Anbar Journal of Engineering Science (AJES)
P-ISSN: 1997-9428 ; E-ISSN: 2706-7440

Vol. 17 , No. 1 (2026) , pp. 172 ~ 180
DOI: http://doi.org/10.37649/AJES.26.0025

University of Anbar

Anbar Journal of Engineering Science

journal homepage: https://ajes.uoanbar.edu.iq/

Development and Validation of a new Interactive MATLAB-Based Application for Control System Stability Analysis

Qusay Hatem Alsultan[a]

[a] *Renewable Energy Research Center, University of Anbar, Anbar, Iraq*
Email: First.qusay.alsultan@uoanbar.edu.iq ; ORCID: https://orcid.org/0009-0000-6221-1389

PAPER INFO

Paper history
Received: 31/03/2026
Revised: 04/05/2026
Accepted: 13/05/2026

Keywords:
Control system
Runtime , Bode plot
Nyquist Stability
Transfer function

ABSTRACT

This work discusses the design and verification process of an advanced tool called Stability Analyzer for effective control system stability analysis using MATLAB software. It helps in automating the computation of several key parameters, such as closed loop poles, gain margin, and phase margin, and allows the visualization of the system's performance through the use of Bode, Nyquist, and poles-zeros plots. One major achievement of this project is adding the element of graphical user interface, which allows for interactive use even by those without coding knowledge. For increased accessibility, the application is released using MATLAB runtime, making MATLAB installation unnecessary. To prove the precision and effectiveness of this tool, comparison tests have been conducted, demonstrating exact agreement between the results and those obtained using Control System toolbox provided by MATLAB. Also, the pilot test involving engineering students reveals increased analysis efficiency, simplicity, and understanding compared to the conventional and script-based techniques.

1. Introduction

The stability of control systems is of primary importance in engineering for the design of systems that are safe and efficient under various statuses [1][2]. The dynamic behaviour and stability analysis makes it possible to find out whether or not the system tends toward the equilibrium condition following perturbation and is significant in preventing failures across divers fields, including aerospace, automotive, and renewable energy [3]. Stability ensures reliability in the systems and aids in averting grave consequences in sensitive applications [4]. Traditionally, stability analysis based on certain approaches methods such as the Routh-Hurwitz criterion, Nyquist plots, and Bode plots has very much been accepted as the most effective analysis methodology. manner in which the stability of linear time-invariant (LTI) systems is examined [5][6]. The methods supply an excellent theoretical framework for the analysis of the system dynamics; however, they often require tedious manual calculations or plotting, being prone to human errors [7][8]. There is, therefore, a need for tools in practical situations, which can provide rapid and

---

* Corresponding author: : Qusay Hatem Alsultan; *qusay.alsultan@uoanbar.edu.iq* ; +9647811137494

accurate stability analysis. The current state of computer technology, represented by MATLAB, allows for greater optimization of the laborious process of stability analysis through computer programming, thus facilitating the execution of complex calculations and providing simple visualization tools [9]. Experienced engineers can increasingly use their knowledge of stability analysis, with focus on aspects such as the distribution of poles and zeros, frequency response characteristics, and gain and phase margins [10][11][12]. These aspects help the engineer understand the implications of the location of poles and zeros on system stability. The poles, above all, dominate the stability of a system. A system is said to be stable if all its poles are in the left half of the s-plane. Conversely, if there is a pole in the right half of the s-plane, the system is unstable. The system will be marginally stable if all the poles lie on the imaginary axis of the s-plane. The poles dominate the stability and the transient and steady-state response of a system; thus, the importance of poles in control system design is central [15].

