engineering proceedings

MDPI

Proceeding Paper

Large Language Model-Integrated Teaching Practices in Courses on Python and Automatic Control Principles †

Fangji Zhang, Zhaowei Wang and Lei Fan *

School of Intelligent Manufacturing, Shanghai Zhongqiao Vocational and Technical University, Shanghai 201514, China; zfj965933@163.com (F.Z.); 10508013@163.com (Z.W.)

* Correspondence: 1911565@tongji.edu.cn

† Presented at the 2024 Cross Strait Conference on Social Sciences and Intelligence Management, Shanghai, China, 13–15 December 2024.

Abstract

In the course of studying automatic control for students majoring in Mechatronics and Control Engineering, Python has become the dominant language in artificial intelligence and machine learning as an essential tool for the analysis and design of automatic control systems. In response to the widespread issues of an inadequate ability to apply automatic control principles, an unclear understanding of logical architecture, and a lack of coding abilities in programming for complex systems, we introduce the "Wenxinyiyan" large language models (LLMs) tool. For the height control of the V-22 Osprey tilt-rotor aircraft in helicopter mode, we guided students to develop a control system in a structured question-and-answer learning process and a model-driven approach. This assisted students in establishing a computer-aided design framework for complex systems and enhancing their understanding of control logic. The LLM assisted students in writing high-quality and clean code.

Keywords: large language models; Python; principles of automatic control systems; integrated teaching; teaching effectiveness

check for updates

Academic Editors: Teen-Hang Meen, Jiazhen Huo and Cheng-Fu Yang

Published: 31 July 2025

Citation: Zhang, F.; Wang, Z.; Fan, L. Large Language Model-Integrated Teaching Practices in Courses on Python and Automatic Control Principles. *Eng. Proc.* **2025**, *98*, 43. https://doi.org/10.3390/engproc2025098043

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Automatic control is a fundamental course in many engineering colleges. However, this course requires an understanding of many knowledge points, abstract concepts, and complex, complicated formulas, which demand a higher mathematical level for students. Due to the limitations of students' backgrounds, teacher ability, and class schedule, it is difficult to ensure the teaching quality of this course in the classroom. The principles of automatic control originate from engineering practice and are applied in an increasingly wide range of areas, requiring students to analyze and solve problems using relevant knowledge. This poses higher requirements for teachers and presents challenges to the traditional teaching mode.

2. Traditional Teaching Mode

2.1. Curriculum System of Principles of Automatic Control

The review results of textbooks on "Principles of automatic control" [1–3] present the current focus of the curriculum system, which largely follows the knowledge system outlined in Figure 1. Guided by the performance indicators of automatic control systems, the course offers the basic knowledge of the control systems model, time-domain analysis,

root locus method, and frequency-domain analysis to identify the performance indicators and assess models. It also teaches students the analysis of discrete systems and nonlinear systems using these methods.

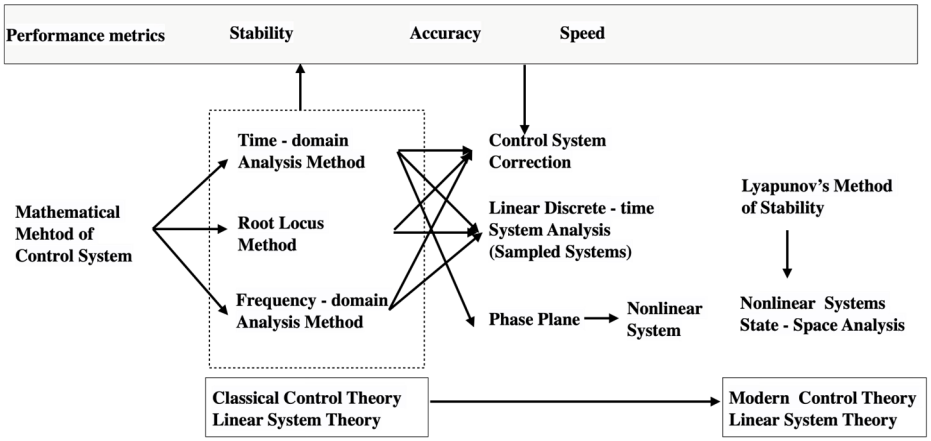


Figure 1. Curriculum system of "Principles of automatic control".

2.2. Course Characteristics

Based on the above curriculum system, the course for principles of automatic control possesses the following characteristics.

