Journal of Materials Engineering, Structures and Computation 3(4) 2024 pp. 01-18 ISSN-1115-5825

NIPES

Journal of Materials Engineering, Structures and Computation

Journal homepage: www.nipesjournals.org.ng

Development of a Software for The Analysis of Simple Vapour Power Cycles with Open Feed Water Heaters

Egware H.O[*a] and Unuareokpa O.J[b]

[ab]Department of Mechanical Engineering, University of Benin, Benin City
Corresponding Author: okechukwu.egware@uniben.edu, omozee.unuareokpa@uniben.edu

ARTICLE INFORMATION

Article history:
Received 19 November 2024
Revised 21 November 2024
Accepted 23 November 2024
Available online November 2024

Keywords:
Development, Standalone Software, MATLAB, Rankine Power Plant, Efficiency

https://doi.org/10.5281/zenodo.14516371

ABSTRACT

Energy goes a long way in affecting the lives of individuals, companies and nations at large. The discovery of electrical energy is the catalyst and driving force for the transformation from the crude age to the jet age. Hence, it is rightful to say, that the steam power plant has seen positive improvements, since the introduction of technology. This improvement covers increased accuracy, elimination of time wastage, and room for smarter research. This study involved the development of a unique software that was used for the analysis of a steam power plant. MATLAB a powerful programming language for Engineers was used to create a standalone application, which can be installed on computers and Android device. Matlab's guide was used enormously as well as GUI, but other functions were created Also, the Steady State Energy Flow Equation was used in writing the Matlab codes for the developed standalone Rankine Cycle analysis software. The developed Standalone software was tested by comparing the results obtained from the software and the results from manual calculations The percentage error difference between the developed standalone and manual computation results of 0.0328% and 0.0395% was obtained for thermal cycle efficiency and net workdone respectively for a simple Rankine Cycle problem. Thus, this revealed a high level of accuracy of the standalone software. This software can therefore be recommended for classroom demonstration in solving Simple Vapour Power cycles with only two configurations (i.e. with open feed water heaters and without feed water heaters).

1. **Introduction**

Power generation, its demand and supply are some of the keys, if not the most important indicators of economic and social prosperity and bliss for a nation and a country at large [1-3]. It is unavoidably crucial and a necessity for human, industrial, military and a nation's development and growth in the long run. It is the aim of any developed or developing country like Nigeria in its power sector to transmit power to all parts of its entity for the efficient distribution and satisfaction of consumer demands. Therefore, power generation is an indispensable component to drive the realization of these aims.

For the power sector to be effective, it has to do what it was built for. To boost the effectiveness of the power sector, the steam power cycle was developed as a means to an end. The Steam power generation although effective, still needs further work and advancement in the aspect of increased efficiency output [4,5]. The 21st century has been termed the jet age for a reason, against earlier centuries technology has evolved and changed the ways by which data and invention spread. Recent survey shows an increase in the rate of transformation of Technology by an exponential amount since the beginning of the 21st century and what has played a major part in this increase in transformation is the advancement of technology through computers [6.7]. With the advent of the computer, the computer became a 'set' having software as its 'element'.

To design properly the equipment of any configuration of the Rankine cycle, four parameters should be analyzed, that are: Heat supplied in the boiler; Work generated in the turbine(s); Required work in the pump(s); and Thermal efficiency of the cycle. These parameters can be determined in a thermodynamic simulation of the process.

Nowadays, many computer programs are available for this purpose, working with different ranges and provides the user with simple configurations of the Rankine cycle. Previous work in this light has been done titled "Thermodynamic Simulation of Steam Power Cycles using GUI-Matlab Interfaces" by Arce and Nian [8]. It was aimed at Simulating different configurations of Rankine cycle, which provided good results, showing that it can be used to improve the functions of the equipment in these thermodynamic cycles and to support teaching disciplines of Applied Thermodynamics. Earlier research work also showed that Magnus [9] developed codes to determine the ideal regenerative Rankine cycle with one open feed heater. Similarly, Arnav Raj Joshi et al. [10] modelled a steam power plant using Python software to solve its thermodynamic problems and Ibikunle et al. [11] developed software for a waste-to-energy steam power plant using Java script. These previous studies have revealed the importance of computer simulation applications in steam power plants.

The previous studies reviewed looked at the various software simulation applications. One of the problems identified was that the particular software is needed to carry out such simulation operations. To address this problem a stand-alone software needs to be developed. The advantage of the stand-alone software is that it can be installed in any window and can be utilized without the primary use to develop it.

In this regard and to advance research in standalone software development a friendly standalone software was developed using MatLab computer program. The developed software aims to simulate several configurations of Rankine cycles, by utilizing the mass, energy and entropy balances in turbines, pumps, boilers and condensers in ideal and real situations.

2. Methodology

2.1 Simple Rankine cycle

A simple Rankine cycle power plant major components consist of feed pump, boiler, steam turbine, condenser and generator. The schematic layout and T – s diagrams are illustrated in Figures 1 and 2 respectively.

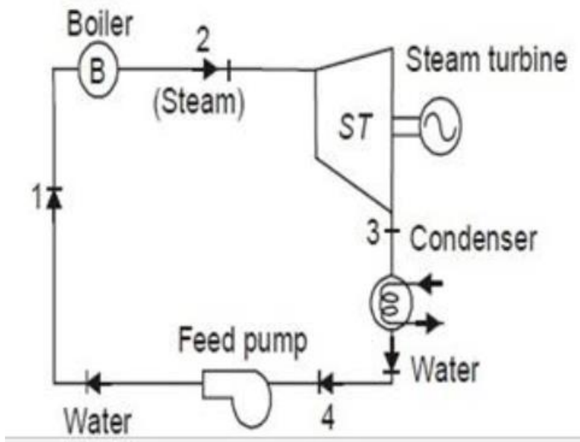

Figure 1. Schematic Diagram of a Simple Rankine Power Cycle

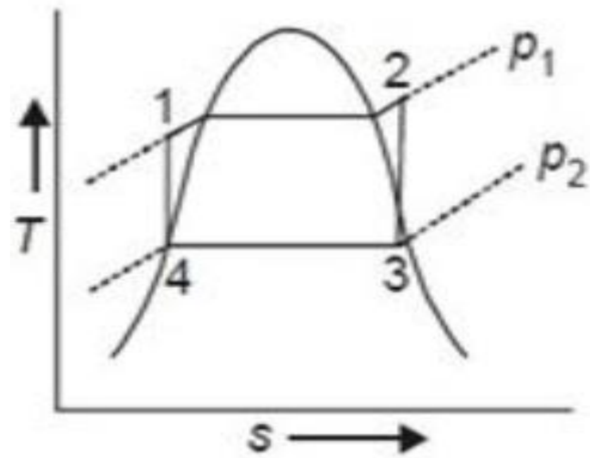

Figure 2. T – s Diagram of a Simple Rankine Power Cycle

From Figures 1 and 2, the different states are explained as follows:

