Materials Research Proceedings 29 (2023) 209-217 https://doi.org/10.21741/9781644902516-24

Development of web-based interface for dry-low emission gas turbine using raspberry Pi

MOCHAMMAD Faqih[1,a*], JEREMY Peter James[1,b], MADIAH Binti Omar[1,c], and ROSDIAZLI Bin Ibrahim[2,d]

[1] Chemical Engineering, Universiti Teknologi PETRONAS, Perak, Malaysia

[2] Electrical and Electronics Engineering, Universiti Teknologi PETRONAS, Perak, Malaysia

[a]mochammad_22000035@utp.edu.my, [b]jeremy_17005433@utp.edu.my, [c]madiah.omar@utp.edu.my, [d]rosdiazli@utp.edu.my

Keywords: Dry-Low Emission Gas Turbine, Rowen's Model, Scilab/XCos, Raspberry Pi, Python, HTML

Abstract. In recent years, Dry-Low Emission (DLE) mode has been popular in gas turbine applications which can reduce emission production while maintaining high performance by operating the combustion at a low firing temperature. However, this type of combustor is prone to experience trips due to lean blowout occurrences and combustion instability. Dynamic simulation is needed to investigate the system's behavior in mitigating the tripping problem. Therefore, this paper proposed a simulator of the DLE gas turbine using Raspberry Pi, which can be accessed through the local server. Rowen's model is modified to represent the DLE gas turbine system, and its physical model is developed using open-source software, namely Scilab/XCos. The model is then attached to a web-based interface developed using Python and HTML programming language. The deployment of the web-based interface enables the gas turbine model to be accessed remotely without caring about the device's location by connecting the mobile to the same Wi-Fi router. This simulator will help improve the technical know-how of DLE gas turbine operation and can be embedded in the plant control system as a predictive maintenance tool.

Introduction

A compressor, combustor, and turbine are the three main sections of a conventional gas turbine [1]. Gas turbine operates on the thermodynamic cycle referred to as Brayton cycle. Brayton cycle is a thermodynamic cycle in which compressed air is mixed with fuel and burned under constant pressure conditions, followed by mechanical power being generated by the hot gas that expands via a turbine to accomplish work [2]. However, the downside of a conventional gas turbine is that because it operates at a high temperature at the primary zone, it produces high emission of CO_x and NO_x. As a result of the high emission of CO_x and NO_x gases, environmental factors become a worry such as the increase of greenhouse gases in the environment which contributes to respiratory diseases, and a challenge is provided to reduce emissions without endangering the turbine's power efficiency. Because of that, Dry-Low Emission (DLE) gas turbines are introduced to solve this problem.

Even though the benefits of DLE gas turbine are apparent, DLE gas turbines also has its flaws. Typically, in DLE combustion, fuel is burned at double the amount of air with a low flame temperature [3,4]. While operating at a low flame temperature reduces NO_x emissions, it also raises the risk of incomplete combustion. Incomplete combustion results in higher contents of CO. As a result, the flame temperature in the combustor must be kept within the temperature range depicted in Fig. 1 to keep NO_x and CO_x gas emissions to a minimum.

Operating at low flame temperature also makes the DLE gas turbine prone to tripping. This is because at low flame temperature, the flame is very weak and unstable which then becomes very easy to flameouts hence causing trips to occur. Therefore, a dynamic simulation for DLE gas turbine model is required to analyse the system's behaviour to ensure stable operation of the DLE gas turbine.