1. The prerequisite courses are theoretical, extensive in content, and challenging in difficulty. Besides mathematical courses such as advanced mathematics, complex variable functions, and linear algebra, the prerequisite courses include college physics, thermodynamics, circuit theory, and electronics technology. From the perspective of application, establishing mathematical models of application systems requires fundamental knowledge across various disciplines. This course has numerous knowledge points and obscure content. Based on the feedback from students and teachers, it is difficult to ensure teaching quality solely through classroom instruction.
2. The course is overly abstract, with inadequate practical components. Although the course originates from practical applications and aims to contribute back to them, many existing textbooks do not adequately address the integration of theory with practice. The majority of experiments are focused on theoretical verification, which directly affects the cultivation of students' engineering practice abilities.
3. Currently, the principles of the automatic control course offered by various universities are primarily credited with 3, 3.5, or 4 credits. Due to the heavy teaching load and primary reliance on classroom instruction, there is a low intersection with other disciplines.
4. The utilization of informatization tools is low. Although Python (3.12.5) and MATLAB (R2024a) support computations and analyses for applications in principles of automatic control, current teaching primarily focuses on introducing MATLAB, with its applications also emphasizing theoretical verification-based practices. MATLAB, as a form of commercial software, has many limitations in its application. Therefore, from the perspective of learning or future knowledge expansion, there is a need to strengthen supportive practical training for students in using Python for principles of automatic control. Students must be proficient in using Python for modeling, time-domain analysis, frequency-domain analysis, and stability analysis in automatic control systems. This lays the foundation for future applications of principles of automatic control in interpreting real-world problems.

3. Teaching Design Based on Large Language Models

3.1. Teaching Design

Based on the student feedback on the learning outcomes of the course, the following implementation model was constructed (Figure 2).

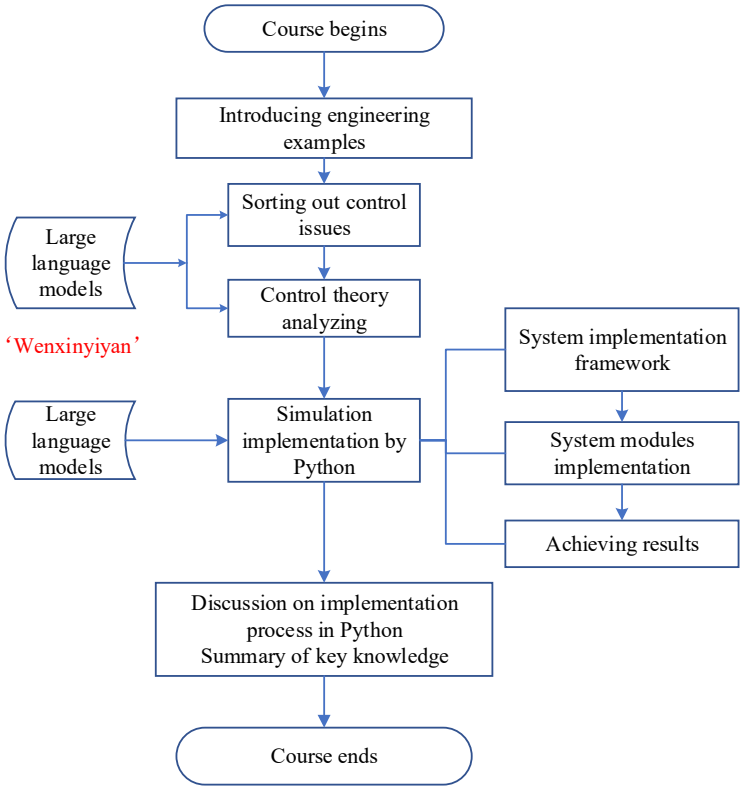


Figure 2. Teaching model based on real-world cases supported by large language models.

Currently, student feedback highlights that one of the difficulties in this course lies in sorting out and modeling practical systems from the perspective of automatic control implementation dimensions, as well as conducting time-domain, stability, and frequency-domain analyses of these models. In addition, students are tested for the informatization modeling of complex systems, mastering the ideas and frameworks for building the informatization implementation of complex systems, developing model libraries, and solving key problems rather than being tested for solving practical problems. Moreover, proficiency in applying mathematical models, constructing complex system frameworks, and writing high-quality code are important professional competencies [4,5]. After evaluating the currently popular large language models (LLMs) in China, we selected "Wenxinyiyan" for the teaching practice. This model is also used for studying control theory problems and implementing complex systems in Python.