1 – 2: High pressure water supplied by feed pump is heated and transformed into steam with or without superheat as per requirement. This high pressure and temperature steam flow into in the steam turbine. Heat added in boiler, for unit mass of steam (Q_{in}) was determined as stated in Equation (1) [12,13].

$$Q_{in} = (h_2 - h_1) \quad (1)$$

2 – 3: Steam available from boiler flow into the steam turbine, where it's adiabatic expansion takes place and positive work is developed. Expanded steam is generally found to lie in wet region. Expansion of steam is carried out to the extent of wet steam having dryness fraction above 85% so as to avoid condensation of steam on turbine blades and subsequently the droplet formation which may hit hard on blade with large force [14]. Turbine work, for unit mass of steam (W_{out}) was determined using Equation (2) [13].

$$W_{out} = (h_2 - h_3) \quad (2)$$

3 – 4: Heat rejection process occurred in condenser at constant pressure causing expanded steam to get condensed into saturated liquid at state 4. Heat rejected in condenser for unit mass (Q_{out}) applying Equation (3) [15].

$$Q_{out} = (h_3 - h_4) \tag{3}$$

4 – 1: Condensate available as saturated liquid at state 4 is sent to feed pump for being pumped back to boiler at state 1. Pump work for unit mass (W_{in}) can be evaluated using Equation (4) [16].

$$W_{in} = (h_1 - h_4) = v_4(p_1 - p_3) \tag{4}$$

The net work done (W_{net}) generated in a simple Rankine cycle can be mathematically obtained as shown in Equation (5) [16]..

$$W_{net} = W_{out} - W_{in} \tag{5}$$

The thermal efficiency (η) and work ratio (WR) of a simple Rankine cycle power plant can be obtained as shown in Equations (6) and (7) respectively as in [18,19].
The thermal efficiency is ratio of the net work done to the heat supplied by the boiler.

$$\eta = \frac{W_{net}}{Q_{in}} \tag{6}$$

The work ratio is the ratio of the net work done to the work generated by the turbine.

$$WR = \frac{W_{net}}{W_{out}} \tag{7}$$

The specific steam consumption by the steam power plant can be determined applying Equation (8).

$$SSC = \frac{3600}{W_{net}} \tag{8}$$

2.2 Rankine Cycle Model with Open Feed Water Heater

Open feedwater (Direct contact) heating means that working medium emerging from the condenser is preheated by mixing with the steam bled from the hot part of the turbine. That is, the extracted or bled steam is allowed to mix with feedwater, and both leave the heater at a common temperature. Open feedwater heaters are simple and inexpensive and have good heat transfer characteristics [20]. They also bring the feedwater to the saturation state. For each heater, however, a pump is required to handle the feedwater. This regenerative Rankine Cycle is the modification of the simple Rankine cycle by adding an open feedwater heater and additional pump. The schematic and T – s diagrams of the open feed heater Rankine Power Plant are shown in Figures 3 and 4 respectively.

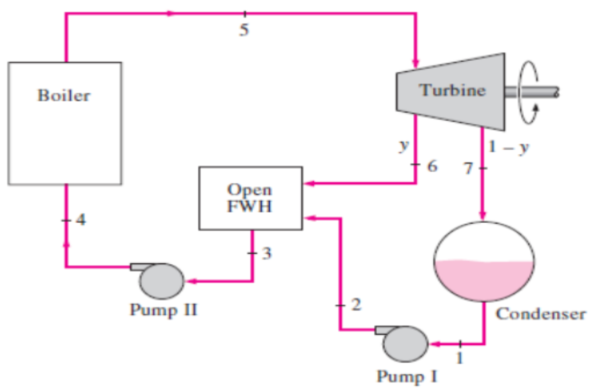

Figure 3. Schematic diagram of Open Feedwater Heater Power Plant

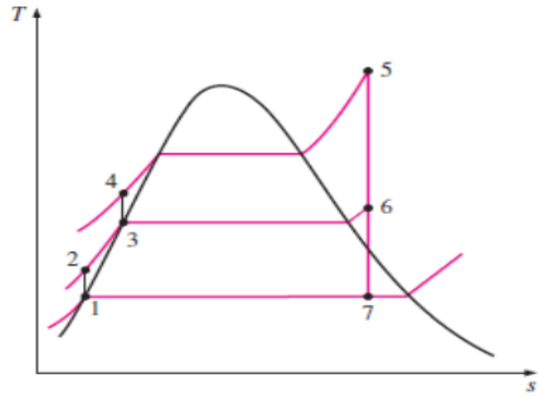

Figure 4. Open Feedwater Heater Power Plant T – s diagram

Figure 4. Open Feedwater Heater Power Plant T – s diagram

As shown in Figures 3 and 4, the following parameters are evaluated applying Equations (9) to (17) as follows [13]:

Heat Supplied (Q_{45}) in the boiler was determined using Equation (9).

$$Q_{45} = h_5 - h_4 \quad (9)$$

The work developed (W_{out}) by the steam turbine can be evaluated by applying Equation (9a)

$$W_{out} = h_5 - h_6 + (1-y)\times(h_6 - h_7) \quad (9a)$$

To determine the various pumps work ($W_{p1}$and W_{p2}) and total pump work (W_{in}), Equations (10) to (12) are used.

$$W_{p1} = v_{f1}\times(1-y)\times(p_2 - p_1) \quad (10)$$

$$W_{p2} = v3\times(p_4 - p_3) \quad (11)$$

$$W_{in} = W_{p1} + W_{p2} \quad (12)$$

The specific enthalpy of steam entry into the boiler at state 4 can be determined using Equation (13).

$$h_4 = W_{p2} + h_3 \quad (13)$$

The net work done, thermal efficiency, work ratio and specific steam consumption for the Open Feedwater heat Rankine Cycle power plant can be evaluated using Equations (14) to (17) respectively.

$$W_{net} = W_{out} - W_{in} \quad (14)$$

$$\eta_{th} = \frac{W_{net}}{Q_{45}} \tag{15}$$

$$WR = \frac{W_{net}}{W_{out}} \tag{16}$$

$$SSC = \frac{3600}{W_{net}} \tag{17}$$

Assumptions made were as follows as stated in [21]:
1) Heat losses from the gas turbine power plant components are negligible.
2) Kinetic and potential energy components are negligible.