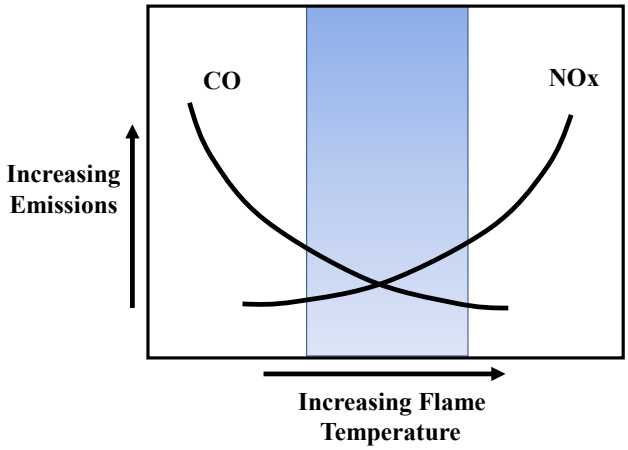

Fig.1 Behaviour of NOx and CO versus flame temperature.

In this paper, Rowen's model is selected to represent the behavior of the DLE gas turbine as shown in Fig.2. Preceding model was developed in MATLAB/Simulink which required high amount of license cost. This becomes a limitation for those who does not have license subscription. Therefore, in this paper, an alternative open source-based software namely Scilab/XCos is utilized to develop the Rowen's model. Scilab/XCos is selected due to the ability of performing simulation of physical model which is as powerful as MATLAB/Simulink with a free access and easy installation.

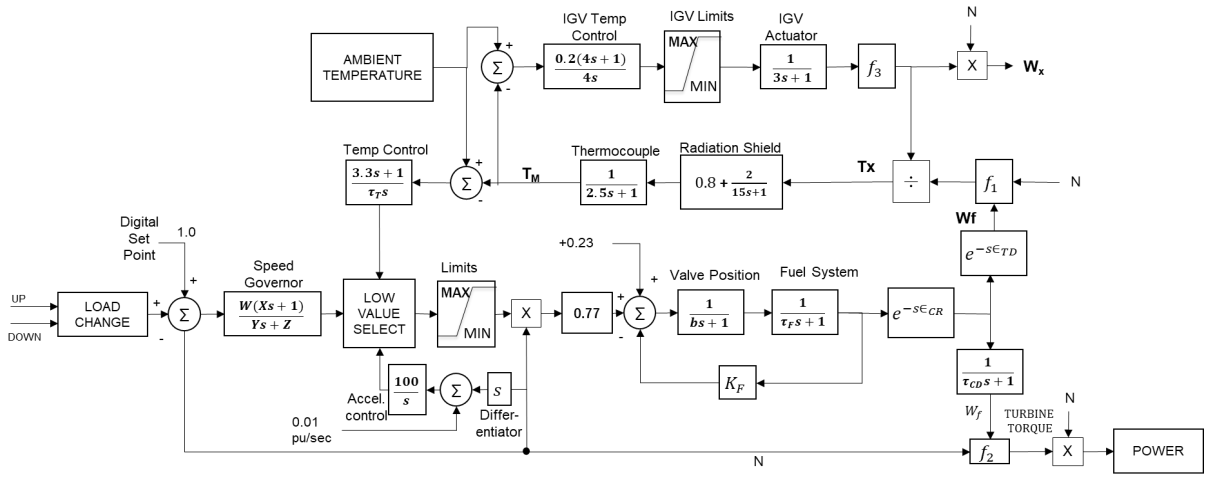

Fig.2 Rowen's model for dynamic studies [5].

Deployment of Dry-Low Emission Gas Turbine Model in Raspberry pi

Rowen's Model

Models that are derived directly from dynamic physical thermodynamic properties and principles are known as physical models. It involves utilizing laws governing thermodynamic behaviour,

such as conservation of mass/power/energy, in the Brayton cycle with several assumptions to obtain the differential equations to represent the dynamic gas turbine behaviour. As stated in [6], the ideal process of Brayton cycte is implemented into differential equations of total mass balance conservation as shown in Equation 1 and Equation 2:

$$\frac{dm}{dt} = \dot{m}_{in} - \dot{m}_{out} \tag{1}$$

$$\frac{dE}{dt} = \dot{m}_{in} i^*_{in} - \dot{m}_{out} i^*_{out} + Q + W \tag{2}$$

Where, $\dot{m}$ is mass flow rate, E is total energy, Q is heat input, and W is work Produced.

