

# A Distillation Column Control Via Graphical User Interface

Yeo Lip Qin, Jarinah Mohd Ali\*

<sup>1</sup> Department of Chemical and Process Engineering, Faculty of Engineering and Built Environment,  
Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia

\*Corresponding Author: jarinah@ukm.edu.my

Copyright©2022 by authors, all rights reserved. Authors agree that this article remains permanently open access under the terms of the Creative Commons Attribution License 4.0 International License

Received: 01 October 2022; Revised: 25 October 2022; Accepted: 20 November 2022; Published: 30 December 2022.

**Abstract:** A graphical user interface (GUI) is developed based on the renowned Wood and Berry distillation column using MATLAB version R2014a for controlling its top product composition. The Proportional Integral Derivative controller (PID) is proposed once the interface has been verified and compared to the literature data. The PID controller is then tuned using the Ziegler-Nichols method, trial-and-error method, and optimization toolbox. The ITAE, IAE and ISE for Z-N method is shown high error values with  $1.899e+53$ ,  $1.913e+51$  and  $3.083e+102$ . The transient response of these three methods is also discussed with optimization toolbox performing best, and Z-N method performing worst. Therefore, to improve the performance, an AI-based fuzzy logic controller is introduced to the system. By using the same value of gain from the Z-N method, the PID-Fuzzy logic controller is seen to provide a stable transient response with a lower value for the ITAE, IAE and ISE which is 1268.15, 71.87 and 157.35 respectively. Additionally, using the GUI, the material balance is also solved, and the McCabe-Thiele plot is generated with just a click. A handy tool, as such will ease the modelling procedure and simplify the control problem of a process system

**Keywords:** *Graphical User interface, Process control, Fuzzy Logic, Distillation column, MATLAB*

## 1. Introduction

Whether it is a computer or a smartphone, each hardware has a user interface (UI) that contributes to a great user experience. A good UI with a user-friendly feature will ensure that each task is carried out smoothly and the desired output is achieved in a few clicks [1]. The first user graphics interface (GUI) was developed by researchers at the Xerox Palo Research Center (PARC) in the 1970s and was the beginning of a series of computer graphics innovations to the GUI. Released in 1981, the Xerox Star was the first ever system that could be described as a fully integrated desktop computer [1]. As time progresses, the GUI releases began to include features like color, higher resolution displays and increased processing capacity.

GUIs have been also utilized in Chemical Engineering back in 1915 when it is integrated with several common industrial operations or processes. The modeling of chemical processes always involves many complex steps involving many parameters, and variety of reaction mechanisms, thus MATLAB is one of the preferable tools to develop and design such models. However, the programming for modeling in MATLAB tends to be difficult for users who is a beginner or unfamiliar with the software. Manual mathematical calculation will also consume time. Therefore, a user-friendly GUI is a better option, where the mathematical calculations and modeling output can be obtained in just one click. Other than that, the GUI developed will be able to give accurate data on the material balance since the data from the GUI has been verified with

**Corresponding Author:** Jarinah Mohd Ali, Department of Chemical and Process Engineering, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia, +60389118377

theoretical data.

In this paper, considering the renowned Wood & Berry distillation column as a case study, a GUI is developed to observe the response and control the top product composition by using a hybrid PID-Fuzzy controller. It will also compare to the Ziegler-Nichols tuned PID controller to highlight its advantages. By a few clicks, it will show the error analysis and transient response of the process. The McCabe-Thiele plot is also presented in the proposed GUI developed in MATLAB R2014.

## 2. MATLAB GUI development

There are mainly 5 steps in the development of the MATLAB GUI. Initially, MATLAB tool called GUIDE (GUI Development Environment) is used to organize the components such as Align Object, Menu Editor, Tab Order Editor, M-File Editor, Property Inspector, Object Browser and Run. GUIDE allows programmers to organize the GUI, select and synchronize the GUI components.