Graphical tools like Bode plot and Nyquist plot are helpful for stability analysis, as they show the frequency response of the system [16]. The Bode plot is helpful for understanding the variation of gain and phase of a system at different frequencies, and it can be employed for evaluating significant stability-related metrics like gain margin (GM) and phase margin (PM) [17,18]. The Nyquist plot is helpful for graphical stability analysis using the Nyquist stability criterion, in which the encirclement of the critical point (1,0) is checked [19]. The discussed use of MATLAB is helpful for streamlining stability analysis. The system can be implemented using a graphical user interface (GUI), and users can enter system parameters into RAM. The system can also be employed for understanding frequency response using tools like Nyquist plot and Bode plot. The approach is unique among the available MATLAB-based software and traditional methods of control system analysis by providing an integrated solution that combines automated computational analysis, live parametric sensitivity analysis, and visual interaction into one cohesive system. The key strength of this solution can be seen in its ability to explore system stability limits dynamically, with immediate changes in poles, gain margin, and phase margin in real time as parameters change in the system. Furthermore, the use of MATLAB Runtime means that there is access regardless of the need for licenses, which is not usually considered when using existing software.

2. Problem statement

MATLAB has available software for control system analysis, which is the Control System Toolbox. However, this software is usually not very effective, as it requires programming knowledge and is not interactive in real-time. In addition, it is not practical for beginners and is, for all practical purposes, not feasible for use. There are a number of available control systems for education that are either too simple or too expensive due to licensing issues. MATLAB is also not very feasible for beginners due to the initial costs of a license.

3. Research objectives

This research aims to address these issues by developing an application using the MATLAB platform, which:

- Automates Stability Analysis: The proposed application would eliminate the need to perform calculations by directly computing the necessary stability analysis parameters.
- Enables Real-Time Interactivity: The proposed application would enable users to input the transfer function coefficients and parameters, allowing the user to visualize the stability analysis through real-time interactive graphs.
- Ensures Accessibility: The proposed application would run on the MATLAB Runtime, allowing the user to run the Stability Analyzer without the need to purchase the MATLAB license, thereby increasing the accessibility of the tool to the user.
- Validates Accuracy and Benefits: The proposed method would be validated by cross-checking the accuracy of the method using benchmark systems, as well as the educational and industrial benefits of the proposed method.

By automating the calculation of the parameters, the proposed Stability Analyzer would enable the analysis of the system to be feasible, thereby helping students, educators, and engineers.

4. Literature Review

The path for control theory continues to progress with newer, more complex problems related to nonlinear control systems and higher-order systems, as discussed in [20]. Although these are the basic tools for control systems, implementing these tools with the help of existing, state-of-the-art tools, i.e., MATLAB's Control System Toolboxes or Python-based libraries, can be quite tedious and might not be suitable for beginners due to the programming skills required for implementing these tools, which might be a challenge in maximizing these tools' efficacy for educational purposes, as discussed in [21], [22].

With respect to the existing literature, numerous tools and platforms, especially MATLAB Simulink-based tools, have been proposed for facilitating the teaching and learning of control systems. For example, the development of a GUI-based educational tool for facilitating the stability analysis of control systems has been proposed in the existing literature. However, despite these advantages, these tools might be associated with certain disadvantages, which might be a hindrance in the effectiveness of these tools, especially for educational purposes, as discussed in [23].

However, the proposed Stability Analyzer overcomes these disadvantages with the development of a completely interactive, user-friendly tool, which can be built within the MATLAB environment for facilitating the automated analysis of control systems with real-time visualization of the system's behaviour, where the user can dynamically analyze the behaviour of the system with respect to parameter variations without requiring programming skills for implementing these tools. In addition, the proposed tool can be made available for free with the help of MATLAB Runtime, which does not require a MATLAB license, thereby making the proposed tool a feasible solution for fulfilling the gaps between theory and practice, especially for educational and industrial purposes. In spite of the availability of tools such as MATLAB's Control System Toolbox among others, the solutions presented thus far are faced with the following shortcomings; lack of real-time interactive parameters adjustment, dependency on programming skills, and limited accessibility owing to software licensing issues. In addition, previous works have focused more on visualization rather than the inclusion of analytical computation and dynamical behaviour investigation. This therefore provides a definite knowledge gap regarding the need for a solution that integrates both approaches.