Rowen's Model is the first physical model of a gas turbine system, which depicts a heavy-duty gas turbine mathematically. The load-frequency and temperature controls are two of the most important controls illustrated in the diagram [7,8]. The assumptions that were made for this model are as followed: 1) Ambient Pressure is 1 atm; 2) Ambient Temperature is 15 ° C; 3) Relative Humidity is 60%; 4) Model developed is for a simple cycle and a single-shaft gas turbine.

With this developed model, calculation of output values, such as turbine temperature, can be calculated from the process inputs such as load. However, this method is very difficult to implement especially in a larger system due to numerous derivations that must be done to achieve the outputs [9,10].

Raspberry Pi

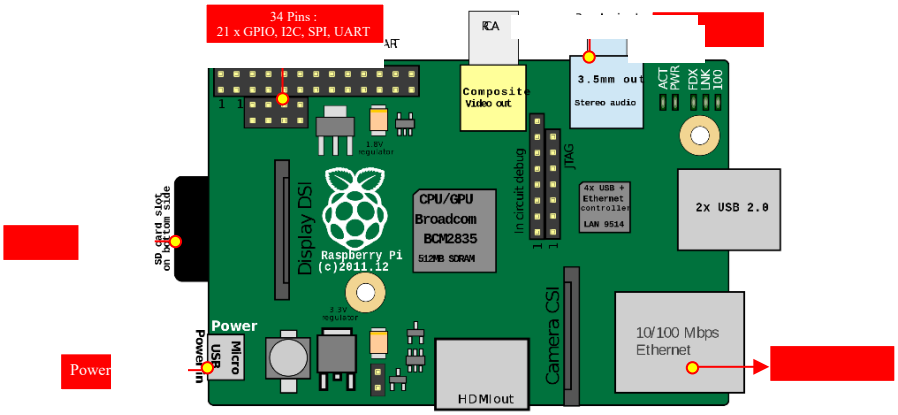

Fig.3 Layout of Raspberry Pi

Raspberry Pi and other single-board computers are low-cost development devices for testing and education [11]. Raspberry Pi operates on the Raspbian Operating System (OS) that is based on Debian (Linux) [12]. Operating on the Raspbian OS enables Raspberry Pi to have access to multiple packages and programs such as the Python programming language that is widely used in machine learning and deployment development. Layout of Raspberry Pi is displayed on Fig. 3. Assuming that the referred model hardware is the Raspberry Pi 4, this hardware includes a system-on-chip (SoC) Broadcom BCM2711, a Quad core Cortex-A72 CPU running at 1.5GHz, a graphical processor unit (GPU) VideoCore IV, and 4GB of RAM memory. The device also does not include a hard disk or solid-state disk by default, instead opting for the use of an SD card for the operating system and for nonvolatile storage. Power supply required by the device is a 5V 3A and the power port on the PCB of Raspberry Pi is a USB type-C connection. General Purpose Input Output (GPIO) ports on the Raspberry Pi function is to facilitate connecting a variety of peripheral devices to Raspberry Pi. USB/Lan ports are also available to connect devices onto the device and with both

the USB/Lan ports and GPIO pins, it is made possible to connect devices to the Raspberry Pi in a variety of ways that can be used for data collection or data dumping. Finally, the HDMI out port located at the device enable the use of a monitor on the Raspberry Pi and through connecting a keyboard and mouse on the USB ports in the device, Raspberry Pi can essentially function as a standalone device. Raspberry Pi can also be access without needing a screen, a mouse or a keyboard connected to it. This can be done through secure shell (SSH) from other devices such as laptops, personal computers, or mobile phones.

Methodology

The development method of the interface is depicted in Fig.4 including the Rowen's Model Integration and Configuration, Development of Web Hosting Service through Python, and Deployment on Raspberry Pi.

