MALAYSIAN JOURNAL OF INNOVATION IN ENGINEERING AND APPLIED SOCIAL SCIENCE (MyJIEAS)
e-ISSN : 2811-3780
VOLUME 04, ISSUE 01

InventsimQD: Graphical User Interface (GUI) Based Teaching Application for Quantity Discounts Inventory Models

Mohd Fauzi Ismail[1*], Nursalbiah Nasir[1], Wan Emri Wan Abdul Rahaman[1], Amirul Abd Rashid[1] and Mohd Hazri Mohd Rusli[1]

[1]School of Mechanical Engineering, College of Engineering, Universiti Teknologi MARA Malaysia, 40450 UiTM Shah Alam, Selangor, Malaysia

Corresponding Author's Email: mohdfauzi305@uitm.edu.my

Article History: Received 13082024; Revised 20082024; Accepted 27092024;

ABSTRACT – The rapid growth of computer software applications proves highly beneficial as a teaching tool for course delivery, particularly in its interactive nature compared to traditional lecture notes. This paper introduces InventSimQD, a graphical user interface (GUI) application developed for simulating and analyzing the quantity discount model. InventSimQD allows users to visualize the impact of key variables in the quantity discount model and facilitates simultaneous comparison of various discount offers, thereby enhancing students' grasp of fundamental concepts. The development of this simulator began with a detailed analysis of user requirements. By generating total cost curves across different discount values, the application provides clear insights into optimal offer selection based on volume requirements. MATLAB serves as the primary platform for compiling both the algorithm and the user interface. One significant advantage of InventSimQD is its compatibility with any personal computer through MATLAB Runtime, allowing installation without an official license. The accuracy of the application's outputs has been validated through theoretical calculations and supports multiple concurrent sessions for real-time comparison of quantity discount scenarios. Consequently, InventSimQD empowers students to gain a comprehensive understanding of the quantity discount model and equips them with the insights needed to effectively choose discount offers that minimize total costs.

KEYWORDS: Quantity discounts, economic order quantity, inventory model, GUI, teaching and learning

1.0 INTRODUCTION

Inventory is one of the most expensive assets as it consumes as much as 50% of invested capital for industries, particularly the manufacturing and engineering related industry. Thus, according to Heizer, knowledge on managing inventory is crucial [1]. Hence the fundamentals of inventory management are normally taught in Industrial Management subject. The first step to learn inventory management is to understand inventory models. Inventory model is a mathematical or computational tool used by businesses to determine the optimal level of inventory to maintain in relation to independent or dependent demand. Among the models for independent demand is quantity discount model.

The quantity discount model examines the effect of price incentives given in the case of larger quantities of products ordered. The cost of ordering inventory is not fixed, rather it decreases as the order quantity increases due to volume discounts offered by suppliers. The goal is to find the order quantity that minimizes the total inventory cost (Tc), considering both the inventory holding costs (H) and the ordering costs (S), while taking advantage of available quantity discounts. Unlike the basic economic order quantity model (EOQ), the holding cost (H) is calculated as percentage (I) of unit price (P). The total inventory cost, Tc is calculated as in Equation (1) where D is annual demand and Q is order quantity.

$$T_C = \frac{DS}{Q} + \frac{QH}{2} + PD \tag{1}$$

The optimal order quantity (Q*) is obtained from Equation (2)

$$Q^* = \sqrt{\frac{2DS}{IP}} \qquad (2)$$

Figure 1 exemplifies the relationship of calculated Q* and the total cost based on available discount options, which would help students to determine the best feasible price and order quantities. If Q* falls under x value, price would be normal while feasible total cost of discount price 1 would only valid if Q* range is between x and y. If Q* is more than y, discount price 2 is preferred.