2.3 Conceptual Design

Open The Graphical User Interface is a MATLAB computer program that enables an individual to communicate with the computer using objects [22]. These objects could be symbols, buttons, visual metaphors, or pointing objects. Put differently, The Graphical User Interface, or GUI, refers to the now universal idea of icons, buttons, etc., that are visually presented to a user as a "front-end" of a software application. It encompasses pull-down menus, slider controls, and check boxes. Conceptual design has three phases of interface design. The required basic input and output parameters needed for the software are presented in Table 1.

Table 1: Software input and output parameters.

S/N	INPUT	OUTPUT
1	Pressure	Work input, work output
2	Temperature	Power
3	Username, password	Efficiency

2.3.1 Design

In this design stage, the components, tasks and sequences are required to make the GUI effective are considered. The paper and pencil method were used to achieve this task, which is the most important stage. In the MATLAB software "'PENCIL" is needed to enhance the paper sketch. Matlab has a great resource for a multipage software and ability to make each page communicate with each other smoothly; although it is quite difficult and requires some tweak. The GUI information for the software development is illustrated in Table 2.

Table2: GUI Layout Information.

S/N	FIGURE NAME	SHORT DESCRIPTION
1	login	allows user access through registration or login
2	plant layout	collects data used to configure and constrain plant layout
3	plant component variables	collects states variable for each plant component
4	plant result	displays the calculated result for the plant data supplied by the user

2.3.2 Construction

MATLAB is a powerful mathematical tool for matrix calculations and almost any other mathematical function required. MATLAB also has the ability to form windows like applications with its programming language. In the design phase of a standalone application for thermodynamic computation, a Graphical User Interface is an end product, there are different methods that can be adopted using MATLAB as a model language. These different possible methods range from the use of GUIDE Editor, to writing of MATLAB functions program or the use of a recent app designer editor. To give a better exposure, the GUIDE system is adopted and the steps involved which range from LAYOUT design, to CODING OF THE M.FILES and IMPLEMENTATION, are as follows:

To initiate GUIDE , type ***guide*** in the MATLAB command window and press ***Enter*** key. This will open the GUIDE Quick Start window as shown in Figure 5, where there are two tabs (Create NEW GUI and Open Existing GUI). Under the Create New GUI tabs, four GUI templates are available. Selecting the Blank GUI (Default) template and pressing Ok at the bottom of the window opens the design window as shown in Figure 5.

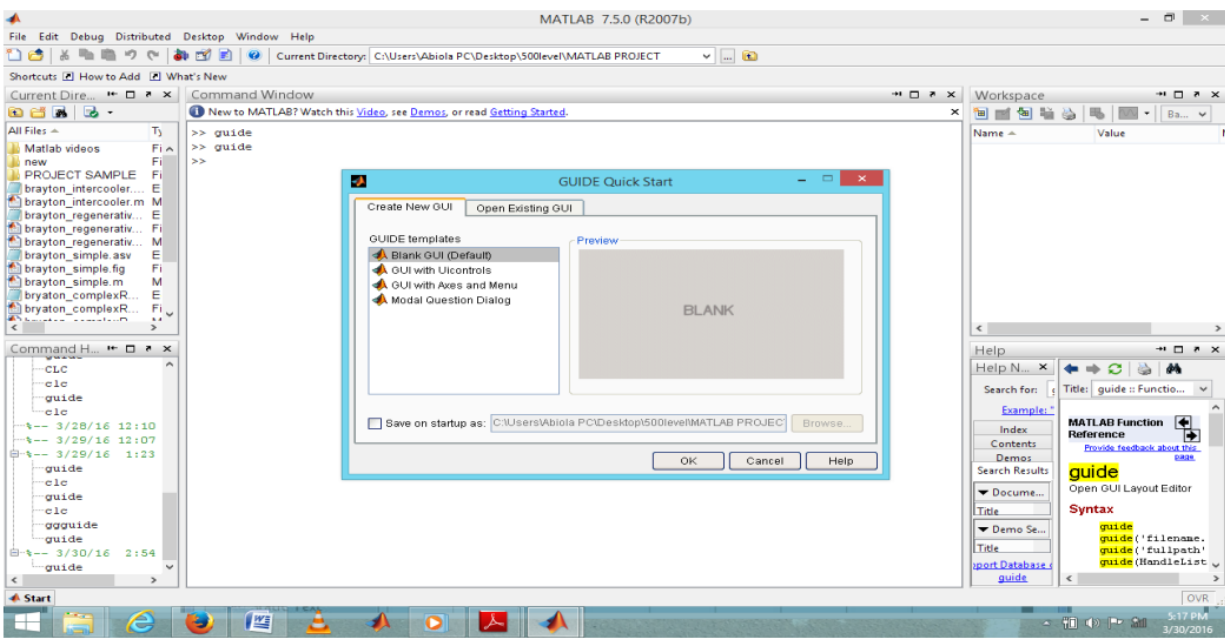

Figure 5. Untitled GUI

The GUI is not yet saved, so at the top of the window it is shown *untitled.fig*. Once the work is saved the title of the GUI will be reflected here. *.fig is the extension of GUI figure files. Generally, a GUI requires two files; **the figure (*.fig) files** where various components are aligned and **the code (*.m) files** where the coding is done. There is also provision to run the GUI using the single *.m file. At the top of the window Menu and Shortcuts can be found. To the right there are some GUI controls which are very important to learn for building GUI.

2.3.3 Login Figure

The login figure is the first component every user encounters. It serves to verify if the user is a registered owner of the software. To be a registered owner, the individual has to have a username, and a password registered in the database of the system as shown in Figure 6. If the user is registered owner (has his details in the database), the login figure grants the user access to the other part of the software, but if the user is not a registered owner, then the user has a choice to either close the application or register. The login interface is presented in Figure 6.

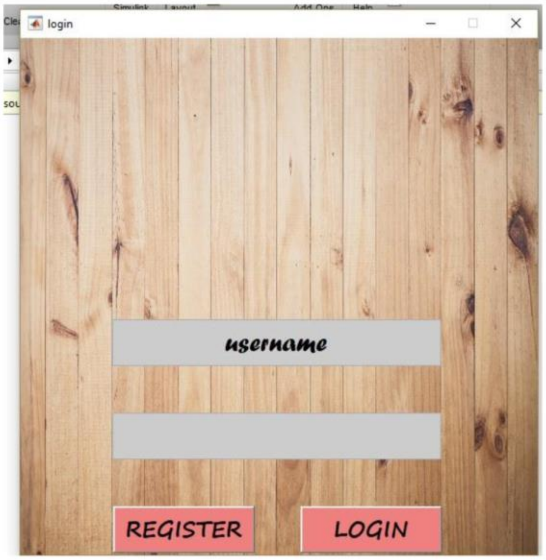

Figure 6. Login interface.