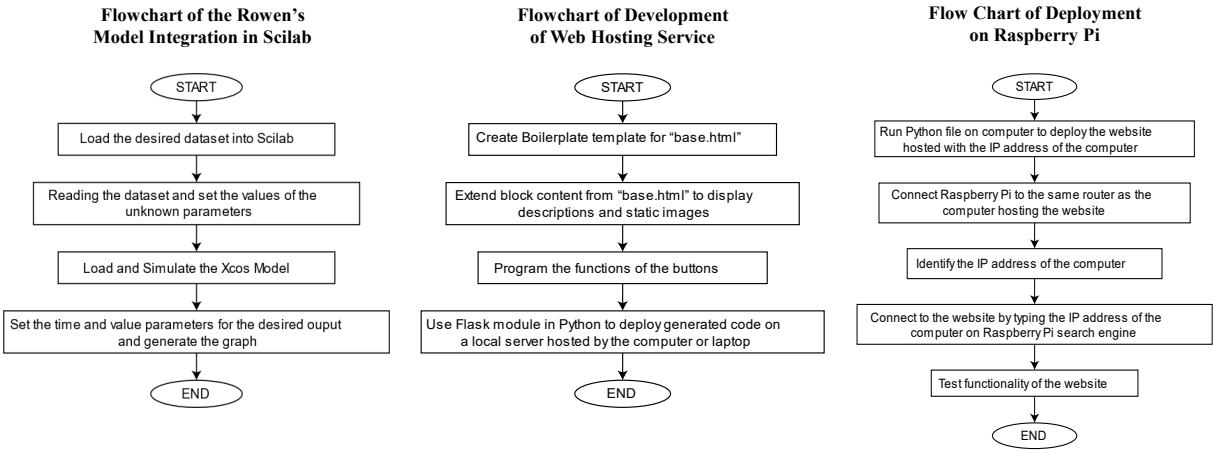

Fig. 4 Flowchart of web-based interface for DLE gas turbine.

The Rowen's model developed using Xcos in the previous section is integrated with Scilab to generate the graphs for the outputs of exhaust flow, temperature after turbine, turbine power, speed, valve position, and IGV.

The flow of development of web hosting service is starting by configuring boilerplate template. Boilerplate template is a template with the necessities needed to first start creating a HTML website. "Base.HTML" is the name of the file that the boilerplate template will be placed it and, in the template, "<%block content%>" are included to extend the "Base.HTML" into another HTML file where the necessary code for the other functions of the website is placed. The extended block content for the "Base.HTML" is written in a new file called "Home.HTML". In the new HTML file, it houses the code for the descriptions and static images placed on the website. Buttons functions for the exhaust flow, temperature after turbine, IGV, speed, turbine power, and valve position of the website is also located in "Home.HTML".To program the functions of the buttons located on the website, the programming language used will be Python and the modules used will be PyTesseract and PyAutoGUI. Finally, to deploy the created website, Flask module on Python is used.

By ensuring that the Python file on the computer is running and the website is hosted with the IP address of the computer to act as a local server, the Raspberry Pi can now be switched on to connect to the deployed website. For the Raspberry Pi to have access to the website hosted locally by the computer, the Raspberry Pi must be connected to the same Wi-Fi router as the hosting computer. After ensuring the connections to the same router is completed, the IP address of the

hosting computer can be identified, and the Raspberry Pi can connect to the hosted website by typing in the IP address of the computer on its own dedicated search engine which is "Chromium". If connection is successful, the Raspberry Pi will now access to the functions available on the website to generate the simulation of the gas turbine model and also retrieve the graphically data of the outputs of the gas turbine model which are exhaust flow, valve position, speed, IGV, temperature after turbine, and turbine power.