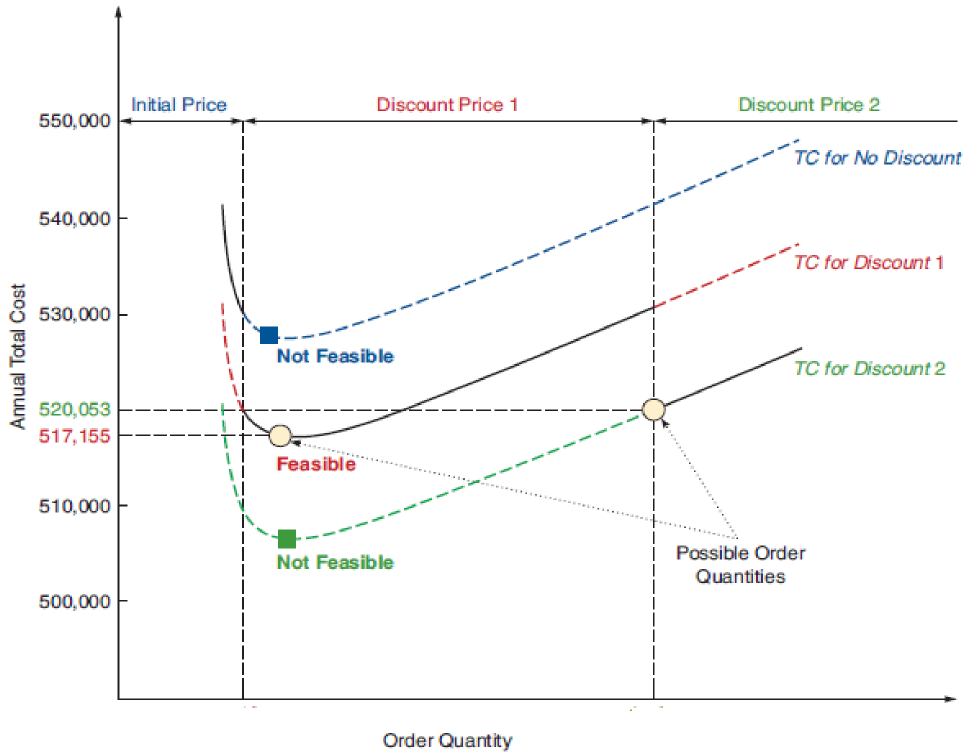

Figure 1. The total-cost curve with quantity discounts [1]

As mentioned by Chang et al and Lent, the rapid growth of Graphical user interface (GUI) applications which provides visual information to its user [7- 8]. GUI could be done by creating the program coding using the general-purpose programming language such as Python and JavaScript. GUI has been applied for many purposes and areas such as surgery assistance done by Md Jan et al [6]. While Alves et al developed GUI based virtual reality game [12]. Other than that, GUI researches related to sports engineering such as lawn bowl trajectory which was done by Medwell et al [14] is also applicable. As GUI delivers information visually, it could be used as a teaching and learning tool. For industrial management course, it could help students to obtain clear and in depth understanding of inventory models and use the information to determine optimum economic order quantities faster. Although quantity discount model calculators could be found on the internet, existing calculators only shows the final answer for Tc and Q*one at a time. In addition, students could not compare and see visually the effect of changing Tc parameters values (S, I, P, D). With the use of applications, it is expected that the students would understand the concept and can use the information to make decisions on optimum EOQ in answering related course assessments.

Based on study by Kim & Kim [9], GUI can also be created by using MATLAB, a high-performance language programming platform specifically for engineers and scientists. MATLAB is a high-level programming language and interactive environment primarily designed for numerical computation, visualization, and programming used in various fields, including engineering, science, mathematics, finance, and beyond. The simplicity and free access to obtain the license has made the MATLAB to be preferred language for developing teaching apps for Science, Technology, Engineering and Mathematics (STEM) subjects such as engineering mathematics based subject as reported by Khedekar et al [2], power system subject by Koc & Aydoğmus' and Nordin et al.[10,13], electromagnetic by Wong & Lim' study [11], thermodynamics subject by Domínguez et al[3], dynamics course by Rossiter[4], information and control curriculum subject as reported by Karel & Tomas [15] and renewable energy courses by Gopabala Krishnan et al. [5]. Gopabala Krishnan et al. concluded that although energy computator (EC) -GUI did not reduce the difficulties of the subject, it did help to enhance the ways to learn more efficiently. Based on students' feedback in the studies, all agreed that apps are very useful in encouraging active independent learning.

