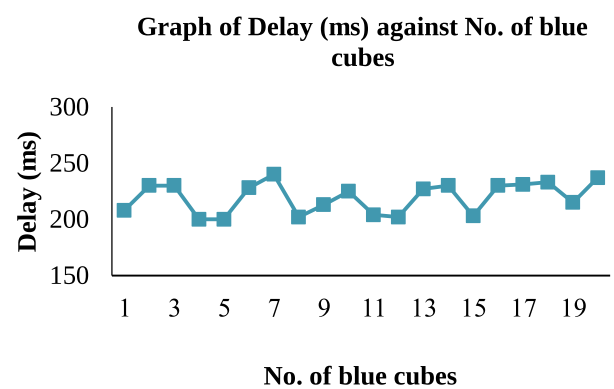


Fig (14) Graph of Delay (ms) against no. of blue cubes

It is interesting to note that all three tests for three different colored cubes in this study show the average time delay that is lower than the Standard Time Delay. Thus, from the graph in figure (12), figure (13), and figure (14), the respective average time delay obtained were as follows:

$$Delay_{blue\_avg} = 219\ ms \quad (1)$$

$$Delay_{yellow\_avg} = 223\ ms \quad (2)$$

$$Delay_{red\_avg} = 231\ ms \quad (3)$$

Therefore, the average time delay can be calculated as below:

$$Delay_{avg} = \frac{\left(Delay_{blue_{avg}} + Delay_{yellow_{avg}} + Delay_{red_{avg}}\right)}{3} ms$$
$$= \frac{(219 + 223 + 231)}{3} ms$$
$$= 224.3333\, ms \qquad \cong 224\, ms$$

The value of $Delay_{avg}$ was rounded off from $224.3333\, ms$ to $224\, ms$. Thus, by comparing the Standard Time Delay which is 250ms, to the calculated $Delay_{avg}$ which is 224ms, it has been shown that the value of average delay for the AR Simulation Apps is considered acceptable in the means of data transfer efficiency in IoT applications. Consequently, this result reflects the objectives for this research paper in which to investigate the efficiency of the AR Simulation application. However, the value of delay for AR-Simulasi Apps had shown that the value is near to the Standard Maximum Delay:

$$\frac{250ms - 224ms}{250ms} \times 100\% = 10.4\%$$

4. CONCLUSIONS

The remote monitoring device was developed by using the AR visualization approach. This was established by applying the programming algorithm to the connection between several mediums. The mediums involved in this research paper were consisting of three sections, which were the real process, the cloud service, and the virtual AR process. The data transfer between each of these mediums was processed by the real-time data collection that was done during the process. However, to achieve the stable data values, the process was repeated several times until a stable value between 200ms to 250 ms was to be achieved.

The findings of this research paper suggest that the virtual conveyor shall be reprogrammed in future research, so it follows the frame area of the camera to prevent the AR projection from disappearing during the monitoring process. When the camera gets too close to the AR marker, for example, the virtual conveyor vanishes. There have been many attempts to solve the problem in previous researches. This problem could interrupt the monitoring process if the phone camera is not correctly positioned to catch the AR marker at a particular distance. Despite the issue, the color sorting process continued as normal after the virtual conveyor vanished and reappeared on the computer.

From the results obtained, the average time delay, $Delay_{avg}$, was compared with the Standard Time Delay for real-time IoT devices. The value of Standard Time Delay is $250ms$, as mentioned in a previous study on automation and control by Brandt, A. in his publication journal in 2015 [20].

It has shown from the time delay that the efficiency in percentage is 10.4% for data transfer between real process and virtual process, with only a 10.4% difference from the Standard Maximum Delay value. Since the value of delay is near the maximum permitted delay value, this issue ought to be improved in the future to prevent extended delay time due to poor connections and other possible technical and wireless issues. Therefore, the high value of delay for the application may possibly be due to several main causes in which some of them is due to many constrained nodes with limited processing power and memory. The routers are linked by unstable connectors that normally support only low data transfer rates, resulting in low transmission rates. Thus, to handle these issues, further machine learning shall be done in future research endeavors within this field of study.

5. REFERENCES

[1] K. Lavingia and S. Tanwar, “Augmented Reality and Industry 4.0,” pp. 143–155, 2020.

[2] R. Hellmuth, J. Frohnmayer, and F. Sulzmann, “Design and Application of a Digital Factory Model for Factory Restructuring,” *Procedia CIRP*, vol. 91, pp. 158–163, 2020.

[3] B. J. Park, S. J. Hunt, C. Martin, G. J. Nadolski, B. J. Wood, and T. P. Gade, “Augmented and Mixed Reality: Technologies for Enhancing the Future of IR,” *Journal of Vascular and Interventional Radiology*, vol. 31, no. 7. Elsevier Inc., pp. 1074–1082, 01-Jul-2020.