Result and discussion

From the generated Xcos model simulation, output parameters such as the exhaust flow, valve position, temperature after turbine, speed, IGV, and turbine power of the simulated gas turbine can be retrieved. To retrieve the output parameters of the simulated model, the gas turbine load dataset was inputted, and the variables used in the Xcos model simulation of the output values were identified as it is case-sensitive. Scilab was used to generate the Xcos simulation for a period of 10 seconds, hence, data simulated will be for the stated time period. With the variables identified, the average values and graphical representation of the output parameters was able to be retrieve using Scilab as described in Table 1 and the simulation result is shown in Fig.5.

Table 1. Parameters of simulation.

Parameters	Average Value
Exhaust Flow	0.845 kg/hr
Valve Position	0.858 %
Speed	1.0 %
IGV	0.382 %
Temperature After Turbine	857.71 °C
Turbine Power	0.869kW

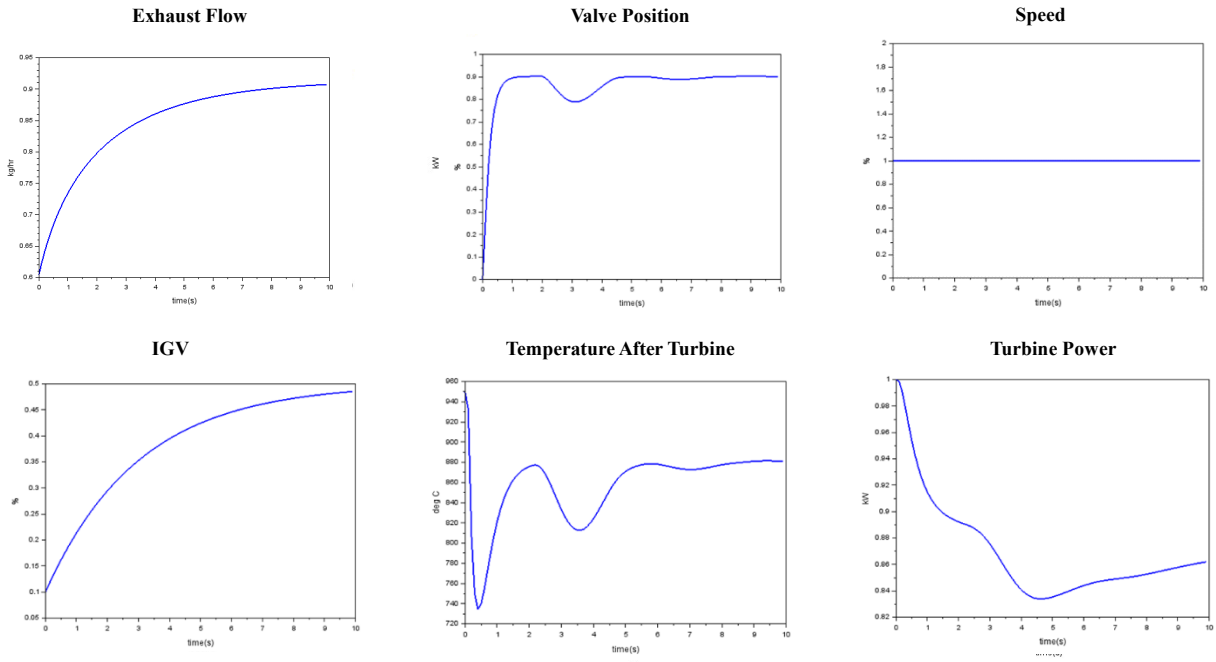

Fig. 5 Simulation result.

Materials Research Proceedings 29 (2023) 209-217 https://doi.org/10.21741/9781644902516-24

For the deployment of web-based interface, “Base.html” file was created using a boilerplate template that contains multiple scripts from “jQuery” and “bootstrap” which was used to indicate the appearance of the website generated. In the “base.html”, extending blocks are also available to further improve the website on a separate html file saved as “home.html”.

In “home.html” file, the general description of the developed gas turbine model and all push able buttons for the output parameters of the gas turbine model are included. Static images such as the UTP logo and gas turbine image is also included in “home.html” to further aid in the visual representation of the gas turbine model interface.