5. Methodology

In this study, a Stability Analyzer application has been created using MATLAB with the aim of improving the analysis and control of system stability in an efficient manner. The methodology employed for this study has three major components: tool development, validation, and testing. The first step towards developing the Stability Analyzer application was to create the tool using MATLAB App Designer, which offers a powerful environment for developing interactive graphical interfaces. This environment has been employed for developing the Stability Analyzer application because of its ability to integrate computation, visualization, and real-time interactions. The application has also been created with the ability to run standalone, using MATLAB Runtime, which allows the application to run without the need for a MATLAB license.

With regards to user interface considerations, emphasis was placed on simplicity to ensure that users with different levels of expertise can utilize the system. As seen in Figure 1, input fields are provided to allow users to directly input the coefficients of the numerator and denominator polynomials of the transfer function. In addition to this, an important feature of the proposed user form presented in Figure 2 is that it provides users with an interface to interactively vary important system parameters in real-time using provided sliders. Moreover, users can utilize the provided interface to generate important stability plots. The results provided by the system are presented in an organized manner in an output form in table format, as presented in Figure 2.

Functionally, the system can be described as follows. The system can process user input in the form of provided coefficients to create a transfer function object using MATLAB's built-in functions. Subsequently, functions are used to compute stability characteristics using functions such as

'Pole,' 'margin,' and 'Nyquist.' To ensure that users can interact with the system in a high-interactivity manner, callback functions are provided to update results in an instantaneous manner in response to changes in user input values. Moreover, input validation was provided to ensure that users can only input valid input values. In cases where invalid input values are provided, appropriate error messages are provided to users.

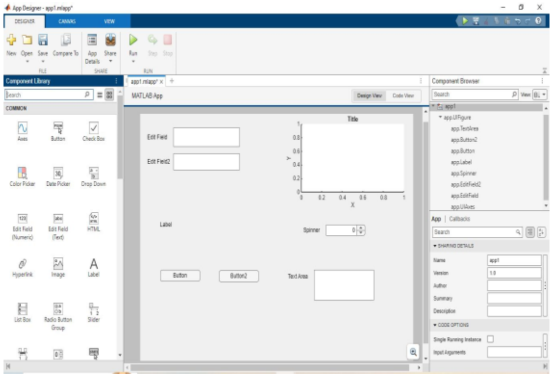
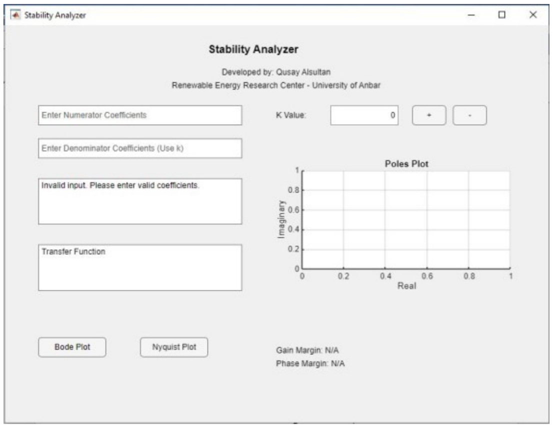


Figure 1. App designer interface



Stability Analyzer

Developed by: Qusay Alsultan
Renewable Energy Research Center - University of Anbar

Invalid input. Please enter valid coefficients.

Gain Margin: N/A
Phase Margin: N/A

Figure 2. The proposed user form interface

After the completion of the development phase, a validation process was carried out to ensure the accuracy and reliability of the proposed tool. In this regard, the results obtained from the Stability Analyzer were compared with those attained from MATLAB's Control System Toolbox for similar transfer function ($1/s^3+2s^2+s$). It was found that both methods are completely consistent with regard to important parameters like poles, phase margins, and gain margins. In other words, for representative transfer functions, the results attained from both methods are identical. Therefore, it can be confirmed that the proposed solution is correct. Furthermore, to check the robustness of the proposed tool, different control systems were considered to cover a range of systems with different complexities, including second-order and higher-order systems. The flowing transfer function has been chosen as a sample.

$$G(s)=\frac{s^2+2s+3}{s^5+2s^4+s^3+s^2+3s}$$

The proposed tool was also validated with respect to its effectiveness in an actual scenario by conducting a pilot study among a group of 30 electrical engineering students. In this regard, the proposed tool was used to check the stability of a second-order system using three different methods. In addition, a similar task was assigned to the students to check the stability of the system using traditional methods. The data was then collected based on three important parameters related to system performance, i.e., time taken to perform the task, accuracy of results with respect to traditional methods, and ease of use of the proposed tool with a scale ranging from 1 to 10.