The login figure has two text input widgets; these widgets have characteristics of the HTML forms for obtaining data from the user. The first text input widget prompts the user to input his username, and the second text input widget requests the user to input his password, the password text is hidden so that external parties cannot view it.

The login figure also has two buttons, a register button and a login button. The login button collects the data the user filled in the text input for username and password and compares it with database values. If there is a match between inputted values and database values, then a message box appears telling the user that his login is successful. However, if there is a mismatch, a message box appears telling the user that his details are not correct. The register button gives the user a form to fill out. The form includes data values for first name, last name, email, username and password.

2.3.4 Login Figure Code.

The *login opening* function is the function called when the login function opens. The background picture is first set by calling the 'axes' inbuilt function, and setting any choice image using "imread" inbuilt function. The function also loads user data into the figure as it opens, this allows for future query of the data.

The *register callback* function is the function bound and connected to the register button, such that any action such as a click on the button is recognized. This function defines the form that is to be filled by the user; in terms of questions that are to be asked and the display size of the form. It also assigns variables to the answers the user inputted into the form. Last name stands for the users last name, email stands for the users email etc. is an if statement that helps the figure to check if there are missing values. It also checks if the user has entered the correct password by asking for the password twice and comparing the data using 'strcmp' inbuilt function.

The *login callback* function is bound to the login button. It intelligently uses and if statement to check if a username exists, and if it does exist, it goes further to check if the user's password is correct, it does all these by querying user filled data against the data already loaded from the database. Another important highlight of the login callback function is its ability to open the next figure and close the current figure. If it is empty then it should be created, but the current figure should be closed.

2.3.5 Plant Layout Figure Components.

Figure 7 is display after the login figure, this figure is called plant layout figure because it shapes the way the plant will be, the components it will have such as the number of turbines, if there will be a feed heater or not. Hence, all relevant calculations depend on this figure; it is the backbone of this study as is the setting configuration of an android phone. This figure defines for the plant, if it will have open feed heaters or not, and in future modifications if the feed heaters will be open or closed. The plant layout figure has two sections; the first section actually defines the plant configuration and the second sections is the labelled 'properties of steam'. This section allows the user to view properties of steam without having to consult a steam table.

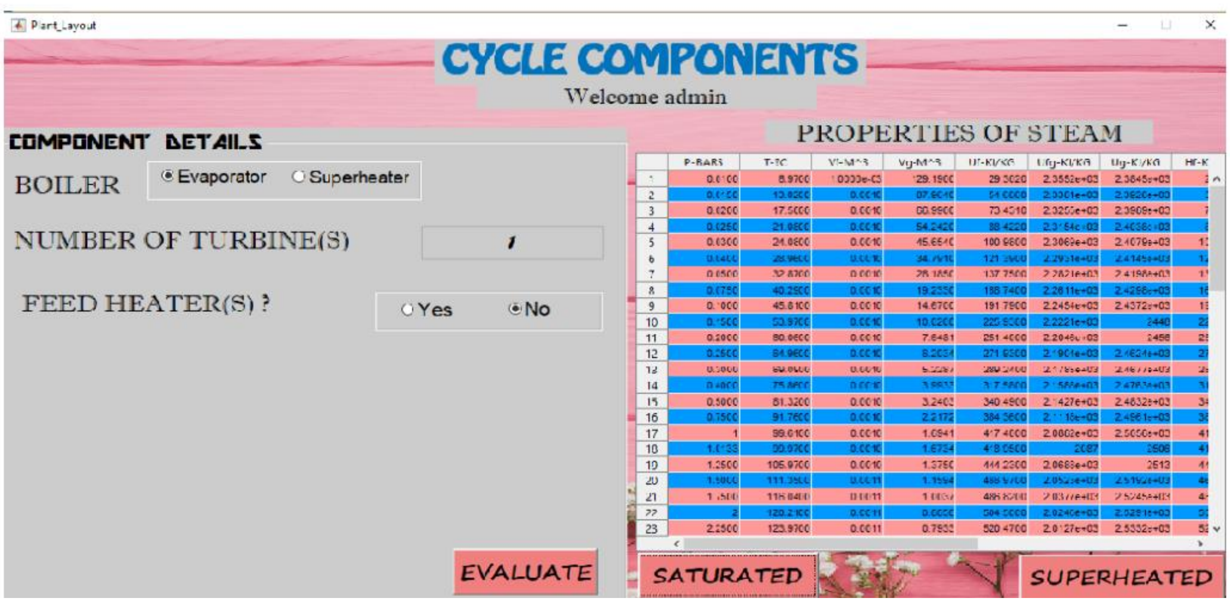

Figure 7. Cycle Components Interface

For the first section of the GUI, a panel was created. A panel is a component used as a container or placeholder for other components, which is simply its function. 'Component details' is the name of the panel, and it holds various static texts and text inputs that are used to get data from the user. The component details panel has a set of radio buttons, this radio buttons are used to determine if the plant is going to work with just an evaporator or it is going to work with a superheater. The component details panel has an 'edit' button next to a static text called 'boiler components', this edits button only appears if the user decide. to work with a superheated. The edit button provides a form for the user to fill out; the form gathers information as if the user wants to have a reheat heating system or just a simple boiler system. On the next line, there is a static text called "Number of Turbines" and a text input by the side, the text input is used to get data from the user as to how many turbines he requires. The next line in the panel holds the key to differentiating between two configurations, a simple plant or a feed heater plant. The line has a static text called 'Feed Heaters', and by the side, it has a radio button group which acts as a switch. Now if the user requires a feed heater, then a new panel becomes visible, this new panel called 'feed heater parameters' contains static text and text inputs used to collect and store values that will be required for calculations, such as the number of feed heaters and number of feed pumps. However, if the user requires no feed heater, then the new panel called 'feed heater parameters will become invisible

.

2.3.6 Plant Layout Figure Code.

For complexity sake and for sake of brevity, functions used in the FIGURE code were enumerated (the main element of a FIGURE is itself the functions used), than explaining lines of codes. The function Plant_Layout_OpeningFcn(hObject, eventdata, handles, varargin) is called when the Plant Layout figure opens. This function sets the background picture of the figure and it loads two excel files into the figure by saving them into variables, this allows the excel data to be passed around different sections of the code. This excel files contains data for the saturated properties of steam and for the superheated properties of steam, they only show up when the corresponding buttons to active them are clicked.
The functions *no_of_turbines_Callback*, *no_of_feedheaters_Callback*, *no_of_feedheater_pump_Callback* ; are used to get the values of number of turbines, number of feed heater pumps and number of feed heater pumps respectively, anytime the user inputs such values. This functions also save the values of this digits into variables, this way values are available to other sections of code.