Nonetheless, recent mechanical engineering courses do not only contain pure engineering courses. Entrepreneur and management related courses such as industrial management are now embedded in engineering syllabus so that it can prepare students to not only be engineers but to manage business and become entrepreneurs after graduation. Thus, the main goal of this study is to develop interactive MATLAB applications, namely InventsimQD for teaching quantity discount models in a compulsory industrial management course for Mechanical Engineering undergraduate students. The InventSimQD would allow users/students to rapidly calculate output parameters (quantity discount EOQ and total cost) based on adjustable input parameters and visually compare few scenarios simultaneously from total cost price range graph which is the InventSimQD output.

2.0 METHODOLOGY

2.1 Establish design specification of InventSimQD

The design requirements of InventSimQD consist of input parameters,output parameters, visual output and ease of use. For the input parameters, all inputs could be manipulated to simulate various conditions. The first set of input parameters were quantity demands (D), setup cost (S), holding cost per unit (H), number of quantity discount options (n), and the type of holding cost calculation. The second set of input parameters were order quantity range (minimum units and maximum units) and unit price (P) for each discount option. As for output parameters requirement, the optimum discounted order quantity and the total cost (Tc) were the required numerical outcomes for the discount quantity model. Meanwhile, the visual output as expected to be in the form of a graph for order quantity – total cost, which would be showing the position of optimum order quantity as well as its total cost comparison to the order quantity – total cost curve of all quantity discount options. It was desirable to be able to show the position of simple EOQ point of each range compares to its valid range. Finally, for the ease-of-use requirement, it was desirable to use an editable table which can be generated according to the number of options while allowing small amendments of the parameters value since there are many parameters required and the flexible nature of the number of options.

2.2 Assumptions and limitations/scope

Opening and closing inventories, lead time, and safety stock were irrelevant for this exercise. The buyer was provided with several options of unit prices paired with the range of order quantity. The prices remain the same during the year. The setup cost (S) was fixed. The holding cost (H) will have two options; fixed or unit price based. The demand (D) is a known quantity for the year.

2.3 InventSimQD workflow

In the depicted process outlined in Figure 2, the operation started with inputting the initial set of parameters: D, S, H-related parameters, and the count of discount options n. Upon pressing button 1, an editable table for quantity discount emerges, accommodating the second set of input parameters: order quantity range and unit price (P). Subsequently, upon activation of button 2, the Quantity Discount (QD) algorithm executes to determine the optimal order quantity for the specific case. Following this computation, the plot and display algorithm takes over, generating the graph and presenting essential information within the designated space.

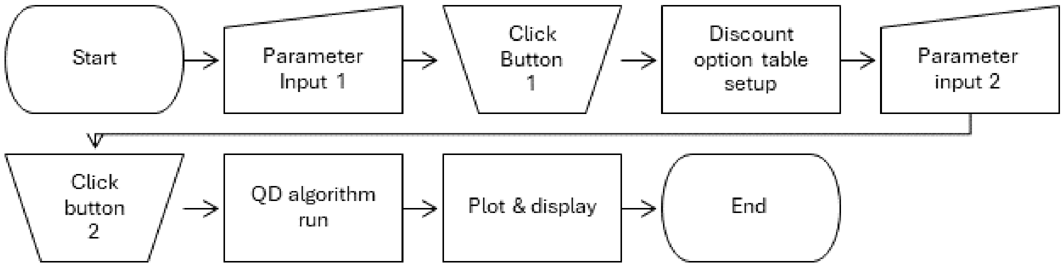

Figure 2. The Flowchart of InventSimQD

2.4 Quantity Discount algorithm

Figure 3 illustrates the QD algorithm utilized in InventSimQD. The initial step involves establishing the holding cost per unit (H) for each discount option, which can either be uniform across all options or vary based on a percentage of the unit cost, resulting in differing holding costs among options. Subsequently, the second step entails computing the basic Economic Order Quantity (EOQ), denoted as Q0, for each option using Equation (1). Following this, the third step involves identifying the optimal order quantity for each option, designated as Qn, as determined by the subsequent Equation (3).