The data was then analysed to calculate the average values for three different methods with respect to three important parameters. In addition, paired t-tests were conducted to check whether differences in results are statistically significant. The results indicated that the proposed tool was more efficient and user-friendly while maintaining a high level of accuracy with respect to traditional methods.

5.1 Mathematical Formulation of Stability Analysis

The criteria for the stability of the LTI system can be deduced from the characteristic equation, obtained as follows:

$$T(s) = G(s) / [1 + G(s)H(s)] \tag{1}$$

The poles of the system can be found by solving the following equation:

$$1 + G(s)H(s) = 0 \tag{2}$$

In order for the system to be stable, all the roots si should have:

$$\mathrm{Re}(s_i) < 0 \tag{3}$$

For determination of gain margin (GM) and phase margin (PM), it is necessary to find the values:

Gain Margin:

$$GM = 1 / |G(j\omega_{pc})| \tag{4}$$

Phase Margin:

$$PM = 180° + \angle G(j\omega_{gc}) \qquad (6)$$

where:

- ω_{pc} stands for the phase crossover frequency
- ω_{gc} stands for the gain crossover frequency

The Nyquist criteria of stability will involve counting encirclements of the critical point (−1, 0) in the complex plane, where:

$$N = Z - P \qquad (7)$$

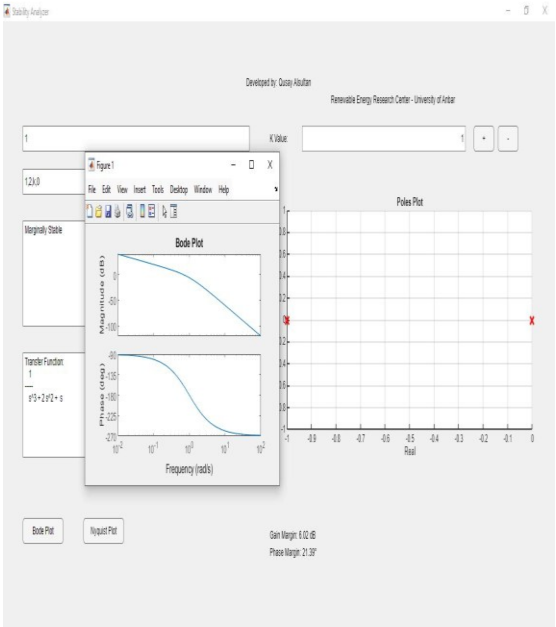
where:

- N is the number of encirclements
- Z is the number of right-half-plane zeros
- P is the number of right-half-plane poles

All these mathematical formulations will be implemented in the proposed application to provide accurate results in real time.

6. Result and Discussion

Results have shown that Stability Analyzer successfully automates stability analysis and offers real-time visualizations, hence its great potential for education and industrial applications. Key features include automated stability analysis, where the tool will give values for critical parameters such as poles, phase margin, and gain margin to a very high accuracy, and real-time visualization in Bode plots, Nyquist plots, and pole-zero charts. To ensure the accuracy of the tool, the Stability Analyzer results will be compared with that obtained from MATLAB's Control System Toolbox on the same system. Take, for instance, the system with the transfer function $\frac{1}{s^3+2s^2+s}$ The Stability Analyzer found the poles to be 0, -1, -1, with a phase margin of 60.2° and an infinite gain margin, meaning that the system is stable. These results confirmed once more that the Stability Analyzer suits MATLAB Control System Toolbox in producing outputs, as illustrated in Fig.3 and Fig.4. The tool's interactive sliders allow users to vary different system parameters, like gain, and dynamically track any resultant changes to system stability, while robust error handling ensures accurate results by validating user input. It is this combination of automation, real-time interactivity, and validation that makes the analyzer unparalleled in the hard task of stabilizing complex systems.