[4] K. van Lopik, M. Sinclair, R. Sharpe, P. Conway, and A. West, “Developing augmented reality capabilities for industry 4.0 small enterprises: Lessons learnt from a content authoring case study,” *Comput. Ind.*, vol. 117, May 2020.

[5] L. F. de Souza Cardoso, F. C. M. Q. Mariano, and E. R. Zorzal, “A survey of industrial augmented reality,” *Comput. Ind. Eng.*, vol. 139, no. October 2019, p. 106159, 2020.

[6] “Six Top Tools to Build Augmented Reality Mobile Apps,” 2020. [Online]. Available: https://www.infoq.com/articles/augmented-reality-best-skds/. [Accessed: 10-Feb-2021].

[7] V. Liagkou, D. Salmas, and C. Stylios, “Realizing Virtual Reality Learning Environment for Industry 4.0,” *Procedia CIRP*, vol. 79, pp. 712–717, 2019.

[8] S. A. Jaafar and S. C. Abdullah, “Development of Monitoring System using the Internet of Things for Industrial Revolution 4.0,” *Int. J. Innov. Technol. Explor. Eng.*, vol. 9, no. 3, pp. 2790–2794, 2020.

[9] M. Chylinski, J. Heller, T. Hilken, D. I. Keeling, D. Mahr, and K. de Ruyter, “Augmented reality marketing: A technology-enabled approach to situated customer experience,” *Australas. Mark. J.*, May 2020.

[10] R. Mehta, J. Sahni, and K. Khanna, “Internet of Things : Vision , Applications and Challenges,” *Procedia Comput. Sci.*, vol. 132, pp. 1263–1269, 2018.

[11] S. Rajkumar, M. Srikanth, and N. Ramasubramanian, “Health monitoring system using Raspberry PI,” *2017 Int. Conf. Big Data, IoT Data Sci. BID 2017*, vol. 2018-Janua, no. 1, pp. 116–119, 2018.

[12] N. Mani, A. Singh, and S. L. Nimmagadda, “An IoT Guided Healthcare Monitoring System for Managing Real-Time Notifications by Fog Computing Services,” in *Procedia Computer Science*, 2020, vol. 167, pp. 850–859.

[13] B. Almadani, M. Bin-Yahya, and E. M. Shakshuki,

"E-AMBULANCE: Real-time integration platform for heterogeneous medical telemetry system," in *Procedia Computer Science*, 2015, vol. 63, pp. 400–407.

[14] S. M. Satapathy, R. Jhaveri, U. Khanna, and A. K. Dwivedi, "Smart Rent Portal using Recommendation System Visualized by Augmented Reality," in *Procedia Computer Science*, 2020, vol. 171, pp. 197–206.

[15] I. Ali, E. Khan, and S. Sabir, "Privacy-preserving data aggregation in resource-constrained sensor nodes in Internet of Things: A review," *arXiv*, vol. 3, no. 1, pp. 41–50, 2018.

[16] S. M. Satapathy, R. Jhaveri, U. Khanna, and A. K. Dwivedi, "Smart Rent Portal using Recommendation System Visualized by Augmented Reality," *Procedia Comput. Sci.*, vol. 171, no. 2014, pp. 197–206, 2020.

[17] J. Egger and T. Masood, "Augmented reality in support of intelligent manufacturing – A systematic literature review," *Computers and Industrial Engineering*, vol. 140. Elsevier Ltd, p. 106195, 01-Feb-2020.

[18] C. Patel and N. Doshi, "'a Novel MQTT Security framework in Generic IoT Model,'" in *Procedia Computer Science*, 2020, vol. 171, pp. 1399–1408.

[19] Z. Ye, W. JianQin, and L. ZhiHong, "An Intelligent Display Platform of Traditional Farming Myth's Virtual Scene Based on Unity3D," *IFAC-PapersOnLine*, vol. 51, no. 17, pp. 678–682, 2018.

[20] A. Brandt, E. Baccelli, R. Cragie, and P. Van der Stok, "Applicability statement: The use of the RPL protocol suite in home automation and building control.," pp. 1–32, 2015.

[21] N. U. Okafor, Y. Alghorani, and D. T. Delaney, "Improving Data Quality of Low-cost IoT Sensors in Environmental Monitoring Networks Using Data Fusion and Machine Learning Approach," *ICT Express*, vol. 6, no. 3, pp. 220–228, 2020.

*For correspondence; Tel. + (60) 189111366, E-mail:sitiaisyahjaafar9@gmail.com