3.2. Role of LLMs

In teaching the course, the "Wenxinyiyan" LLM plays a functional role in both the learning of the principles of automatic control and Python system implementation. In learning automatic control principles, "Wenxinyiyan" assists students in completing the learning of the following contents:

1. The open-loop transfer function and closed-loop transfer function, as well as the formula for the closed-loop transfer function;
2. A solution for the system in the time domain and frequency domain;
3. The stability analysis of the system.

In the process of exploring these topics, the guidance function of "Wenxinyiyan" enabled an in-depth study of the relevant content in the principles of automatic control. This type of learning is characterized by its systematic nature, forward-looking perspective, and completeness. Furthermore, due to the interactive nature of "Wenxinyiyan", the learning experience did not become monotonous. From the perspective of system implementation in Python, "Wenxinyiyan" achieved the following functionalities:

1. Mastering the overall framework implementation of the system;
2. Grasping the programming of system transfer function models using Python libraries for the principles of automatic control;
3. Mastering multiple methods for performing time-domain analysis, frequency-domain analysis, and stability analysis in Python systems.

Through the multi-information source analysis of "Wenxinyiyan", students grasped the writing of Python's source code and the coding of key system components and understood the functional differences and usage scenarios between Python and MATLAB. The use of "Wenxinyiyan" also enabled a preliminary understanding of simulation analysis in the context of the principles of automatic control.

4. Case Design and Study

4.1. Case Design

To enhance teaching quality and student's abilities to connect theory with practice, cases were selected based on the following principles [1,6,7]:

1. Moderate difficulty, capable of covering most of the key points in control theory;
2. Close to reality and being able to resonate with students;
3. Strong scalability, helping students further expand their related knowledge.

Based on these principles, the V-22 Osprey tilt-rotor aircraft height control system in helicopter mode was selected for teaching. The control block diagram of the system is shown in Figure 3.

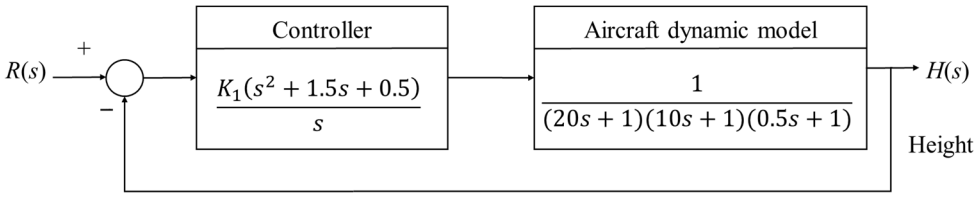


Figure 3. V-22 osprey tilt-rotor aircraft height control system structure diagram in helicopter mode [1].

4.2. Case Study

With the support of "Wenxinyiyan", the students conducted theoretical analysis and calculations on the system based on the principles of automatic control theory.

1. The open-loop transfer function of the system is given below.

$$G(s) = \frac{K_1(s^2+1.5s+0.5)}{(20s+1)(10s+1)(0.5s+1)} = \frac{K_1(s+0.5)(s+1)/100}{(s+0.05)(s+0.1)(s+2)} \quad (1)$$

2. The gain of the system root locus is given below as K_2.

$$K_2 = K_1/100 \quad (2)$$

3. With the assistance and guidance provided by the LLMs, one can further derive the closed-loop transfer function and its characteristic equation.

4. The closed-loop transfer function is given below.

$$\phi(s) = \frac{G(s)}{1+G(s)} = \frac{K_1(s+0.5)(s+1)/100}{K_1(s+0.5)(s+1)+(s+0.05)(s+0.1)(s+2)} \quad (3)$$

5. The closed-loop characteristic equation of the system is given below.

$$s^4 + 2.15s^3 + \left(0.305 + K^*\right)s^2 + \left(0.01 + 1.5K^*\right)s + 0.5K^* = 0 \quad (4)$$

To cultivate holistic perspectives, students were guided to learn the basic ideas and goals of system architecture design through LLMs, namely, achieving software reuse through repeated architectural patterns. Starting from the actual situation and based on the drawing of the call/return architecture principle, the system was designed using a main program/subroutine style and a hierarchical architecture style for this particular design [5] (Figure 4).