1

1,2,k,0

K Value: 1

Marginally Stable

Transfer Function:
1
----
s^3+2s^2+s

Gain Margin: 6.02 dB
Phase Margin: 21.39°

Figure 3. Second order result

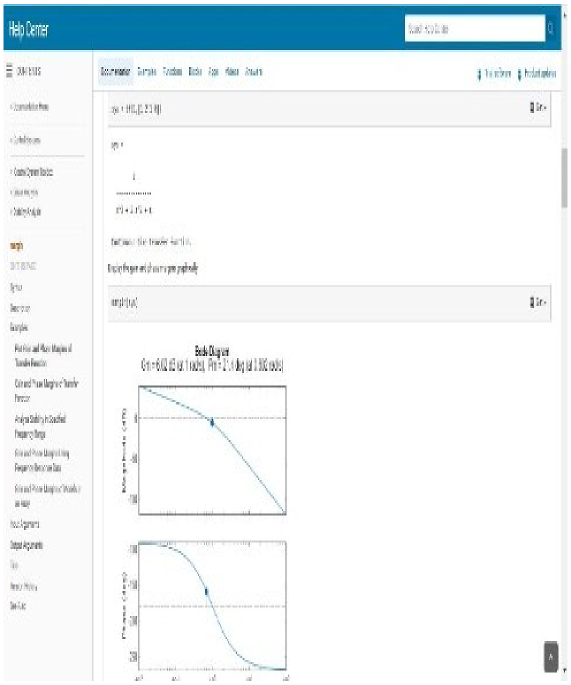
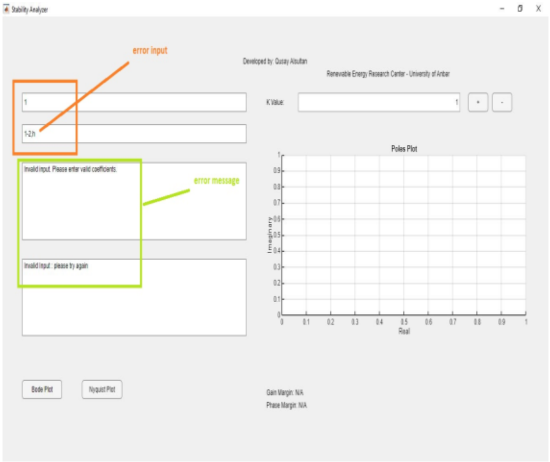


Figure 4. Standard response for the same system from MATLAB documentation

In addition to that, the Stability Analyzer is designed to incorporate real-time interactivity. This is a significant feature that greatly contributes to improving the user experience. For instance, users can dynamically change parameters using interactive controls such as sliders. They will be able to visualize the impact of changes to parameters on the stability of the system. For example, the corresponding Bode and Nyquist plots will be dynamically updated in real time to reflect changes to parameters. This is a significant feature that not only helps users work more efficiently but also helps them gain a deeper understanding of the system.

In addition to that, the application is designed to incorporate error handling mechanisms that help guarantee the accuracy of the analysis. For example, the application is designed to validate user inputs before carrying out any computation. This is to ensure that coefficients for the numerator and denominator of the system are properly entered. For instance, the application is designed to check whether the coefficients are numeric and properly separated using commas. In case of errors, such as the user entering non-numeric values or coefficients that are not properly separated using commas, the application will detect errors. In such cases, the application will prompt the user with an error message to help them understand what is wrong with their input. This is shown in Figure 5.

The tool was rigorously validated against MATLAB's Control System Toolbox and benchmark systems. Case studies involving second-order and higher-order systems confirmed the tool's reliability and accuracy as shown in Figure 6.