$$QQ_n = \{QQ_0 ,\ QQ_{min} < QQ_0 < QQ_{max} \quad QQ_{min} ,\ QQ_0 < QQ_{min} \quad QQ_{max} ,\ QQ_{max} < QQ_0 \qquad (3)$$

Qmin and Qmax were the minimum and maximum order quantity for a given price option, accordingly. In the fourth step, we calculate the total cost (Tc) of each option using the quantities (Qn) and holding costs (H) obtained earlier. Then, in the fifth step, we identify the minimum Tc (TCmin) and determine the optimal discounted order quantity (QQD) corresponding to Qn the TCmin.

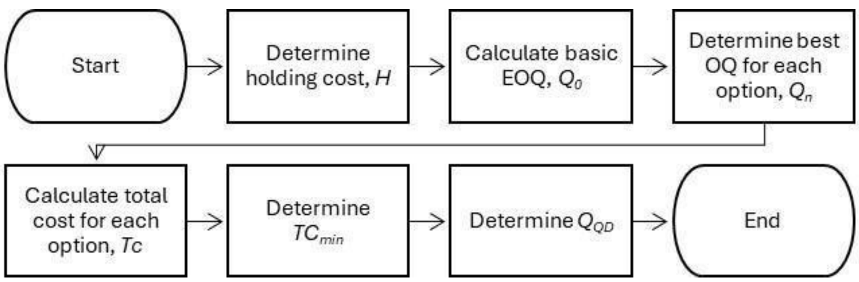

Figure 3. The Flowchart of Quantity Discount Algorithm

2.4 Plot and display

In InventSimQD, the plotting implementation utilizes iterative plotting facilitated by the "hold on" function, allowing continuous updating of axes for adding multiple curves. Figure 4 demonstrated the plotting and display algorithm. The process began with resetting the axes and initiating the "hold on" function. Next, the total cost (Tc) curves were plotted for each option within the valid quantity range, distinguished by colored and thick lines, with corresponding legends recorded. Subsequently, the x-line and y-line were plotted to indicate the optimal order quantity (QQD) and the corresponding minimum total cost (TCmin), updating legends accordingly. Upon completing legend updates, the legend function ceases, followed by plotting the total cost curve for each option across the entire quantity range with less significant line thickness and color.

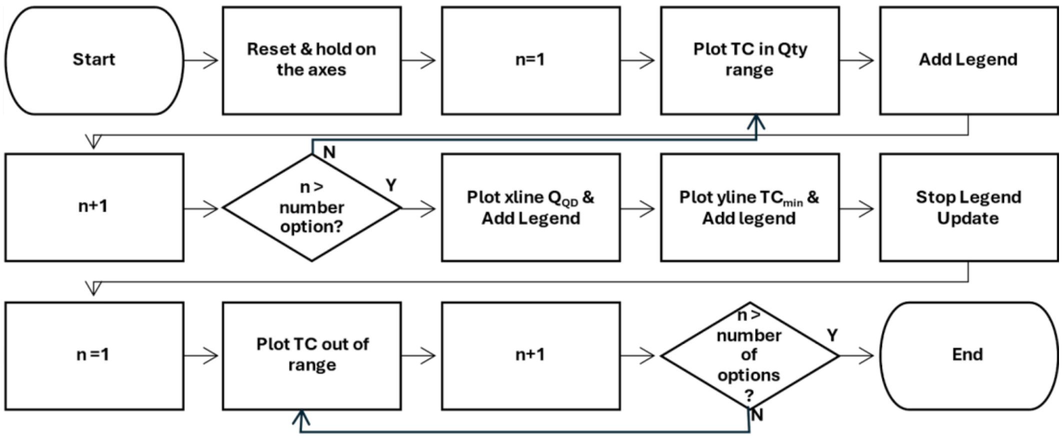

Figure 4. The Flowchart of Plot and Display Algorithm

3.0 RESULT

The feasibility and practicality of InventSimQD was verified by comparing the capability of the software to represent the calculations based on certain inventory parameters given to the student. Table 1 was the example of test question given to the students for the quantity discounts model theoretical calculations. Students were asked to determine the best EOQ value resulting from the lowest inventory cost of different discounts options within 40 minutes during the test.