Next, is to use the MATLAB tool called Property Inspector (in GUIDE) to tag each of the component by giving a name to it and this is to give each component its characteristics, such as its color, the text it displays, and so on. After that, the figure is saved as Fig file and M file. The Fig file will contain the actual GUI that has been designed while the M-file contains the code that loads the encoding and callback for each of the GUI element. Both files are usually in the same directory. They are similar to the task of compiling and programming a GUI. When the GUI is placed in the Layout Editor, what it does will be saved in the Fig. File. When the GUI encoding is programmed, what it does will be saved in a similar M-file.

Finally, the coding is written to implement the actions set by each callback function in m-files. A callback function is often used to write and associate with the GUI figure or specific GUI component. It can complete actions in response to an event for its component which allows it to control GUI or component behavior. Such programming is called an event-driven programming [2]. Distillation is one of the oldest unit operations and is the most widely used separation technique in process industry. In this project, a simple binary distillation process is chosen which is the Wood & Berry distillation column. When modelling a binary distillation column via representing all the mathematical relationships, several assumptions need to be considered. Such considerations will facilitate the modelling of a binary distillation process. The assumptions for a 20-stage binary distillation column which have also been taken into consideration as [3]:

1. Constant relative volatility
2. Constant hold-up of liquid in all trays
3. Binary mixture
4. No chemical reactions
5. No vapor hold-up takes place in any stages

6. Constant pressure
7. Perfect mixing and equilibrium for the vapor-liquid at all stages

The first part that is developed is the material balance of the DC as shown in Figure 1. The edit box is used for users to input value such as feed flowrate, feed mole fraction, reflux ratio and show the value after calculation whereas panels are used to keep the GUI tidy and easier to be categorized.

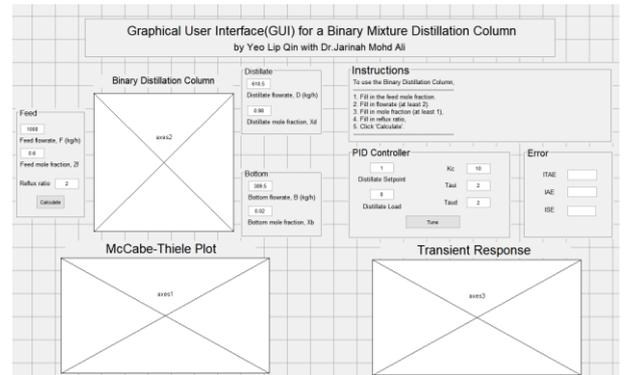


Figure 1. GUI designed in MATLAB GUIDE

After designing the layout of the GUI at GUIDE, it is time to incorporate the coding into an m-file. All the coding for calculation and plot will be done under the callback for pushbutton1 or 'Calculate' button. Figure 2 below shows the coding for the calculation of flowrate and mole fraction.

First, any value that the user inputs in the edit box is a string. Hence, a str2double function is used to convert string to double precision value. The string is obtained using get function from the handles from GUIDE. Then, an If statement is used to determine whether the user has entered in the edit box since it is allowed to have blank edit boxes. If the value entered from an edit box is a real number and is positive, this indicates there is already a value entered and the value will remain so. If edit box has not been entered any value, the GUI will solve for it using calculations. This is repeated for all six edit boxes.

```

84 % --- Executes on button press in pushbutton1.
85 function pushbutton1_Callback(hObject, eventdata, handles)
86 % hObject handle to pushbutton1 (see GCBO) %...%
89 F = str2double(get(handles.input1, 'string'));
90 Zf = str2double(get(handles.input2, 'string'));
91 Xd = str2double(get(handles.input3, 'string'));
92 D = str2double(get(handles.input4, 'string'));
93 Xb = str2double(get(handles.input5, 'string'));
94 B = str2double(get(handles.input6, 'string'));
95 if isreal(Zf) && rem(Zf,0.00001)==0;
96 Zf = Zf;
97 else
98 Zf = (Xd*D+Xb*B)/F;
99 end
100 if isreal(F) && rem(F,0.00001)==0;
101 F = F;
102 else
103 F = D + B;
104 end
105 if isreal(Xd) && rem(Xd,0.00001)==0;
106 Xd = Xd;
107 else
    
```

Figure 2. MATLAB Coding

After obtaining the values, a set function is used to display each of the value back to their edit box. Then, the McCabe

Thiele plot is plotted using those values. To show the graph plotted inside the GUI instead of a pop-up figure, an axes function is used to have the plot plotted inside the axes in GUI.