The function *btgrp1_SelectionChangedFcn(hObject, eventdata, handles)* connects and binds actions to the yes/no radio buttons. This function controls visibility of the feed heater section for ease of use. If the user selects 'No', which means the user does not want to have feed heaters in his plant, the feed heater section remains hidden. But if the user decides to have feed heaters, he selects the 'Yes' radio button and the feed heater section appears, where the user can configure the feed heater parameters to their choice.

The function *btgp2_SelectionChangedFcn (hObject, eventdata, handles)* connects and binds actions to the Evaporator/Superheater radio buttons. They control the visibility of the edit button and the clicking property of the yes/no radio button. If the user decides to work with an evaporator, the edit button remains hidden and the yes/no radio button is active for clicking. But if the user selects super heater, the edit button appears, and the user can input how many reheats he wants, at the same time the yes/no radio buttons become inactive.

The *saturatedbutton_Callback and superheatedbutton_Callback* functions are similar in action. The saturatedbutton_Callback function is connected to the 'saturated' button, this function is responsible for displaying properties of saturated steam on the figure table. Likewise, the superheated button Callback function is used to display the properties of superheated steam on the figure table, and it is connected or bound to the superheated button.

The function *evaluatebutton_Callback(hObject, eventdata, handles)* is a utility function that performs a number of things. First, it checks through all the user input if the choices are in tune, also to see if errors exist. Combination choices such as reheating and feed heaters, uneven number of feed heaters and pumps, uneven number of reheat and number of turbines, are cases the software considers as errors. For instance, a user cannot have number of feed heaters as four and number of feed heater pumps as seven. The software flags an error as a notification. The number of feed heaters and pumps should tally to avoid errors. After concluding the necessary checks, the function opens up the Plant Layout Variables figure and feeds it the data it has gathered from the user, it automatically sets up the next figure so that the user is able to advance.

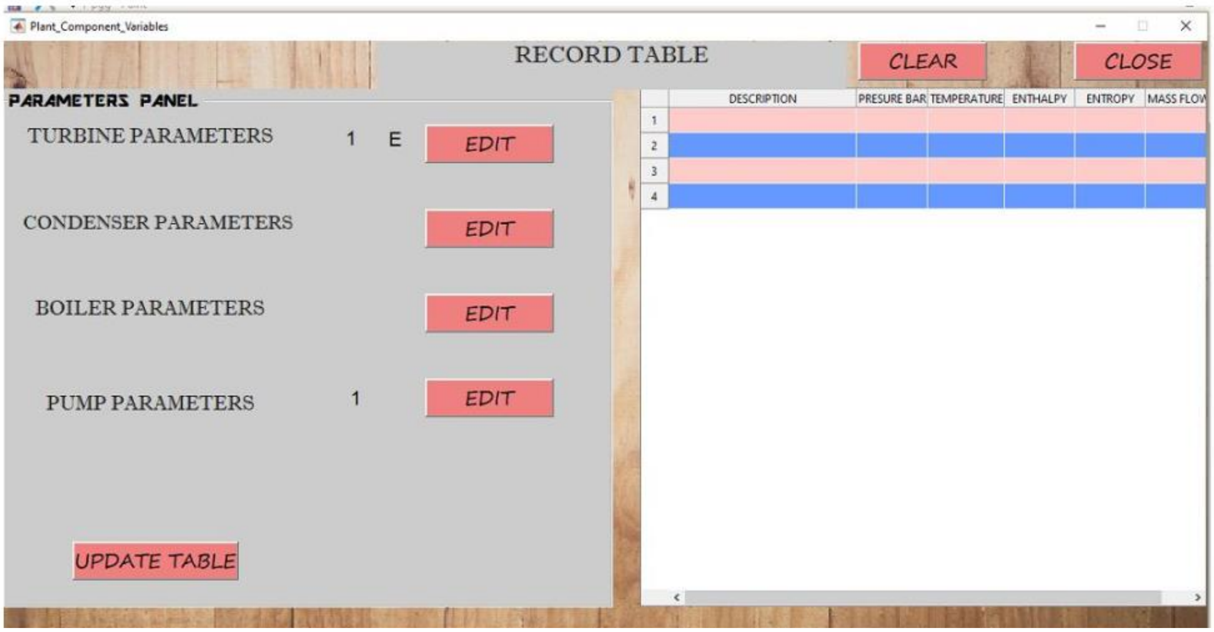

Figure 8. Record Table

The plant component variable figure is the calculation figure and has robust functions under it. The Figure 8 has two sections, the first section contains component variables and data input. Here the corresponding input and output temperatures and pressures are inputted for each component, depending on the data gotten from Figure 7. If in the Figure 7 the number of turbines was chosen to be 4, then the turbine component of the plant component variables figure will ask the user for input and output values of pressure and temperature for both high pressure and low-pressure turbine. But if the number of turbines chosen is 1, the only a high-pressure turbine will be available for customization.

The Figure has the ability to get the corresponding entropy and enthalpy values for each components state, it does this by calling multiple functions at the same time. After it does that, it updates itself and the table it has. The figure is responsible for making robust calculations for efficiency, work ratio, work done etc. the figure computes missing entropy values by applying looping and conditional statements, since these entropy and enthalpy values are needed for computation of other criteria. The second section of the figure holds a table for displaying the user input and the calculations that have been run for the various component enthalpy and entropy. Lastly the figure is responsible for opening the next and final figure, it supplies data to the figure and makes sure it opens in the best state.

2.3.7 Plant Variable Figure Components.

The plant component variable figure has a table element used for collecting and analyzing user data, the table element data can be in form of a cell array, structure array and an excel sheet. In this study cell arrays were chosen since the data can be manipulated more and have a wide range of functions that work on them. The table element is so flexible that it allows the Figure to update it using loops and conditional statement.
The Figure also has some buttons that are important to mention. The close button as the name implies, closes the figure and opens the login figure so that the user can have another session of usage. The clear button has the ability to clear the user data connected to the table; the clear button gets called when the user wants to input a different set of data or when mistake has been made. The update table button is the most important button in the figure, it holds the key to everything including calculations, data update and calling different functions for error checks. The update button is responsible for controlling the visibility of the result button. The result button is responsible and in control of opening the next figure, it does important calculations such as work input, work output, heat supplied, heat rejected calculations etc. the various edit buttons are responsible for generating user forms for each component, the user forms are customized to get user data for inlet and outlet component variables. The data gotten from these forms are saved into variables, so that they can be passed around functions. Other Figure components include static text explaining to the user which component he is working with.

