Teaching reform of "Signal Processing" course based on Python

Kurban Ubul [1, a], Mayira Yibulayin [1, b], Shuai Yuan [1, c], and Alimjan Aysa [1, d]

[1]School of Computer Science and Technology , Xinjiang University;

[a] kurbanu@xju.edu.cn, [b] 997108771@qq.com, [c] 739343917@qq.com, [d] alim@xju.edu.cn

Abstract. In electronic information majors, "Signals and Systems" and "Digital Signal Processing" are core foundational courses that bridge previous courses and advanced courses, it integrate knowledge from circuit analysis, advanced mathematics, and complex functions into essential courses like communication principles and digital signal processing, with a lot of knowledge,wide coverage,and strong theoretical characteristics.This paper proposes a teaching reform of the "Signal and Systems" course based on Python,introducing the language of Python in teaching can enable students to further understand the course of Signals and Systems, which can improve students' interest in learning and thus increase their learning motivation.

Keywords: Python; signal and system; teaching reform.

1. Introduction

The course "Signal and System" is a fundamental requirement for electronic information majors, tasked with studying the essential theory and analysis methods of signals and linear time-invariant systems. It necessitates a comprehensive understanding of signal transformation theory and analytical techniques for linear time-invariant systems,for subsequent professional courses, such as "digital signal processing", "Communication principle" plays the role of connecting the previous and the next[1].The teaching process of this course requires not only the imparting of theoretical knowledge, formula derivation, and class assignments, but also the inclusion of classroom demonstrations and practical exercises to facilitate students' comprehension and mastery of the taught theory.Nowadays, Matlab is widely employed by most colleges and universities as a supplementary tool to enhance teaching effectiveness. This is achieved through the incorporation of classroom demonstrations, course experiments, and curriculum design, which not only enriches the content of courses but also enhances the flexibility and interactivity of instructional methods, ultimately leading to favorable educational outcomes.

Matlab is capable of performing matrix operations, drawing functions and data, implementing algorithms, creating user interfaces, connecting programs of other programming languages, etc. It is primarily utilized in engineering calculations, control design, signal processing and communication, image processing, signal detection, financial modeling design and analysis as well as other fields. In particular Matlab with signal processing package, we can easily make signals and systems analysis of numerical computation, visualization and modeling and system design, simulation and debugging, etc[2]. It enables students to focus on understanding and applying basic concepts and principles taught in the classroom without having to manually calculate numerous mathematical formulas like before. The Matlab, however, there are great limitations:

First, the cost, as a commercial software, Matlab is not free, and the price for obtaining a legitimate license is substantial, which is a considerable expense for ordinary students. Even for universities that have obtained authorization, due to the many toolboxes in Matlab, there is a significant investment in hardware and software required to establish a Matlab environment for multiple experiments. Although many people use the public free version of Matlab it differs significantly from the legitimate version in terms of function and performance making it unsuitable for teaching "Signal and System" due to its high cost.

Second, while using Matlab may be faster than traditional programming languages C/C++ when solving technical calculation problems it still requires considerable effort to re-learn how to use it thus making getting started time-consuming and labor-intensive. And the program written by MATLAB only can be run on MATLAB platform, and it has not the ability to cross-platform, so that portable is poor[3].

Third, most extension libraries used by Matlab are not open source meaning students can only call them during usage but do not know how they work thereby failing to achieve expected teaching goals;

Fourth, The original intention of Matlab is to facilitate the modeling of scientific computing problems, which does not align with the fundamental principles and applications emphasized in the "Signal and Systems" course content.

Additionally, in practice, Matlab primarily caters to high-level researchers and has a relatively limited scope of application in industry, making it less appealing to undergraduate students who aspire to acquire practical skills that are directly applicable in their future work endeavors. Consequently, students perceive Matlab as a transient tool that holds no relevance beyond the duration of the course. This lack of enthusiasm among students towards learning Matlab also adversely impacts the overall teaching quality of the course.

All these aforementioned issues represent tangible challenges within our teaching process that necessitate immediate resolution for enhancing the instructional efficacy of "Signal and Systems" In recent years, given Python's rapid growth and development as a programming language, employing Python as an alternative to Matlab and establishing a "Signal and Systems" teaching framework based on Python have emerged as viable solutions for addressing these concerns.