Two (2) main Python modules were used to give functions to the buttons available on the website which are PyTesseract and PyAutoGUI. As stated previously, PyTesseract is an optical character recognition (OCR) tool for Python that is used to recognize and read the text embedded in images while PyAutoGUI is a module that lets Python scripts control the mouse and keyboard to automate interactions with other applications. Extraction of the graphs and value generated from the previous subsection is done using the two (2) stated Python modules. As an example, to extract the graph and mean value of exhaust flow from the Xcos simulation, PyAutoGUI is used to automate the mouse and keyboard of the local server (computer used to host the website) to input the dataset submitted by the client or user through the “Choose File” s on Scilab to start the Xcos simulation of the gas turbine model and screenshot the graph and mean values of the exhaust flow simulated. From the screenshotted mean value of the exhaust flow, PyTesseract is then used to convert the image into text to display the value on the website. Other clickable buttons such as the buttons for “Valve Position”, “IGV”, “Temperature After Turbine”, “Turbine Power”, “Speed”, and “Run” also perform the similar functions as explained for the exhaust flow button function. However, for the “Choose File” button, the function of this button is to open the file explorer browser of the client’s or user’s computer to input their desired dataset and after the file has been submitted, the file will then be saved locally on a folder on the computer hosting the website.

As all the clickable buttons functions are completely programmed, the website was then tested. Firstly, to submit the desired dataset of the load of the gas turbine, the “Choose File” button is clicked, and the file explorer browser will be prompted open to select the file that is to be submitted.

After locating the load dataset from the file explorer browser, the file is then submitted through clicking the “Submit” button. Moving on, the “Run” button is then clicked which will then run a Python script to retrieve the data generated from the simulated Xcos model of the gas turbine model. After the website has been updated with the data retrieved from the Xcos model, the exhaust flow, speed, valve position, temperature after turbine, and turbine power buttons can be clicked to display the graphs of each of the parameters.

As the website html is generated and all button functions are programmed, the website can then be deployed using the Flask module in Python. The computer hosting the deployed website is acting as the local server for the website and any devices connected to the same Wi-Fi router with the local server will have access to the website hosted.

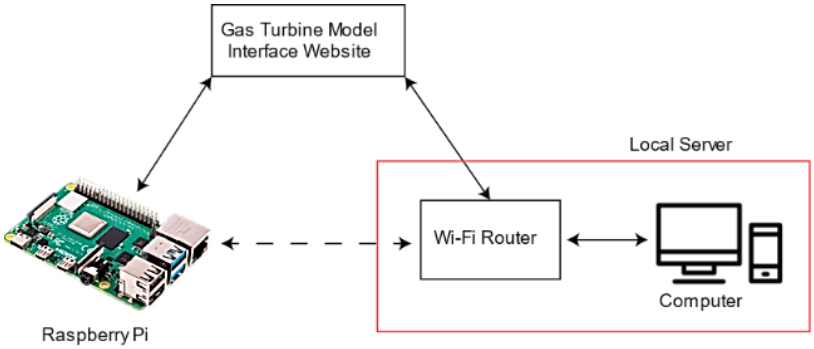

Fig.6 Connections for Deployment into Raspberry Pi

Fig.6 displays the connections that were made to make the Raspberry Pi have access to the gas turbine model interface website that was created in the previous subsection. The computer hosting the server and the Raspberry Pi must be connected to the same Wi-Fi router which is done wirelessly without the need of cables to have access to the website as the website is run locally on the computer instead on a cloud computing web server. Besides connecting to the same Wi-Fi router, the Raspberry Pi access the website for the gas turbine model interface by typing the URL of the hosted website in the web browser available in Raspberry Pi which is Chromium Browser. The URL of the hosted website is the IP address of the local server at port 5000.