2.3.8 PLANT COMPONENT VARIABLES FIGURE CODES.

The function *arranger(tdata,k,i,superheat,k2,satheat,handles,bt)'* is a function we created to work with the turbine component. It takes saturated values of steam, superheated values of steam, counter variable, number of rows of saturated and superheated steam values, user filled data gotten from the form and gives a cell array called dhata as output. This function gets data from the user for the turbine, specifically the pressure (and temperature in some cases), and checks if the user wants to work with a turbine that gets heat from an evaporator or a super heater. If the heat supplied is from an evaporator, the function intelligently uses saturated steam values to run its calculation of entropy and enthalpy based on a certain pressure inputted by the user, it also knows how to get the outlet values of entropy and enthalpy without much stress, since it has saturated values of steam at hand. After necessary calculations, it updates and saves this data in a cell array. If the heat supplied to the turbine is from a super heater, then the function knows that it is going to need both saturated and superheated values for calculation. It is also able to calculate dryness fraction as well as exit conditions of the turbine, and of course it needs both pressure and temperature values to do its calculation. Interpolation for corresponding pressure and temperature values are also available in this function. After calculating various entropy and enthalpy values, it updates the cell data and saves it to a variable for future use.

The function *h1=superr(j,superheat,tdata)* was created manually to work with the arranger function.it is called by the arranger function any time superheated calculations and interpolations are to be made. It accepts as input a counter variable; which keeps track of the position of data, superheated steam tables and the user data, it returns the enthalpy of steam at a particular pressure and superheat temperature as output.

The *dhata=satliq(gg,tdata,k2,satheat)* function is another function we created to help analyse and calculate values for the condenser. The condenser is a component where heat is rejected and the quality of steam is low(some degree of liquid). Since the condenser rejects heat, it therefore means that based on rankine constraint or modification, the outlet of the condenser will be liquid and the state of the fluid will be saturated liquid. Hence the function satliq, is used to calculate entropy and enthalpy values at saturated liquid conditions for the condenser. It also interpolates for pressures not included in the tables, and temperature values are not important here.

The function *newsat = saton(tdata,k2,satheat,a1)* was created as a means to analyse and calculate data for pumps. Pumps are plant components that increase the pressure on a fluid isentropically or adiabatically with a lower efficiency. Pumps used in the steam cycle work with saturated liquid and thus the function “saton” uses saturated values of steam to make necessary pump calculations, taking note of the volume at lower pressure and the pressure difference. The function takes in saturated values of steam and a counter variable as input and returns a data set as output. After necessary calculations, a variable saves the data and updating the figure table commences.

The *Plant_Component_Variables_OpeningFcn* function is responsible for loading all the components and elements needed by the figure before it opens. It also takes care of the background picture for the figure, including the fonts, colour and other styling. It also prepares the next figure, so that when updated data is ready, it can be open to accept it.
The function *turbpbutton_Callback* binds and connects actions to the turbine edit button. The function creates a user form; this user form collects data about the high pressure and lowpressure turbine. Data such as inlet pressure and temperature, outlet temperature and pressure are needed for enthalpy and entropy calculations. The data gotten from the user resides in a variable and is set for display by the figure table. This function also does error checking to see if the user actually inputted values.

The *condpbutton_Callback* function behaves just the same way as the previous function. It is attached or bound to the condenser button, such that whenever the condenser button is clicked the function is called. It is responsible for creating a form that is able to gather information about the condenser, information such as inlet pressure and temperature, outlet pressure and temperature are also gathered although not necessarily important. It also performs user checks to see if turbine data is non-empty; if turbine data is absent, an error arises. Upon data collection, it updates the figure table to display the latest results.

The function *boilpbutton_Callback(hObject, eventdata, handles)* is responsible for generating a form for collecting data related to the boiler or super heater, to this function inlet pressure and outlet pressure is required but not a necessity, the data given here is updated automatically irrespective of values inputted by the user. This function also does checks for empty or unfilled data and it updates the figure table after data acquisition from the user.

The function *pumpbutton_Callback(hObject, eventdata, handles)* is a more involving function than the previous one, in that it serves to gather information about the feed pumps as well as the boiler pump. It is able to differentiate if the plant has feed pumps or not. If the plant has no feed pumps the runtime of the function is shorter than when the plant runs on feed heaters and feed pumps. When the function is done collecting data, it updates the figure table
The function *heaterbutton_Callback(hObject, eventdata, handles)* collects data such as mass flowrate entering the heater and pressure rating of the heater. This function is unique in the sense that the user is aware of how many feed heaters are available and what info it needs. As usual, it is able to update the table after necessary calculations.

The function *closee_Callback(hObject, eventdata, handles)* and other similar functions are used for closing the figure, resetting the table data to have a blank appearance and over riding the cancel button at the top of the figure. These functions are useful in ensuring smooth exchange of data.

The function *resultbutton9_Callback (hObject, eventdata, handles)* is connected to the result button as the name suggests. It is responsible for gathering the enthalpy value for each state or component and making the corresponding calculation. These calculations include work input, work output, net work done, heat supplied etc. After necessary calculations, variables store the result of the calculations. This function also has as its responsibility, the opening of the result figure, after it opens this figure it supplies the data it has to it, so that it can be displayed by the figure.

The function *updatebutton_Callback(hObject, eventdata, handles)* is a robust function that makes use of 80% of all the other functions explained for this figure. It does update of user filled data by going through all the data the user has inputted, of course this is made possible by using a for loop (a while loop is also an alternative. In the process of going through the user it calls, other functions that it needs to help it make necessary calculations. For example, when it gets to the data the user has filled for the turbine, it calls the arranger function to help it gets the corresponding enthalpy and entropy values, it also knows how to treat high pressure and low-pressure turbine. The same reasoning goes for the condenser and the pump, but when it gets to the boiler, it just gets values from already updated data. Lastly, the function removes empty data elements and replaces it with zero.

2.4 Developed software Testing

The parameters from a working power plant data from Onkar [23] as shown in Table 3 were inputted into the software for validation purpose. Also, Equations (1) to (17) were used as manual computation process for the data in Table 3. For the validation of standalone developed software, the percentage error method between the standalone developed software and manual computation processes was adopted as stated in [21,24-26]. Equation (18) was used to compute the percentage error between the developed software output such as thermal efficiency and net work done generated and the others methods parameters as expressed in [27].

$$\%Error = \frac{Actual\_Output - Model\_Output}{Actual\_Output} \times 100\% \quad (18)$$

Table 3: Parameters Data Used for Testing the Developed Software

S/N	Parameters	Value
1	Boiler Pressure (bar)	50
2	Condenser Pressure (bar)	0.05
3	Turbine Isentropic Efficiency	1
4	Turbine Isentropic Efficiency	1

3. Results and Discussion

The values from Table 3 were inputted in the standalone developed software following the procedure described in Section 2.3 and the results obtained are display in Figures 9 and 10.