2. Advantages of Python for teaching

Python is an interpreted high-level programming language that supports scripting, process-oriented and object-oriented programming paradigms. It boasts concise code, simple syntax, and excellent scalability. Python's versatility provides the same level of interactivity and flexibility as Matlab while having purer object-oriented features than Matlab. As a cross-platform programming language, PYTHON has been ported to many platforms, represented by Linux, Windows, Mac and Android[4]. Its source code can be easily embedded into C/C++ programs, and third-party libraries written in C/C++ can also be directly invoked by Python programs. In contrast to Matlab's limitations in this regard. Moreover, Python offers a much richer scientific computing ecosystem compared to Matlab with a cleaner and easier learning curve.

Python also has a very rich library related to data science, which is widely used in engineering, data analysis, and artificial intelligence[5]. For signal processing and systems analysis tasks specifically, numpy, scipy, and matplotlib are widely used for analyzing signals in both time and frequency domains along with filtering operations and graph visualization.Numpy serves as the fundamental library for high-performance scientific computing and data analysis, serving as a cornerstone for numerous high-level libraries such as Scipy and Pandas. Numpy introduces an efficient multidimensional array (ndarray) with vector arithmetic capabilities, along with a comprehensive set of standard mathematical functions applicable to entire datasets. This results in concise and expedient code execution without necessitating iterative loops. For instance, as depicted in Figure 1, scipy is a specialized Python toolkit built upon numpy that offers convenient and user-friendly functionalities for scientific and engineering purposes. It encompasses modules for statistics, optimization, integration, linear algebra, Fourier transform, signal and image processing techniques, filter design, and other typical methods related to signals and systems. Matplotlib serves as a fundamental Python library for plotting data; numerous third-party visualization libraries are developed on top of it. With this library at their disposal, Python users can effortlessly generate line charts, bar charts, etc., while also being able to customize labels and legends or resize plots akin to those created using Excel.The Python standard library provides wave and pyaudio modules, as demonstrated in FIG. 2, for efficient processing of digital signals such as speech and images that are widely used in practical applications. Additionally, there are numerous third-party libraries available for various transform domain analysis and processing tasks related to wav format audio files. When combined with the Python standard library, these libraries collectively occupy less than 200MB of storage space, which is significantly smaller compared to Matlab's gigabyte-sized software.

Therefore, the incorporation of Python into the instruction of Signals and Systems, along with the teaching reform scheme centered on Python-based system design, offers several advantages:

It reduces both institutional and individual expenses associated with creating experimental environments.

The simplicity of programming and the extensive open-source community provide beginners with a lower learning threshold and abundant educational resources.

Open-source third-party function repositories offer robust support for students to comprehend course content and apply their knowledge to solve practical problems.

The wide-ranging application prospects serve as a motivational factor for students to learn Python thoroughly and engage in profound research as well as project development.

This paper implements a teaching reform approach based on Python for Signal and System courses, which has been practically tested to assess its effectiveness.

```
In [1]: import numpy as np  #导入numpy库
        arr = np.array([1,2,3,4,5,6]) #定义ndarray
        arr * 6
Out[1]: array([ 6, 12, 18, 24, 30, 36])

In [2]: arr+arr
Out[2]: array([ 2,  4,  6,  8, 10, 12])

In [3]: arr.std() #求标准差
Out[3]: 1.707825127659933
```

Figure 1: numpy library demo

```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(-20, 20, 500)
print(x)
y = np.cos(x)
plt.plot(x, y)
plt.show()
```