After successfully connecting, the Raspberry Pi has the ability to conduct the simulation of the gas turbine model remotely regardless of the distance it is from the local server as long as it is still connected to the same Wi-Fi router. Loading the new datasets and generating the average values and graphs of the outputs of the gas turbine simulation is also possible to be done from the Raspberry Pi itself and this shows that the deployment of the website is successful as the gas turbine model interface website is able to be access remotely. Fig.7 presents the gas turbine model interface on Raspberry Pi accessed through Chromium browser.

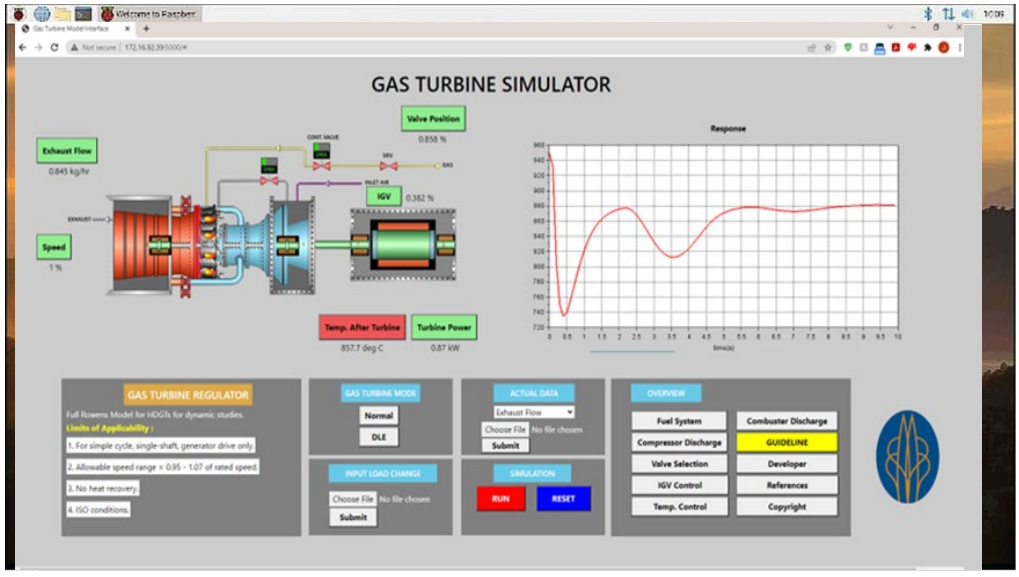

Fig.7 Web Interface Display of Temperature After Turbine Graph

Conclusion

In conclusion, the development and deployment of the gas turbine model interface was successfully conducted. The development of the gas turbine model, that was based on a previously done gas turbine model using MATLAB/Simulink, was converted into Scilab/Xcos model to as Scilab/Xcos is an open-source language compared to MATLAB/Simulink. From the generated Scilab/Xcos model, the acquiring of data was conducted using Scilab to select the needed dataset for the model and the graphs and mean values were acquired. With the Scilab/Xcos model built, the creation of the gas turbine model interface and deployment was done using Python and HTML. HTML was used to create the buttons and overall frontend of the website to host the gas turbine model interface, however, for the backend of the website that focusses on the function of the buttons, Python programming language was used instead. The functions of the button used mainly two (2) python modules which are PyTesseract and PyAutoGUI. PyTesseract is an optical character recigbutuib (OCR) tool for python while PyAutoGUI enables Python scripts to control the mouse and keyboard to automate interactions with other applications. Through using both modules, the functioning of the buttons is done to simulate the gas turbine model developed in Scilab/Xcos according to the dataset file submitted by the user. Upon simulating of the gas turbine model, the mean values and graphs generated for each output parameters of the gas turbine model, such as exhaust flow and speed, can be acquired. As for the deployment of the website, the Python module known as Flask was used. Flask is a micro web framework written in Python and using Flask, the website created was able to be deployed on a local server (which is the hosting computer) and any devices connected to the same Wi-Fi router as the local server will have access to the hosted website. Hence, the Raspberry Pi is able to establish a connection to the deployed website and through that, the remote access of the gas turbine model interface is able to be conducted as well as data collection and storage.