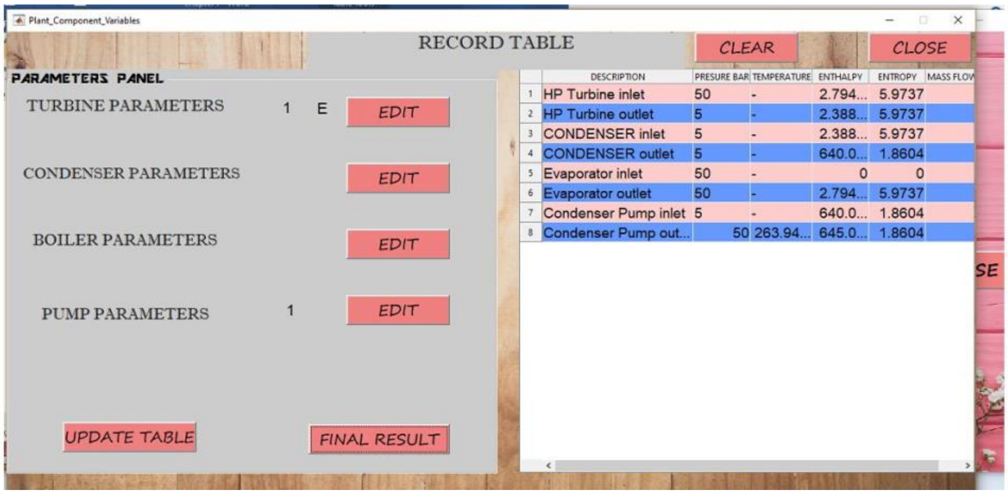

Figure 9. Recorded dada on the standalone software

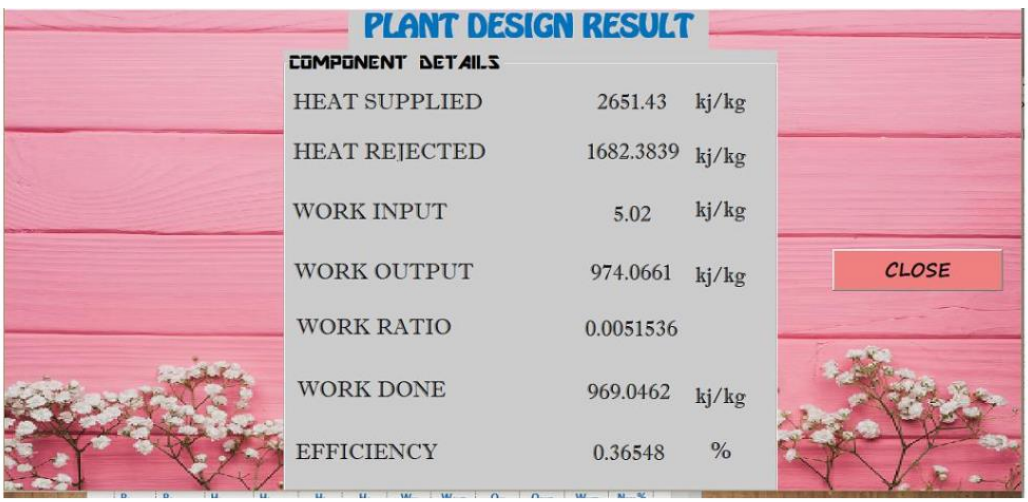

Figure 10. Plant Design Result

The results of the manual computation using Equations (1) to (17) as applicable are presented in Table 4.

Table 4: Results from Manual Computation

S/N	Parameters	Values
1	Boiler Pressure (bar)	50
2	Condenser Pressure (bar)	0.05
3	Specific Enthalpy at State 1, h_1 (kJ/kg)	142.84
4	Specific Enthalpy at State 2, h_2 (kJ/kg)	2794.30
5	Specific Enthalpy at State 3, h_3 (kJ/kg)	1819.80
6	Specific Enthalpy at State 4, h_4 (kJ/kg)	137.82
7	Specific Pump Work, Win (kJ/kg)	5.02
8	Specific Turbine Work, Wout (kJ/kg)	974.45
9	Specific Heat Supplied in Boiler, Qin (kJ/kg)	2651.46
10	Specific Heat Rejected in Condenser, Qin (kJ/kg)	1681.98
11	Net Workdone, Wnet (kJ/kg)	969.43
12	Thermal Cycle Efficiency (η)	0.3656

The test results comparing the Developed standalone software and manual calculation using Equation (18) are presented in Table 5.

Table 5: Percentage Error Results

S/N	Output Parameters	Standalone Software	Manual	% Error
1	Net workdone (kJ/kg)	969.0462	969.43	0.0395
2	Thermal Cycle Efficiency (η)	0.36548	0.3656	0.0328

Parameters from a steam power plant was analyzed using both manual calculation and the developed software. Important parameters suitable for design computation were adopted, neglecting losses and energy changes as assumed. As obtained from Table 5, both the manual and software solutions produced an efficiency of 36.56 % and 36.548 % respectively with a percentage difference as low as 0.0328% which is a high level of accuracy. Similarly, the net workdone obtained for the developed standalone software and manual computation are 969.0462kJ/kg and 969.43kJ/kg respectively with a percentage error of 0.0395%. The test results obtained for the developed standalone software showed they in the same region as the research works of Egware et al. [21]; Miguez Da Rocha [24], Wallentinen [25]; and Egware and Kwasi – Effah [26] that deployed similar test method in their previous study. This means that the software can be used and applied for power plants simulation operations since the values of errors obtained are small. This software with help to reduce the cumbersome involve in steam power plant output manual computation. Thereby, it will be recommended for use in institutions of learning, as its positive effect in engineering education will be enormously felt.

4. Conclusion

A standalone software using MATLAB was developed. The software accepts input and output data in numeric and text for various ideal steam power cycle cases. Data handled include condenser and turbine inlet temperatures, pressure, mass flow rate, heat supplied, network, power, efficiency, and outlet to the turbine. The developed software was used to compute for the following cases: simple system, system with reheat and regeneration, and open feed heater configuration. This software is also capable of displaying steam properties and some charts. The developed standalone software was tested with real-time data, and its results were satisfactory. This software could serve as one of the needed tools by Power plant designers and maintenance engineers, in making necessary calculations and decision-making for the design and maintenance of Steam power plants which could improve the generation of electrical power in Nigeria.