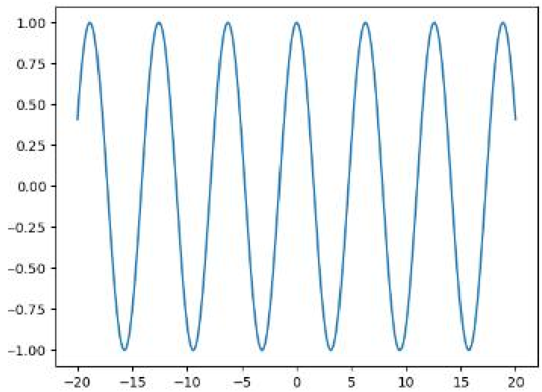

Figure 2: Matplotlib library demo

3. Thoughts and plans for educational reform

Currently, the teaching content of Signal and System in most colleges is primarily divided into two components: classroom instruction and independent study. Classroom instruction consists of theoretical teaching and experimental demonstrations, while independent study involves completing assigned exercises outside of class. During classroom instruction, the teacher first explains fundamental concepts and formulas, interspersed with practical experiments to students take the initiative to think and learn, teachers are responsible for guiding answers, and create practice opportunities for students, so that students can freely exert their imagination. The selection of after-class exercises is provided by the teacher, which students complete independently and submit as a course design report. The grade for this report contributes proportionally to the final grade. Additionally, there is a corresponding course interface on the school website that allows teachers to upload course materials and assignments for students to download and submit their homework electronically. An online Q&A platform also facilitates mutual communication among students, creating a vibrant independent learning environment.

The classroom instruction is primarily divided into three sections. Firstly, the substitute teacher provides an overview of the key knowledge points in this section of the course, enabling students to gain a comprehensive understanding of the content. Secondly, each knowledge point is explained in detail, accompanied by deducing key formulas and live demonstrations using Python. Due to its simple and concise syntax as well as open-source third-party libraries, Python facilitates students' comprehension and application of learned concepts into practical scenarios. Lastly, there is a class summary and homework assignment session where the teacher recaps the acquired knowledge and addresses any questions from students on-the-spot before assigning homework.

Taking the chapter on frequency domain analysis of continuous-time systems as an example, teaching can be conducted through these three stages during the Fourier transform section. In stage one, a brief introduction covers signal decomposition along with definitions, principles, and properties of Fourier transform and inverse Fourier transform. Stage two provides a detailed explanation of the principles behind Fourier transform and inverse Fourier transform, presenting the derivation on-site and thoroughly discussing the properties of Fourier transform. Subsequently, Python code is implemented based on the defined formula for Fourier transform to perform signal analysis. Taking a specific function as an example, both the function itself and its corresponding image are illustrated in Figure 3.

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty}(a_n \cos(n\Omega t) + b_n \sin(n\Omega t))$$

$$a_n = \frac{2}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)\cos(n\Omega t)dt, n = 0,1,2,\cdots$$

$$b_n = \frac{2}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)\sin(n\Omega t)dt, n = 0,1,2,\cdots$$

（Eq. 1）

$$f(x) = \begin{cases} e^x, -\pi \le x < 0 \\ 1, 0 \le x < \pi \end{cases}$$

（Eq. 2）

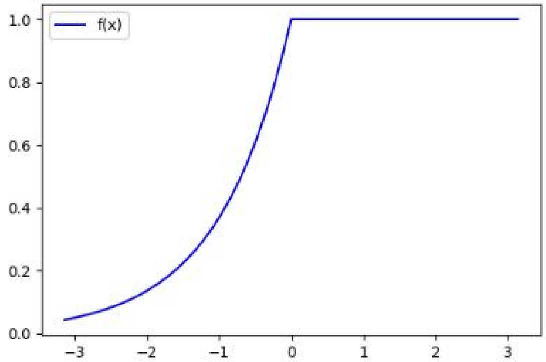

Figure 3 f(x) function and image

The Python code for computing the Fourier series expansion (refer to Eq. 1) is presented below, with the resulting graph depicted in Figure 4.

```
from pylab import *
x = mgrid[-10:10:0.02]
n = arange(1, 1000)
def fourier_transform():     # definition of Fourier Transform
        a0 = (1-exp(-pi))/pi+1
        s = a0/2
        for i in range(1, 100, 1):
                s0= ((1-(-1)**i*exp(-pi))/(pi*(1+i**2))*cos(i*x)+1/pi*((-i*(1-(-1)**i*
exp(-pi)))/(1+i**2) + (1-(-1)**i)/i)*sin(i*x))
                s = s+s0
                plot(x, s, 'orange', linewidth=0.6)
                title('fourier_transform')
                show()
fourier_transform()
```

