PriMera Scientific Engineering
Volume 7 Issue 4 October 2025
DOI: 10.56831/PSEN-07-226
ISSN: 2834-2550

PriMera Scientific Publications

# Study of Computer Programming Majoring in Software Engineering using Matlab to Python Programming Language at University

**Type:** Research Article
**Received:** April 09, 2025
**Published:** September 29, 2025

**Citation:**
Trinh Quang Minh., et al. "Study of Computer Programming Majoring in Software Engineering using Matlab to Python Programming Language at University". PriMera Scientific Engineering 7.4 (2025): 29-43.



**Trinh Quang Minh*, Trinh Hue, Ngo Thi Lan, Bui Xuan Tung, Nguyen Chi Cuong, Lam Tan Phuong, Tran Minh Tan and Nguyen Minh Hieu**

*Faculty of Engineering - Technology, Tay Do University, Can Tho City, Viet Nam*

***Corresponding Author:** Trinh Quang Minh, Faculty of Engineering - Technology, Tay Do University, Can Tho City, Viet Nam.

## Abstract

Software Engineering is a broad and diverse field that encompasses various aspects of software development, maintenance, and management. Software development with software development lifecycle (SDLC) and software design and architecture. Software project management and software development projects from planning to implementation with data science and machine learning with data processing and analysis and machine learning algorithms to design and develop complex software systems. Software testing and quality assurance. Protecting systems and data from security threats. Data Scientist/Analyst with the task of analyzing data and developing machine learning models. Managing and optimizing the software development and deployment process. Helping to process and analyze data effectively, enabling the development of machine learning and artificial intelligence models. Ensure that software is developed as required, on time, within budget and with high quality. Use technical methods and tools to manage and optimize the software development process. Not too focused on using tools, techniques and technologies to develop and deploy software like the software engineering industry. Not required to always use the latest technologies to develop software efficiently and quickly like software engineering. And the software engineering industry must focus on using databases and database management systems in each future development project. Software engineering focuses on using tools and technologies such as the latest specific databases to develop software efficiently and quickly. And the Software Engineering industry focuses on the technical process and methods (using different techniques, different databases as long as they solve the problem of the project being encountered) to develop software in a systematic and organized manner.

***Keywords:*** software engineering; software development process; data science; machine learning and artificial intelligence models

## Abbreviations

DSP - Digital Signal Processing.
DFT - Discrete Fourier Transform.
FFT - Fast Fourier Transform.
HTTP - HyperText Transfer Protocol.
OOP - Object-Oriented Programming.
API - Application Programming Interface.
JSON - JavaScript Object Notation.
ZT - Zero Tolerance.

## Introduction

Digital Signal and Processing is to provide students with knowledge of Digital Signal Processing. The course focuses on DSP theory and practical applications of DSP (Digital Signal Processing) and the course has the following contents. Master the fundamentals of digital signal and image processing and its many applications and understand the transformations: ZT (Zero Tolerance), DFT (Discrete Fourier Transform), FFT (Fast Fourier Transform). Design digital filters, design digital systems. DSP (Digital Signal Processing) is a technology used to set up different filter positions and avoid noise. Machine Learning Machine Learning helps analyze large and complex data to find patterns and trends, thereby making accurate predictions. For example: sales prediction, weather forecast. Machine learning is used to develop natural language processing applications such as automatic translation, sentiment analysis, and chatbots. Machine learning helps computers recognize and classify images, recognize faces, and detect objects in videos. Recommendation systems use machine learning to suggest products, movies, or songs based on user preferences and behavior. Examples: Netflix, Amazon. Machine learning helps detect and prevent cyberattacks by analyzing unusual behavioral patterns. Machine learning is used to diagnose diseases, analyze medical images, and predict treatment outcomes. The Advanced Project Management course equips you with the skills and knowledge needed to effectively manage large-scale, complex projects. Define the project's goals, scope, and requirements. Plan the project's timeline, resources, and budget in detail and identify, assess, and plan for potential risks. Using tools and techniques to mitigate risks and manage quality is to ensure that the products and services of the project meet the established quality standards and use quality testing and assessment methods. People management with the task of building and managing effective project teams and developing leadership and communication skills. Change management with the task of handling change requests in the project flexibly and effectively and ensuring that changes do not negatively affect the progress and quality of the project. Using project management software and tools such as Microsoft Project, Jira, Trello. Applying Agile, Scrum, and Kanban project management methods. Data Science and Big Data Analytics with methods of collecting data from various sources. Storing big data using systems such as Hadoop, HDFS, and NoSQL databases such as MongoDB, Cassandra. Data Processing and Cleaning with data cleaning and preprocessing techniques to remove missing values, noisy data. Using tools like Pandas, Spark to process big data. Data analysis and visualization from using tools and libraries like Matplotlib, Seaborn, Tableau to visualize data. Data analysis to find patterns and trends. Machine learning and artificial intelligence with applying machine learning algorithms to predict and classify data with using libraries like Scikit-learn, TensorFlow, Keras. Big Data Analytics with using tools and platforms like Apache Spark, Apache Flink to analyze big data and distributed and parallel data processing techniques. Data Mining with data mining methods to extract useful information from large data sets and using algorithms like clustering, association rule mining. IoT (Internet of Things) with the development and management of IoT systems, connectivity and communication between devices.