Nomenclature

h_n	Specific enthalpy (kJ/kg)
S_n	Specific enthropy (kJ/kgK)
P_n	Pressure (bar)
T_n	Temperature (^{0}C)
W_{in}	Work input (kJ/kg)
W_{out}	Work output(kJ/kg)
Q_{in}	Heat supplied(kJ/kg)
Q_{out}	Heat rejected(kJ/kg)
v	Specifc volume (m^3/kg)
WR	Work ratio
η	Thermal cycle efficiency
SSC	Specific steam consumption (kg/kWh)

References

[1] Anam, B. E., Ijim, U. A., Ironbar, V. E., Otu, A. P., Duke, O. O., & Achuk Eba, M. B. (2024). Economic recovery and growth plan, economic sustainability plan and national development plan (2021-2025): The Nigerian experience under President Muhammadu Buhari. Cogent Social Sciences, 10(1), 2289600.

[2] Somoye, O. A. (2023). Energy crisis and renewable energy potentials in Nigeria: A review. Renewable and Sustainable Energy Reviews, 188, 113794.

[3] Onochie, U. P., Egware, H. O., & Eyakwanor, T. O. (2015). The Nigeria electric power sector (opportunities and challenges). Journal of Multidisciplinary Engineering Science and Technology, 2(4), 494-502.

[4] Mittal, A., Brajpuriya, R., & Gupta, R. (2023). Solar steam generation using hybrid nanomaterials to address global environmental pollution and water shortage crisis. Materials Today Sustainability, 21, 100319.

[5] Siwal, S. S., Sheoran, K., Saini, A. K., Vo, D. V. N., Wang, Q., & Thakur, V. K. (2022). Advanced thermochemical conversion technologies used for energy generation: Advancement and prospects. Fuel, 321, 124107.

[6] Merali, Y., & McKelvey, B. (2006). Using Complexity Science to effect a paradigm shift in Information Systems for the 21st century. Journal of Information technology, 21(4), 211-215.

[7] Boehm, B. (2006, May). A view of 20th and 21st century software engineering. In Proceedings of the 28th international conference on Software engineering (pp. 12-29).

[8] Arce, P. F., & Vieira, N. F. (2017). Thermodynamic Simulation of Steam Power Cycles using GUI-MatLab Interfaces. Int. Journal of Engineering Research and Application, 7(1), 88-93.

[9] Magnus H. (2013). Modelling of Steam Power with open Feed water heater, A Report from the Department of Mechanical Engineering, IIT, Guwahati.

[10] Joshi, A. R., Deo, A., Parashar, A., Mishra, D. R., & Dumka, P. (2023). Modelling Steam Power Cycle using Python. Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol, 9(1), 152-162.

[11] Ibikunle, R. A., Titiladunayo, I. F., & Akinnuli, B. O. (2021). Development of a Software System for Selecting Steam Power Plant to Convert Municipal Solid Waste to Energy. Sustainability, 13(21), 11665.

[12] Damasceno, A., Carneiro, L., Andrade, N., Vasconcelos, S., Brito, R., & Brito, K. (2020). Simultaneous prediction of steam production and reduction efficiency in recovery boilers of pulping process. Journal of Cleaner Production, 275, 124103.

[13] Obanor, A., Dolor, G., Egware, H., & Aliu, S. (2015). Journal of Advanced & Applied Sciences (JAAS). Journal of Advanced & Applied Sciences (JAAS), 3(01), 24-31.

[14] Wang, T. (2017). The gas and steam turbines and combined cycle in IGCC systems. In Integrated gasification combined cycle (IGCC) technologies (pp. 497-640). Woodhead Publishing.

[15] Egware, H. O., & Ighodaro, O. O. (2023). Evaluating the effect of ambient air temperature on the exergy sustainability of a 153MW gas turbine power plant. International Journal of Thermofluids, 18, 100375.

[16] Singh, P. (2005). Optimization of internal hydraulics and of system design for pumps as turbines with field implementation and evaluation (Doctoral dissertation, Karlsruhe, Univ., Diss., 2005).
[17] Qu, J., Feng, Y., Zhu, Y., Zhou, S., & Zhang, W. (2021). Design and thermodynamic analysis of a combined system including steam Rankine cycle, organic Rankine cycle, and power turbine for marine low-speed diesel engine waste heat recovery. Energy Conversion and Management, 245, 114580.
[18] Egware, H., Obanor, A., & Itoje, H. (2014). Thermodynamic evaluation of a 42MW gas turbine power plant. International Journal of Engineering Research in Africa, 12, 83-94.
[19] Egware, H. O., & Obanor, A. I. (2013). Energy analysis of omotosho phase I gas thermal power plant. Inter. J. Engineer. Technol. Sci, 1(4), 206-207.
[20] Gonca, G. (2018). Thermo-ecological performance analysis of a double-reheat Rankine cycle steam turbine system (RCSTS) with open and close feed water heaters. International Journal of Exergy, 25(2), 117-131.
[21] Egware, H. O., Obanor, A. I., Aniekwu, A. N., Omoifo, O. I., & Ighodaro, O. O. (2021). Modelling and simulation of the SGT5–2000E gas turbine model for power generation. Journal of Energy Technology and Environment, 3(2).
[22] Thabit, A. A., Alzidi, E., Mohammed, M. A., & Palchyk, P. (2024, April). Development of a System Based on a Graphical User Interface Using MATLAB Tools. In 2024 35th Conference of Open Innovations Association (FRUCT) (pp. 744-754). IEEE.
[23] Onkar S. (2009), Applied Thermodynamics, New Age International (P) Limited Publishers, Third Edition, England
[24] Miguez Da Rocha A. (2010), Analysis on Solar Retrofit in Combined Cycle Power Plants, Master Thesis, - Institut fur Energietechnik und Thermodynamik, Technische Universit¨ at Wien Fakult¨ at fur Maschinenwesen und Betriebswissenschaften von , Madrid Spain.
[25] Wallentinen B.S (2016), Concentrated Solar Power Gas Turbine Hybrid with Thermal Storage, Master Thesis in Mechanical Engineering, Department of Energy and Process Engineering Norwegian University of Science and Technology.
[26] Egware, H. O., & Kwasi-Effah, C. C. (2023). A novel empirical model for predicting the carbon dioxide emission of a gas turbine power plant. Heliyon, 9(3).
[27] Zhang, W., Ma, L., Jia, B., Zhang, Z., Liu, Y., & Duan, L. (2023). Optimization of the circulating cooling water mass flow in indirect dry cooling system of thermal power unit using artificial neural network based on genetic algorithm. Applied Thermal Engineering, 223, 120040.