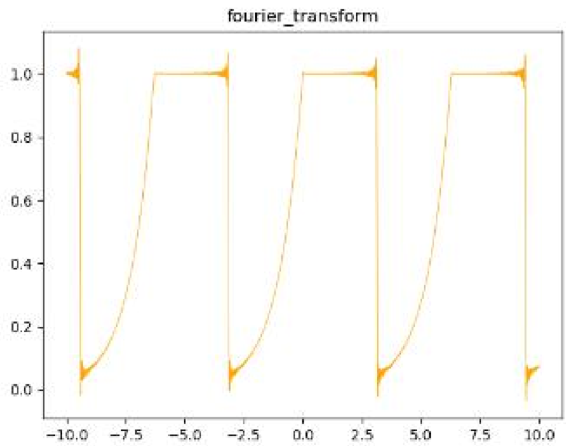

Figure 4 Fourier transform result

The third stage primarily involves summarizing the teaching content, assigning theoretical and computer homework, as well as providing on-site counseling and answering questions. The experimental course is conducted through centralized arrangement, in-class guidance, and independent completion, organized into three stages. Additionally, Python experimental textbooks are provided for students' reference.

① Task assignment where teachers introduce and allocate experimental tasks along with necessary materials and references.

② Experiment implementation where students independently write Python code to conduct experiments based on assigned topics, record experimental data, and analyze results. Due to its small size and simple installation process, Python software greatly facilitates the creation of an experimental environment for students. Moreover, its syntax similarity to C/C++ makes it easier to grasp compared to Matlab; thus reducing the learning threshold while facilitating student learning and usage.

③ Summary of Experiments where students organize their findings into experiment reports which are then reviewed by the teacher.

This course consists of five experiments including discrete signal time-frequency analysis, discrete system time-frequency analysis,digital filter design,digital signal filtering processing,and speech signal digital filtering.Students are required to program according to the instructions provided in the experimental textbooks while also comparing their work with existing Python library functions in order to deepen their understanding of what they have learned.

4. Conclusion

Python is the most widely used computer language at this stage, with the advantages of powerful, intuitive, fast operation and so on., within the curriculum of "Signals and Systems" enables instructors to not only save time but also enhance students' classroom learning efficiency. Moreover, it significantly aids students by fostering their interest in the subject matter and facilitating the development of active learning skills as well as analytical problem-solving abilities.The Python language analyzes various signal characteristics from deep to shallow, employing relevant formulas to address inquiries, and ultimately presents them graphically. This approach enables the intuitive and simplified visualization of complex concepts, thereby exemplifying the efficiency and convenience of "Signals and Systems".

5. Acknowledgment

This study was supported by the Teaching Reform Project of Postgraduate Education in Xinjiang Uygur Autonomous Region, China (No. XJ2022GY09) and the Xinjiang University Postgraduate Excellence Course on Information Theory and Coding (No. XJDX2022YJPK19).

References

[1] B. Han, C. Zhang and X. Qin, "Based on Matlab signals and systems course project-driven teaching method research," 2011 IEEE 3rd International Conference on Communication Software and Networks,Xi'an,China,2011,pp.466-469,doi:10.1109/ICCSN.2011.6013873.

[2] Luhe Hong and Jianli Cai, "The application guide of mixed programming between MATLAB and other programming languages," 2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE), Singapore, 2010, pp. 185-189, doi: 10.1109/ICCAE.2010.5452058.

[3] Y. Huang, J. Zhao, Y. Qiang, T. Hou, X. Ren and H. Sun, "The Reform and Exploration of Intelligent PYTHON Language Teaching," 2021 16th International Conference on Computer Science & Education (ICCSE), Lancaster, United Kingdom, 2021,pp.883-888,doi:10.1109/ICCSE51940.2021.9569670.

[4] H. Zhang, P. Yang and Y. Niu, "Application of Python Scientific computing library and Simulation in Circuit Analysis," 2023 IEEE 12th International Conference on Communication Systems and Network Technologies (CSNT), Bhopal, India, 2023, pp. 892-898, doi: 10.1109/CSNT57126.2023.10134600.

[5] X. Zhang, W. Li and G. Wang, "Construction and Application of the Project-based Teaching System of Statistical Experiment Course using Python Language," 2023 5th International Conference on Computer Science and Technologies in Education (CSTE), Xi'an, China, 2023, pp. 15-19, doi: 10.1109/CSTE59648.2023.00010.