Table 1. Setting Parameters for EOQ Verification

Quantity	Price per unit (RM)	D (annual demand)	S (RM)	H =IP(RM)	EOQ	Cost
1-1999	5.00			0.35* 5 = 1.75	$\sqrt{\frac{2(20000*12)(20)}{(1.75)}} = 2342$	**Not Feasible**
2000 - 3999	4.80	240,000	20	0.35 * 4.8 = 1.68	$\sqrt{\frac{2(20000*12)(20)}{(1.68)}} = 2391$	**RM1,156,015.97**
4000 or more	4.50			0.35 * 4.5 = 1.58	$\sqrt{\frac{2(20000*12)(20)}{(1.575)}} = 2469$ = adjusted to 4000 bottles	**RM1,084,350.00**

The same input parameters which were D, S, H, and discount option were inserted in the Yearly Inventory Conditions box of the GUI dashboard. The output of the simulation is as shown in Figure 5.

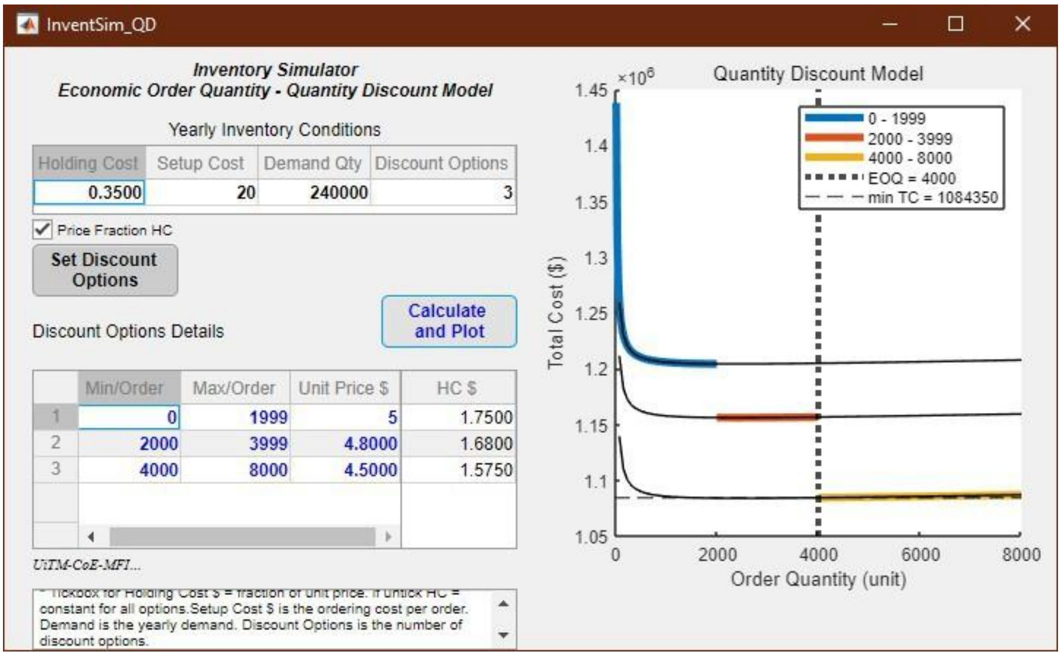

Figure 5. The InventSimQD EOQ Graphical Result

The graphical result obtained from InventSimQD showed the same optimal EOQ of 4000 as compared to manual calculation hence the accuracy of the algorithm can be confirmed. On top of that, it can clearly indicate the optimum EOQ directly from the graph (dotted line). Without the graphical illustration, the students have tendency to directly use the calculated EOQ value (2469) and multiply it with the cost (RM 4.50) without realizing that this cost was only applicable if the quantity order 4000 and above which lead to wrong total cost. In other words, the student would notice if they wrongly calculated the answer, and this helps the students to have better understanding of EOQ when the student used this apps.