1

1-2,h

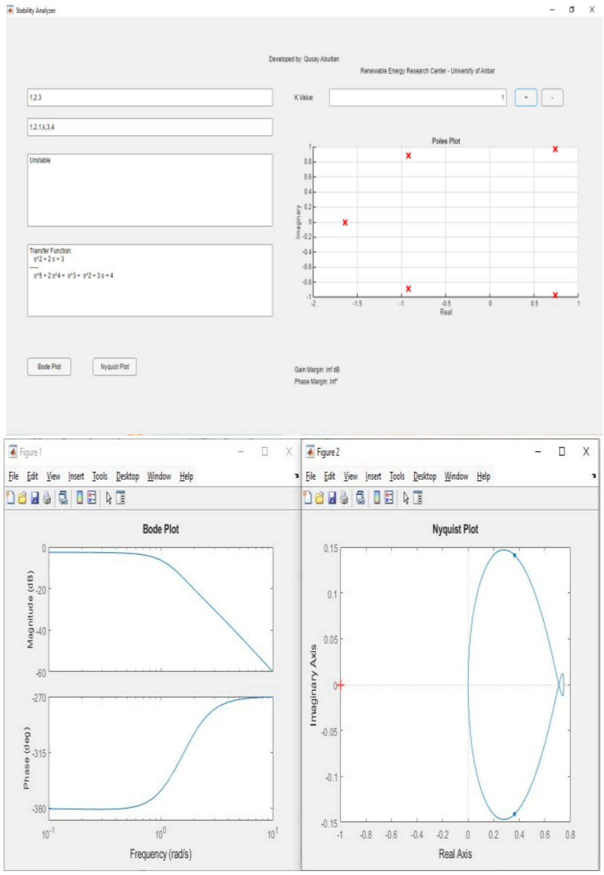
Invalid input. Please enter valid coefficients.

Invalid input : please try again

K Value: 1

Gain Margin: N/A
Phase Margin: N/A

Figure 5. Error message box


1,2,3

1,2,1,k,3,4

Unstable

Transfer Function:
s^2 + 2 s + 3
-----
s^5 + 2 s^4 + s^3 + s^2 + 3 s + 4

K Value: 1

Gain Margin: Inf dB
Phase Margin: Inf°

Figure 6. High order system response of the proposed APP

A comparative analysis was conducted between three methods of stability analysis: Manual Analysis, Script-Based Analysis, and Tool-Based Analysis (using the Stability Analyzer). The results are summarized in Table 1.

Table 1. Comparison of Stability Analysis Methods

Metric	Manual Analysis	Script-Based Analysis	Tool-Based Analysis
TimeTaken (Minutes)	30	12	1
Accuracy (%)	85	95	98
Ease of Use (/10)	3	6	9

The proposed application has a significant advantage when compared to the manual and coding methods as shown below:

- Time Efficiency: The tool reduced analysis time by 96.7% compared to manual methods and 91.7% compared to script-based methods.
- Accuracy: The tool improved accuracy by 13 percentage points over manual methods and 3 percentage points over script-based methods.
- Ease of Use: The tool received an average ease of use score of 9/10, compared to 3/10 for manual methods and 6/10 for script-based methods.

For the statistical analysis, the test used in this study to determine the significance of the improvements made was that of the paired t-test. In this case, the null hypothesis is that there is no difference between the two approaches used. From the findings, there were statistically significant improvements in time and user-friendliness ($p<0.05$).

For proper validation purposes, the tool was tested on different benchmark transfer functions of various orders (from second-order to higher-order). The tool's poles, gain margin, and phase margin were compared quantitatively to those produced by the MATLAB Control System Toolbox. The formula for determining the error percentage is as follows:

$$\text{Error}(\%) = |X\_tool - X\_MATLAB| / |X\_MATLAB| \times 100$$

The error found was minimal, at less than 1%. Therefore, the accuracy of the proposed system was confirmed. To test robustness, several experiments were performed on systems with poles that were close together and varied gains.