### 3. Results and Discussion

Figure 3 shows the final GUI developed. On the top right is a panel with detailed instructions on how to use the GUI whereas on the left is the material balance solver for the DC. As the user input enough inputs and click ‘Calculate’, the GUI will calculate and fill in the values for the unfilled edit boxes and a McCabe-Thiele plot will be plotted as well. The McCabe-Thiele plot is a simplified tool to help users understand the process inside the distillation column and it is also a method of calculating the number of theoretical trays required for a binary distillation column.

To give visual aid for the GUI, a picture of distillation column is shown in between the inputs. Below the instructions panel is the PID controller for the DC. The use of PID controllers will be further discussed in another section later.

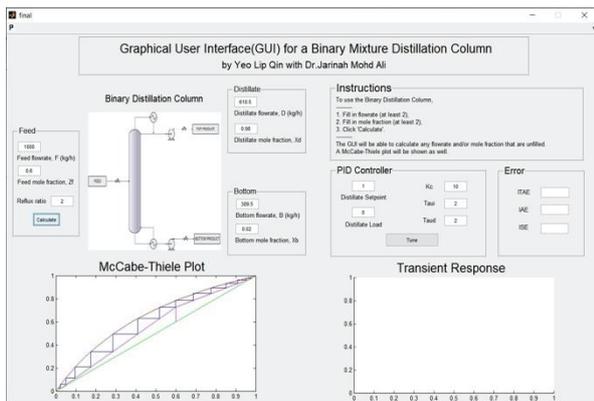


Figure 3. GUI for Distillation column

To validate the accuracy of values in the distillation column in the GUI, a model of distillation column is developed in MATLAB. The script can prompt the user to input values, then it will show the value that is not entered and plot the McCabe Thiele plot.

To compare the values, a set of theoretical data [4] is used where the feed flowrate, distillate and bottom flowrate, distillate mole fraction and bottom mole fraction is entered to calculate the feed mole fraction and number of trays. The results are shown in Table 1 below.

From the result, both script data and GUI data showed almost the same results. This is mainly because they are all assumed as steady state and the equations used for material balance are the same. For the number of trays, the theoretical data used Fenske equation to calculate while both script and GUI data used McCabe-Thiele plot to manually count the trays. The result of the McCabe-Thiele plot is shown in

Figure 4.

Table 1: Comparison of data from different set

Parameter	Theoretical Data	Script Data	GUI Data
Feed Flowrate (kg/h)	100	100	100
Distillate Flowrate (kg/h)	25	25	25
Bottom Flowrate (kg/h)	75	75	75
Distillate mole fraction	0.9	0.9	0.9
Bottom mole fraction	0.1	0.1	0.1
Feed mole fraction	0.3	0.3	0.3
Number of trays	5.97	6	6