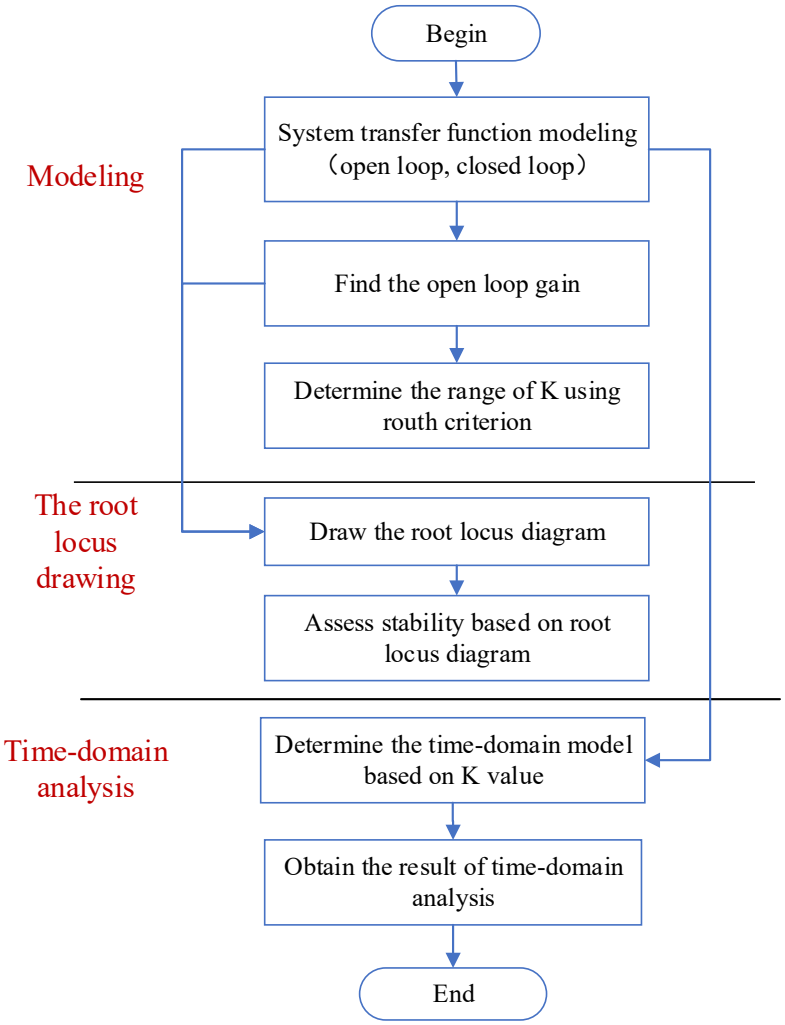


Figure 4. Implementation flowchart for V-22 altitude control system.

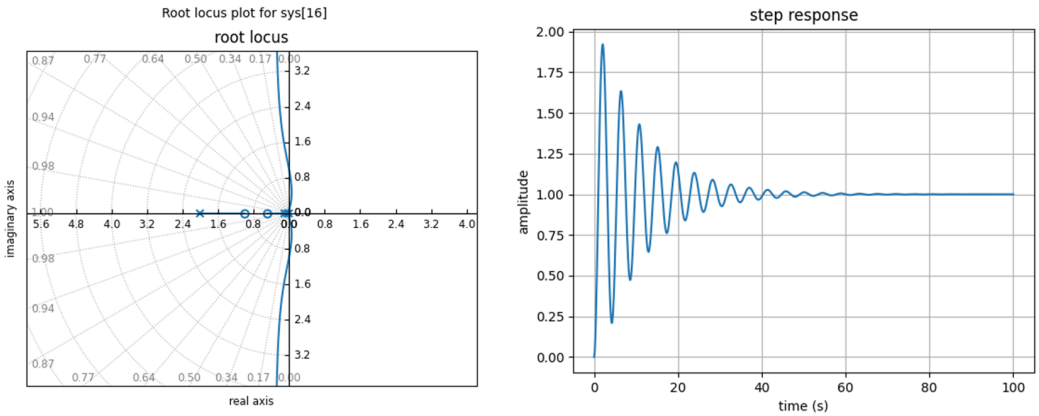
Based on the above architecture, the system was developed modularly according to functional modules, such as system modeling, root locus plotting, and time-domain analysis. Relevant resource libraries were also established to facilitate system reuse. Based on the system architecture design, interactions with the LLMs were conducted by focusing on modeling automatic control systems, time-domain solution seeking, and root locus plotting. Learning the corresponding Python implementation methods [8] was carried out

on this basis. An example of the implementation of transfer functions is given as follows (Algorithm 1).