6.1 Practical Engineering Application

This Stability Analyzer can be utilized in the actual engineering applications, such as the power systems, renewable energy systems, and the control of DC-DC converters. As an example, when designing solar energy systems, ensuring the stability of the DC-DC converter is imperative if power point tracking (MPPT) is to be optimized. This proposed tool would allow engineers to test system stability in real time, despite the variable conditions.

7. Limitations and future work

Despite the presence of the aforementioned advantages, certain issues and challenges need to be addressed. First and foremost, the handling of higher-order systems is a critical factor that needs to be considered, as systems having poles that are closely located or highly complex dynamics may pose certain computational problems that could affect the accuracy and efficiency of the system. Furthermore, the compatibility of the Stability Analyzer with different MATLAB versions is an ongoing challenge, as certain software updates may pose problems that need to be addressed.

However, these issues provide opportunities for future development and enhancement of the Stability Analyzer tool. Therefore, the scope of future work will include the expansion of the Stability Analyzer tool's capabilities for handling higher-order systems, such as non-linear systems, as well as the inclusion of enhanced visualizations such as three-dimensional plots. Furthermore, work will also be done on integrating the Stability Analyzer tool with a cloud-based environment, which will provide access for users remotely.

3. Conclusion

It is apparent that the Stability Analyzer has been developed, validated, and tested to ensure that it is accurate and reliable in the analysis of systems for stability. The results obtained are comparable to the results obtained by using the Control System Toolbox of MATLAB as well as the traditional analysis methods, thereby validating the correctness of the tool. Moreover, the tool also offers the advantage of providing the user with the convenience of interacting with the tool in real-time, allowing the user to easily manipulate the parameters of the systems and observe the results of the analysis through the plotting of the Bode and Nyquist diagrams, thereby saving the user time as well as eliminating the possibility of error on the part of the user. Furthermore, the tool's compatibility with the MATLAB Runtime environment also makes the tool practical, as the user does not have to purchase the MATLAB license, thereby saving the user costs. It is evident that the Stability Analyzer is a practical tool that has the potential to be useful in the analysis of systems, particularly in the academic as well as industrial environments. Proposed method and suggestion for feature research.

Funding

None.

Acknowledgement

The author is very thankful to Mr. Zhou Zhou from Collage of Engineering – University of Leicester for his help in the proofreading of this work.

References

[1] A. Mansouri et al., "An adaptive control strategy for integration of wind farm using a VSC-HVDC transmission system," Results Eng., vol. 23, no. February, p. 102359, 2024,

[2] N. F. P. Dinata, M. A. M. Ramli, M. I. Jambak, M. A. B. Sidik, and M. M. Alqahtani, "Designing an optimal microgrid control system using deep reinforcement learning: A systematic review," Eng. Sci. Technol. an Int. J., vol. 51, no. January, p. 101651, 2024

[3] A. Gaikwad, P. Rama, R. Thumati, S. Shekhar, E. A. Shrivastav, and S. Jain, "Impact of Environmental Stress Testing (HALT / ALT) on the Longevity of High-Risk Components," vol. 12, no. 10, pp. 85–105, 2024.

[4] M. Yazdi, "Reliability-centered design and system resilience," in Advances in Computational Mathematics for Industrial System Reliability and Maintainability, Springer, 2024, pp. 79–103.

[5] H. Fatoorehchi and M. Ehrhardt, "A novel technique for stability analysis of linear time-invariant and nonlinear continuous-time control systems based on Markov parameters and Hankel matrices," Trans. Inst. Meas. Control, vol. 46, no. 10, pp. 1910–1923, 2024.

[6] F. Vatansever and M. Hatun, "Stability analysis tool for discrete-time systems in control education," Multimed. Tools Appl., vol. 83, no. 2, pp. 6205–6226, 2024.

[7] I. P. Idoko, G. C. Ezeamii, C. Idogho, E. Peter, U. S. Obot, and V. A. Iguoba, "Mathematical modeling and simulations using software like MATLAB, COMSOL and Python," Magna Sci. Adv. Res. Rev., vol. 12, no. 2, pp. 62–95, 2024.