However, the curves appear horizontal to the right causing minimum total cost or the basic EOQ of each curve are not clearly visible in the graph. This is due to the low holding cost set in the question ($0.35) as compared to the ordering cost ($20). The holding cost is smaller as it is imposed for each of the quantities, while ordering cost is imposed each time certain quantities are ordered, not per item.

This type of question may not be suitable to be used during the exercise in the classroom as the poor shape of the graph may make it difficult for students to understand concepts. For the simulated graph of InventSimQD to show similar pattern as the theoretical of EOQ curve display in Figure 1, the holding cost values need to be balanced with ordering cost value. For instance, when the holding cost is set to 10, the new graph would look much similar as shown in Figure 6. This will avoid unnecessary confusion for the students related to this topic.

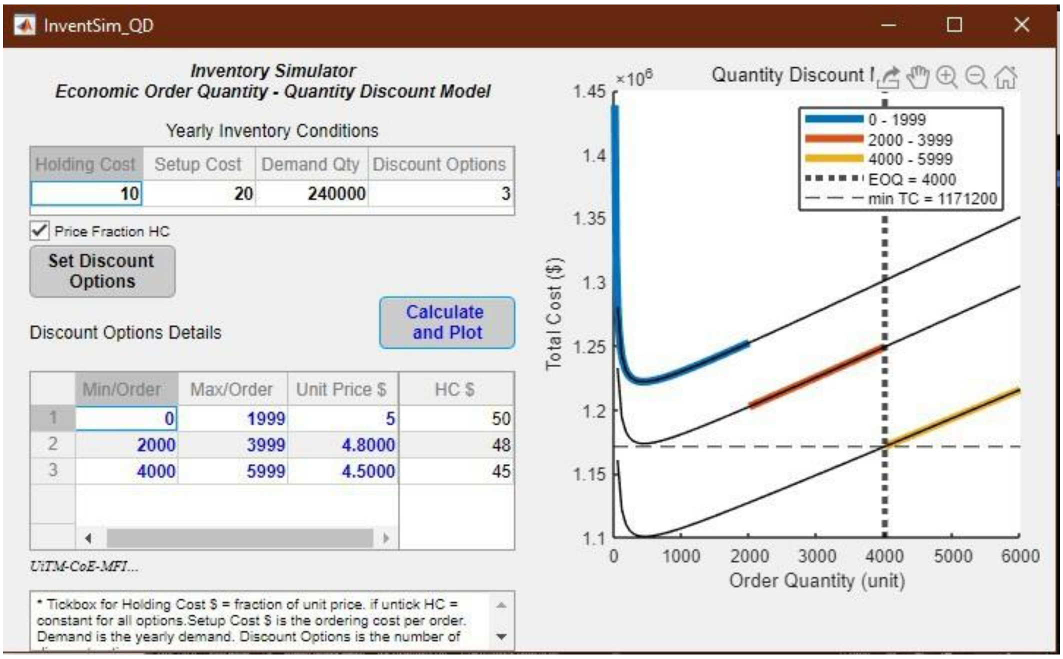

Figure 6. Expected curve presentation as result of holding cost changes.

4.0 CONCLUSION

The GUI based InventsimQD apps was proven successfully developed by using MATLAB. The validity and rapid visualization of InventsimQD output can offer visual understanding for the students as well as helping the lecturer to deliver the subject. Nonetheless, the discount costs display was not exactly like the theoretical total cost-curve graph. This was due to the value of holding cost per unit is so small as compared to the ordering cost. For future works, graph display must be improved to show more visible EOQ points at different quantity discounts options. There must be balanced values of both ordering and holding cost that could change the curve as per cost curve theoretical graph.

ACKNOWLEDGMENTS

This research was not funded by any grant. The authors would like to thank School of Mechanical Engineering, College of Engineering, Universiti Teknologi MARA for encouraging this research.