Algorithm 1. Typical code: Transfer function generation code

```
import numpy as np
from collections import deque
import control as ctrl
def LoopG(num_set,den_set):
    for i in range(len(num_set)):
        print('len-',len(num_set))
        print(len(num_set))
        if len(num_set) == 1:
            num_LoopG = list(num_set)
            print("num_LoopG=",num_LoopG)
        else:
            if  i == 0:
                num_LoopG = np.polymul(num_set[i],num_set[i+1])
                continue
            if  i == (len(num_set) -1):
                break
        num_LoopG = np.polymul(num_LoopG,num_set[i+1])
    for j in range(len(den_set)):
        if len(den_set) == 1:
            den_LoopG = np.polymul(den_set[0],[1])
        else:
                if  j == 0:
                    den_LoopG = np.polymul(den_set[j],den_set[j+1])
                    continue
                if  j == (len(den_set) -1):
                    break
                den_LoopG = np.polymul(den_LoopG,den_set[j+1])
        Open_LoopG=ctrl.TransferFunction((num_LoopG),(den_LoopG))
        Func_LoopG = ctrl.feedback(Open_LoopG, 1)
        return(Open_LoopG,Func_LoopG)
```

The results of this example are shown in Figure 5.



Result of root locus analysis Result of time-domain response analysis

Figure 5. Example execution results.

The students were guided to expand their knowledge in at least two areas through the knowledge extension and guidance function of the LLMs. On the one hand, during the course, learning the principles of automatic control was enhanced, especially in the frequency-domain analysis of automatic control systems, analysis related to proportional-integral-derivative (PID) controllers, and learning other control algorithms in automatic control principles. On the other hand, students were guided to learn to develop automatic control systems using Python, including multiple attempts in the development of transfer functions, implementing stability analysis, frequency-domain analysis, and various graphical representations, and expanding their learning to include the Python implementation of PID.

5. Conclusions

We integrated Python with the purpose of achieving the principles of automatic control, supported by "Wenxinyiyan". Students learned automatic control systems based on Python by selecting real-world systems. The innovation of this teaching method was to stimulate student interest in learning the theoretical knowledge of automatic control. Through the integration of Python with automatic control systems, students could understand a complementary relationship between traditional theoretical knowledge and practical teaching based on modern information technology, enabling students to understand the idea that "the superior man has no other advantage but his ability to make good use of things", seek resources, and use tools to solve problems. By introducing the "LLMs", education was integrated with entertainment to mobilize students' subjective initiative in learning, cultivate their self-learning abilities, and achieve remarkable results with half the effort.

Student classroom learning efficiency was effectively improved with the proposed teaching method in this study. Additionally, due to the case designs being close to reality and interdisciplinary, students increased their after-school proactive learning time, and the number of students seeking help increased. This enhanced the direct interaction between students and knowledge through the "LLMs", and the interaction between students and teachers increased.

Author Contributions: Conceptualization, F.Z. and Z.W.; methodology, F.Z.; software, F.Z.; validation, Z.W. and L.F.; formal analysis, F.Z.; investigation, Z.W.; resources, F.Z.; data curation, L.F.; writing—original draft preparation, F.Z.; writing—review and editing, L.F.; visualization, F.Z.; supervision, F.Z.; project administration, F.Z.; funding acquisition, F.Z. and L.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work was sponsored by the Natural Science Foundation of Shanghai, Zhongqiao Vocational and Technical University (No. ZQZR202420).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting this study are not publicly available due to [legal/ethical/commercial] restrictions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hu, S.; Jiang, B.; Zhang, S. *Principles of Automatic Control*, 8th ed.; Science Press: Beijing, China, 2023.
2. Wang, W. *Principles of Automatic Control*, 3rd ed.; Higher Education Press: Beijing, China, 2020.
3. Zou, J. *Principles of Automatic Control*, 1st ed.; Machine Press: Beijing, China, 2020.

4. Cao, Y.; Liu, Y.; Tian, K. The teaching design of the basic of control theory based on Matlab. *Educ. Teach. For.* **2020**, *32*, 248–250. (In Chinese)
5. Xia, H.; Xu, S. Automatic control principle classroom teaching based on MATLAB. *China Educ. Technol. Equip.* **2023**, *14*, 49–52. (In Chinese)
6. Xu, L.; Ye, Y.; Chen, G. Research on teaching stability analysis based on MATLAB and automatic control principles. *Technol. Wind.* **2021**, *28*, 102–104.
7. Lei, X. The exploration of automatic control based on UAV. *J. EEE* **2022**, *4*, 84–86.
8. Dong, F. *Python Programming Fundamentals and Applications*; China Machine Press: Beijing, China, 2021.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.