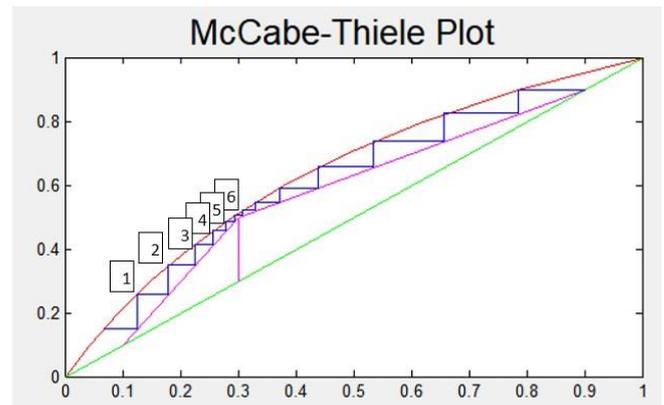


Figure 4: Number of trays in McCabe-Thiele plot

Next, since the GUI has been completed, a control system is implemented in the distillation column. There are many control strategies applied to distillation column system. In this work, a PID controller is implemented to control the setpoint of the top product composition. The distillation column model used is Wood and Berry as this is the most commonly used model in distillation columns and it has also been verified and validated experimentally [5].

process. It consists of a setpoint for the top product composition, a disturbance load, PID block controller, subsystem for transfer function, a mux block with scope for output response, and another subsystem for the figure of merits (ITAE, IAE, ISE). The Simulink model is shown in Figure 5.

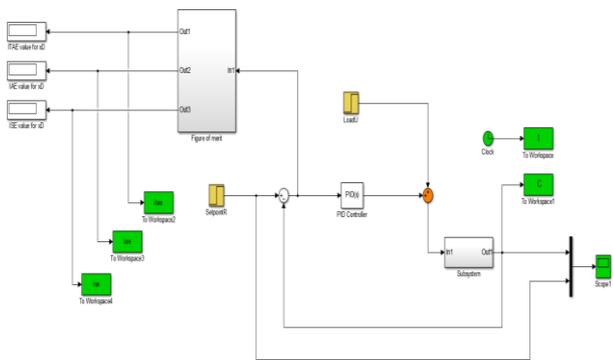


Figure 5: Simulink Block Diagram of control system

In this work, the transfer function which normally has two inputs, and two outputs is simplified to focus on only the top product and hence only one input from user and one output is produced in the scope.

The main idea here is to allow the user to input the value of the controller gain ( $K_c$ ), integral time constant ( $\tau_i$ ) and derivative time constant ( $\tau_d$ ) in the GUI, the inputs will be registered in the Simulink model and simulation will start as they click ‘Tune’. However, this is not achievable as the GUI has a different “hidden” workplace unlike a normal MATLAB script. Hence, when the user inputs a value in the GUI, it is undetectable by the Simulink model.

This limitation can be easily overcome by using a code based PID control function like PID tune function which eliminates the need for Simulink. However, PID tune function is only applicable for SISO (Single Input Single Output) transfer functions whereas the transfer function used is a MIMO (Multiple Input Multiple Output).

As a result, the GUI is only able to show the transient response and the users must input the gain and the setpoints in Simulink manually. The GUI is shown in Figure 6 with an untuned transient response with setpoint of 0.8 and no load. The figure of merit which includes integral time absolute error (ITAE), integral square error (ISE), and integral absolute error (IAE).

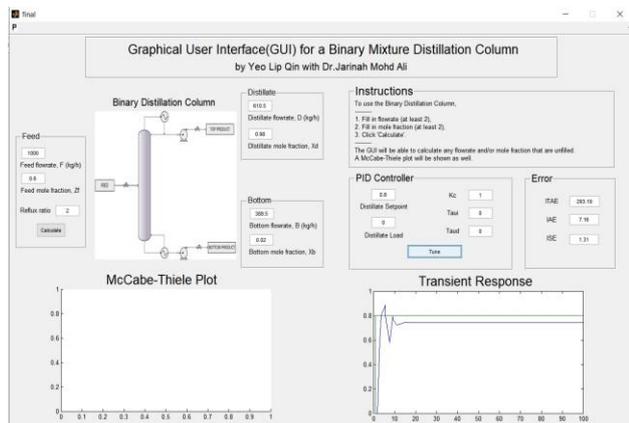


Figure 6: GUI with control system and transient response

Tuning can be further improved as the untuned transient

response has shown offset and overshoot. To tune the process, 3 methods are used which are Ziegler-Nicholas method, trial-and-error method, and optimization toolbox in MATLAB. The results are shown in Figure 7 and Table 2 below:

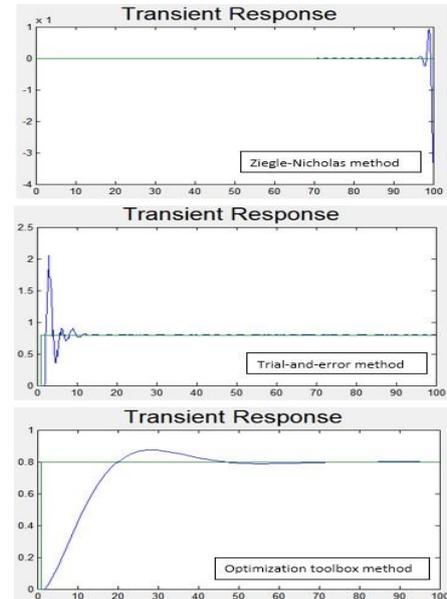


Figure 7: Transient response based on three methods

Table 2: Results parameter from the three methods

Parameter	Untuned	Ziegler-Nicholas	Trial and error	Optimization tool
Simulation time(s)	100	100	100	100
$K_c$	1	2.15	1.85	0.1314
$P_u$	-	3.71	-	-
$\tau_i$	0	3.309	1.45	0.0098
$\tau_d$	0	4.306	0.9	-0.6718
ITAE	293.19	1.899e+53	12.14	97.09
IAE	7.16	1.913e+51	2.74	8.58
ISE	1.31	3.083e+102	1.73	4.11

From the transient response, it is observed that the Z-N method can keep the response at the setpoint for a long time, but it gets into chaos later as the response curve diverges. The values of ITAE, IAE and ISE is also off the chart as the response gets wilder as time goes on.

On the other hand, both trial-and-error method and optimization toolbox give relatively better response, but trial-and-error method can achieve a response with low settling time and no offset but there is a high overshoot and signs of overdamping while Z-N method produces a low overshoot and stable response, but the settling time is significantly longer. Both methods showed a good result on the figure of merits, but the errors on trial-and-error method is relatively lower than Z-N method [6].

To tackle the problem faced in using Z-N method, a fuzzy logic controller is introduced. The fuzzy logic is designed using Fuzzy Logic Designer in Simulink MATLAB. Figure 8 shows the Fuzzy Logic Designer framework.

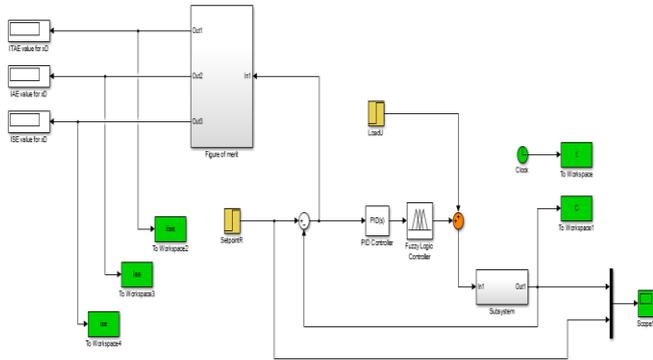


Figure 8: Fuzzy Logic Controller model in Simulink

The fuzzy controllers rule base is composed of 9 (3×3) rules as shown in Figure 9. For the rules for fuzzy control, *e* and *de* represent the significant error and rate of change of significant error respectively. Those two parameters are significant for the fuzzy logic controller to work. Furthermore, Mamdani inference system is used as inference engine and the method for defuzzification in the fuzzy controllers is centroid method.

After applying fuzzy logic controllers into the system, the gains from the Z-N method were used. The transient response after the addition of the fuzzy logic controller shown in Figure 10 has shown a more stable response compared to the transient response produced by the Z-N method while Table 3 shows the results of the comparison of the Z-N method with the Z-N method augmented with a fuzzy logic controller.