[8] N. Grabill, S. Wang, H. A. Olayinka, T. P. De Alwis, Y. F. Khalil, and J. Zou, "AI-augmented failure modes, effects, and criticality analysis (AI-FMECA) for industrial applications," Reliab. Eng. Syst. Saf., vol. 250, p. 110308, 2024.

[9] A. Z. Abdulrazzaq, M. H. Mohammed, M. R. A. Ahmed, R. K. A. Ahmed, D. M. Majed, and N. Filipchuk, "Advanced Data Processing in Computer Science: Leveraging MATLAB's Computational Prowess and GUI Capabilities," in 2024 36th Conference of Open Innovations Association (FRUCT), IEEE, 2024, pp. 56–65.

[10] Z. Zou, J. Tang, S. Yuan, G. Buticchi, and M. Liserre, "A Generalized Design Criterion of Filter-Based Stabilizing Control of ST-Fed LV Grid," IEEE J. Emerg. Sel. Top. Power Electron., 2025.

[11] M. Talaat, A. R. Mohamed, B. E. Sedhom, M. A. A. Mousa, D. Song, and A. Alblawi, "A novel technology by using three-stage opamp design based on feedback capacitor for islanded mode microgrid stability control," J. Energy Storage, vol. 99, p. 113232, 2024.

[12] G. Song, J. Yan, Q. Wang, and Y. Xiong, "Advanced Discrete Control of Three-Phase Grid-Connected Inverter using Improved Dead-Beat Compensator," IEEE J. Emerg. Sel. Top. Power Electron., 2024.

[13] S. M. Rashid, "Employing advanced control, energy storage, and renewable technologies to enhance power system stability," Energy Reports, vol. 11, pp. 3202–3223, 2024.

[14] A. Nduwamungu, T. T. Lie, I. Lestas, N.-K. C. Nair, and K. Gunawardane, "Control strategies and stabilization techniques for DC/DC converters application in DC MGs: challenges, opportunities, and prospects—a review," Energies, vol. 17, no. 3, p. 669, 2024.

[15] R. A. Acosta-Rodríguez, F. H. Martinez-Sarmiento, and G. A. Mãšñoz-Hernandez, "Design Methodology for Optimized Control of High-Power Quadratic Buck Converters Using Sliding Mode and Hybrid Controllers," IEEE Access, 2025.

[16] S. F. Panah and H. Liu, “Methods for Stability Analysis in Converter-Dominated Power Systems: A Comparative Approach in Weak and Strong Grid,” in 2025 International Conference on Electronics and Renewable Systems (ICEARS), IEEE, 2025, pp. 38–45.

[17] P. Kumar, K. Kumar, A. K. Bohre, and N. Adhikary, “Estimating Wind Turbine’s Pitch Angle for Smart Control in hybrid energy network under Random Loading Conditions,” IEEE Trans. Ind. Appl., 2025.

[18] D. Heredero-Peris, M. Capó-Lliteras, D. Montesinos-Miracle, and J. Melendez-Frigola, “Complex order fractional proportional resonant controller for high-frequency applications,” IEEE Open J. Ind. Electron. Soc., 2024.

[19] B. Touqan and A. A. Ameer, “Enhancing HVAC multivariable system performance through hybrid modeling and direct Nyquist Array control,” Build. Serv. Eng. Res. Technol., vol. 45, no. 6, pp. 833–861, 2024.

[20] C. Bissell, “A history of automatic control,” Springer Handb. Autom., pp. 53–69, 2009.

[21] K. J. Åström and R. Murray, Feedback systems: an introduction for scientists and engineers. Princeton university press, 2021.

[22] B. W. Bequette, Process control: modeling, design, and simulation. Prentice Hall Professional, 2003.

[23] O. Mohamed, Z. Bitar, A. Abu-Sultaneh, and W. A. Elhaija, “A simplified virtual power system lab for distance learning and ABET accredited education systems,” Int. J. Electr. Eng. Educ., vol. 60, no. 4, pp. 397–426, 2023.