Acknowledgement

The authors are thankful to Ministry of Higher Education Malaysia (MOHE) and Universiti Teknologi PETRONAS for the support in carrying this research through grant PRGS (PRGS/1/2020/TK09/UTP/02/2) and YUTP (015LC0-382).

References

[1] Anheden, M. (2000). Analysis of gas turbine systems for sustainable energy conversion (PhD dissertation, Kemiteknik). Retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-2914

[2] Olumayegun, O., Wang, M., & Kelsall, G. (2016). Closed-cycle gas turbine for power generation: A state-of-the-art review. Fuel, 180, 694-717. https://doi.org/10.1016/j.fuel.2016.04.074

[3] Hazel, T., Peck, G., & Mattsson, H. (2014). Industrial Power Systems Using Dry Low Emission Turbines. IEEE Transactions On Industry Applications, 50(6), 4369-4378. https://doi.org/10.1109/tia.2014.2346697

[4] Nemitallah, M., Rashwan, S., Mansir, I., Abdelhafez, A., & Habib, M. (2018). Review of Novel Combustion Techniques for Clean Power Production in Gas Turbines. Energy & Fuels, 32(2), 979-1004.

[5] Faqih, M., Omar, M. B., & Ibrahim, R. B. (2022, August). Development of Rowen's Model for Dry-Low Emission Gas Turbine Dynamic Simulation using Scilab. In *2022 IEEE 5th International Symposium in Robotics and Manufacturing Automation (ROMA)* (pp. 1-5). IEEE.

[5] Asgari, H., Chen, X., Morini, M., Pinelli, M., Sainudiin, R., Spina, P., & Venturini, M. (2016). NARX models for simulation of the start-up operation of a single-shaft gas turbine.

Applied Thermal Engineering, 93, 368-376.
https://doi.org/10.1016/j.applthermaleng.2015.09.074

[6] Omar, M., Ibrahim, R., Abdullah, M., & Tarik, M. (2018). Modelling and System Identification of Gas Fuel Valves in Rowen's Model for Dry Low Emission Gas Turbine. 2018 IEEE Conference On Big Data And Analytics (ICBDA).
https://doi.org/10.1109/icbdaa.2018.8629705

[7] Faqih, M., Omar, M. B., Ibrahim, R., & Omar, B. A. (2022). Dry-Low Emission Gas Turbine Technology: Recent Trends and Challenges. *Applied Sciences*, *12*(21), 10922.

[8] Bahashwan, A. A., Ibrahim, R. B., Omar, M. B., & Faqih, M. (2022). The Lean Blowout Prediction Techniques in Lean Premixed Gas Turbine: An Overview. *Energies*, *15*(22), 8343.

[9] Omar, M., Tarik, M. H. M., Ibrahim, R., & Abdullah, M. F. (2017, November). Suitability study on using rowen's model for dry-low emission gas turbine operational performance. In *TENCON 2017-2017 IEEE Region 10 Conference* (pp. 1925-1930). IEEE.

[10] Tarik, M. H. M., Omar, M., Abdullah, M. F., & Ibrahim, R. (2017, November). Modelling of dry low emission gas turbine using black-box approach. In *TENCON 2017-2017 IEEE Region 10 Conference* (pp. 1902-1906). IEEE.

[11] Shalan, H. E., Hassan, M. M., and Bahgat, A. B. G. (2011). Parameter estimation and dynamic simulation of gas turbine model in combined cycle power plants based on actual operational data. J. Am. Sci. 7,303–310.

[12] Lima, Z., García-Vázquez, H., Rodríguez, R., Khemchandani, S., Dualibe, F., & del Pino, J. (2018). A System for Controlling and Monitoring IoT Applications. Applied System Innovation, 1(3), 26. https://doi.org/10.3390/asi1030026