REFERENCES

1. J. Heizer, B. Render, , & C.Munson, ”Principles of operations management : sustainability and supply chain management (10th ed.).” *Pearson Education Limited*. 2017.
2. M.D. Khedekar , S.J Aher, M.B. Mandale , D.D. Bobalade , V.M. Patil , & S.N. Yadav, “Enhancing the Teaching-Learning Experience with the Implementation of MATLAB Tool: A Case Study.” *Educational Administration: Theory and Practice*, *30*(5), 10891–10903. 2024.
3. J.C. Domínguez, D.Lorenzo, , Julián García, C. Hopson, V. Rigual, , M. Virginia Alonso, & M. Oliet, “MATLAB applications for teaching Applied Thermodynamics: Thermodynamic cycles.” *Computer Applications in Engineering Education*, *31*(4), 900–915. https://doi.org/10.1002/cae.2261, 2023.
4. J. A.Rossiter, “MATLAB apps to support the learning and understanding of simple system dynamics.” *IFAC-PapersOnLine, 55(17), 121–125. https://doi.org/10.1016/j.ifacol.2022.09.267,*2022.
5. L.Gopabala Krishnan, A. Albani, Y. Segar, A.R. Ridzuan , M.K.Ismail , M.A.Musa , K. Gunaseelan, T.S. Subramaniam, & M.Z.Ibrahim, “The student response on the use of renewable energy graphical interface simulator in learning environment.” *Front. Educ.* 8:1023499. doi: 10.3389/feduc.2023.1023499. 2023.
6. M. Md Jan, N. Zainal, & S.H. Yusof, “Development of Graphical User Interface (GUI) for Surgery Assistance Application.” *Jurnal Kejuruteraan*, *35*(6), 1501–1512. https://doi.org/10.17576/jkukm-2023-35(6)-23, 2023.
7. T.-H. Chang, T. Yeh, , & R. C. Miller, “GUI testing using computer vision.” *Proceedings of the 28th International Conference on Human Factors in Computing Systems - CHI ’10*. https://doi.org/10.1145/1753326.1753555, 2010.
8. C. S. Lent, ”Learning to Program with MATLAB.” *John Wiley & Sons.* 2022.
9. B. C. Kim, , & C. H. Kim, “A Study on the Development of GUI Software using MATLAB.” *Proceedings of the KIEE Conference*, 449–451. https://koreascience.kr/article/CFKO200003977689491.page, 2000.
10. S. Koc, & Z. Aydoğmus, “A matlab/gui based fault simulation tool for power system education.” *Mathematical and Computational Applications*, *14*(3), 207–217. 2009.
11. S. Y. Wong, & S. Y. Lim, “Electromagnetic education: Development of an interactive GUI for demonstrating wave polarization.” *Computer Applications in Engineering Education*, *28*(5), 1190–1219. https://doi.org/10.1002/cae.22296, 2020.
12. S. Alves, A. Callado, & P. Jucá, ”Evaluation of Graphical User Interfaces Guidelines for Virtual Reality Games.” *IEEE Xplore*. https://doi.org/10.1109/SBGames51465.2020.00020, 2020.
13. A. H. M. Nordin, R. F. Mustapa, M. E. Mahadan, N. H. Ahmad, , & N. Y. Dahlan, . ”Transformer interactive learning tool based on MATLAB Simulink and GUI.” *IEEE Xplore.* https://doi.org/10.1109/ICEED.2017.8251162, 2017, November 1.
14. P. R. Medwell, L. A. Brooks, & B. S.Medwell, “Analysis of the Lawn Bowl Trajectory as a teaching tool for Sports Engineering: development of a graphical user-interface.” *Procedia Engineering*, *13*, 531–537. 2011.

[1] P. Karel, & Z.Tomas, “Multimedia Teaching Aid for Students of Basics of Control Theory in MATLAB and SIMULINK.” *Procedia Engineering*, 100, 1*ring and Technology*, *3*(June), 179–197.