The error also shows a large reduction after the addition of fuzzy logic controllers. From the comparison, the hybrid controller has shown better results because the hybrid controller can overcome the shortcomings in the PID controller thus improving the system stability.

	<b>e</b>	<b>N</b>	<b>Z</b>	<b>P</b>
<b>de</b>		<b>N</b>	<b>Z</b>	<b>P</b>
<b>N</b>	<b>N</b>	<b>N</b>	<b>Z</b>	<b>P</b>
<b>Z</b>	<b>N</b>	<b>Z</b>	<b>P</b>	<b>P</b>
<b>P</b>	<b>Z</b>	<b>P</b>	<b>P</b>	<b>P</b>

Figure 9: Fuzzy rules

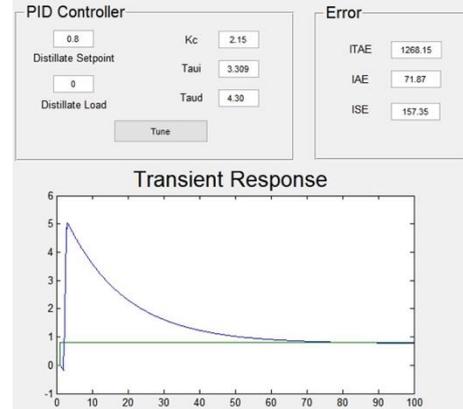


Figure 10: Transient response of hybrid Fuzzy-PID in GUI

Table 3: Comparison of single PID and Fuzzy-PID controller

Parameter	Single PID	Fuzzy-PID
Simulation time(s)	100	100
$K_c$	2.15	2.15
$P_u$	3.71	3.71
$\tau_i$	3.309	3.309
$\tau_d$	4.306	4.306
ITAE	1.899e+53	1268.15
IAE	1.913e+51	71.87
ISE	3.083e+102	157.35

#### 4. Conclusion

This study has developed a GUI which allows distillation column process to be more easily accessible to the users and replace complex model development with programming using MATLAB. Users will be able to use the MATLAB GUI developed with minimum knowledge of MATLAB. With the GUI, the mass balance as well as the mole fraction can be determined after input is entered with just one click. McCabe Thiele’s plot will also be displayed in the GUI using the same input to get the theoretical number of trays. The GUI will be able to save users time compared to manual calculations which waste time and energy. The hybrid Fuzzy-PID controller. system is also embedded in the GUI for user to observe the control performance of the distillation column.

#### 5. Acknowledgements

We are very grateful to Universiti Kebangsaan Malaysia for supporting this research.

## **6. References**

- [1] S. R. Banka, S. M. Dutta. Interactive MATLAB program for design of multicomponent distillation column using FUG method, *International Journal on Advances in Engineering Technology and Science*, 21–25, 2016.
- [2] J.P. Cardoso, K. Tannous. Development of a computational tool for designing multicomponent distillation columns, *Computer Applications in Engineering Education*, 908-922, 2020.
- [3] S. Lone. Modeling and Simulation of a Hybrid Process (Pervaporation + Distillation) using MATLAB, *Journal of Chemical Engineering & Process Technology*, 1-10, 2015.
- [4] N.A. Rahim, Z. Ahmad. Graphical user interface application in MATLAB environment for water and air quality process monitoring. *Chemical Engineering Transactions*, 97–102, 2017.
- [5] E.A. Lagarda-Leyva, A. Bueno-Solano, H.P. Veja-Valdez, D.O. Machado. Dynamic model and graphical user interface: A solution for the distribution process of regional products. *Applied Sciences (Switzerland)*, 1–15, 2020.
- [6] Z.L. Edaris, S. Abdul-Rahman. Performance comparison of PID tuning by using Ziegler-Nichols and Particle Swarm Optimization approaches in a water control system. *Journal of Information and Communication Technology*, 203–224, 2016.