## Materials and Methods

### *Materials*

Textbooks and online resources: python programming and numerical Methods is a tutorial that covers the basics of Python, data structures, functions, object-oriented programming (OOP) and numerical methods, Python standard libraries such as os, sys, math, datetime,... and external libraries such as NumPy, Pandas, Matplotlib for data science... Exception handling (try-except: catch and handle

exceptions, create custom exceptions). Network programming (HTTP requests: using the requests library and API: Working with API, JSON). Data science and machine learning such as the NumPy library for handling arithmetic arrays, the Pandas library for processing and analyzing data, the Matplotlib library and the Seaborn library for plotting and visualizing data, the Scikit-learn library for machine learning algorithms. Professional certificate in computer science for Python programming: coursera partners with over 350 leading universities and companies to deliver flexible, career-relevant online learning to individuals and organizations around the world. Offers a wide range of learning opportunities—from hands-on projects and courses to job-ready certificates and degrees. Lecture notes and slides: professors often provide detailed lecture notes and slides that cover key concepts in Python programming, algorithms, and software engineering principles. Interactive coding platforms: sites like codecademy.com with the motto is master your language with lessons, quizzes, and projects designed for real-life scenarios and learning to code shouldn't be painful, leetcode.com with the motto is Python programming from zero to expert and learn the theory and complete many practice exercises to master your programming skills. And hackerrank.com helps thousands of companies recruit and upskill the next generation of developers, providing interactive coding exercises and challenges to practice Python programming.

***Methods***

*Lectures and tutorials*: regular lectures to introduce theoretical concepts of software engineering. tutorials and lab sessions to practice coding across subjects such as python for engineers, digital signal processing, machine learning, image and video processing, etc.

*Exercises and projects*: individual and group projects to apply Python programming to real-world scenarios based on prior knowledge and hands-on experience. Exercises to reinforce learning and assess understanding across each subject area.

*Exams and tests*: periodic tests and quizzes to test knowledge and problem-solving skills across software engineering disciplines.

*Peer learning and collaboration*: group study sessions and collaborative projects enhance learning through peer interaction in class and seminars from software engineering theses and dissertations.

*Online courses and MOOCs*: enroll in online courses such as those offered by Coursera, edX, or Udacity to supplement your university studies in software engineering.

*Research and development*: participate in research projects that require Python programming for data analysis, machine learning, digital signal processing, image and video processing, etc., or software development in the direction of software engineering.

## Results and Discussion

The audio signal processing system aims to solve the problem of filtering noise in speech, enhancing sound quality, and creating a foundation for higher applications. The main function is to record audio signals, analyze signals in 2 time and frequency domains. Design a 2-3 kHz band-pass digital filter and play back the filtered signal.

Signals in the time domain with the goal of displaying amplitude changes over time to see the signal structure and evaluate signal quality (noise level, clarity). Illustrations use graphs to illustrate waveforms with waveforms displayed on graphs with the X axis being time (seconds) and the Y axis being amplitude. This helps visualize changes in volume, silence, or loud audio passages. Noise analysis with noise determined based on a very small amplitude threshold (eg 0.02). The percentage of amplitude values below the threshold is calculated. If this ratio is too high, the signal may be noisy or unclear. We can judge that if the noise accounts for a large proportion (e.g. more than 50%), the signal is considered unclear. This information is displayed as an annotation on the graph with the meaning that the time domain is very useful for directly observing loud audio passages or silences and helping users better understand the amplitude changes of the signal.

*Frequency domain signal analysis*: the goal is to analyze the frequency components in the audio signal and identify important frequencies, especially in the voice range.

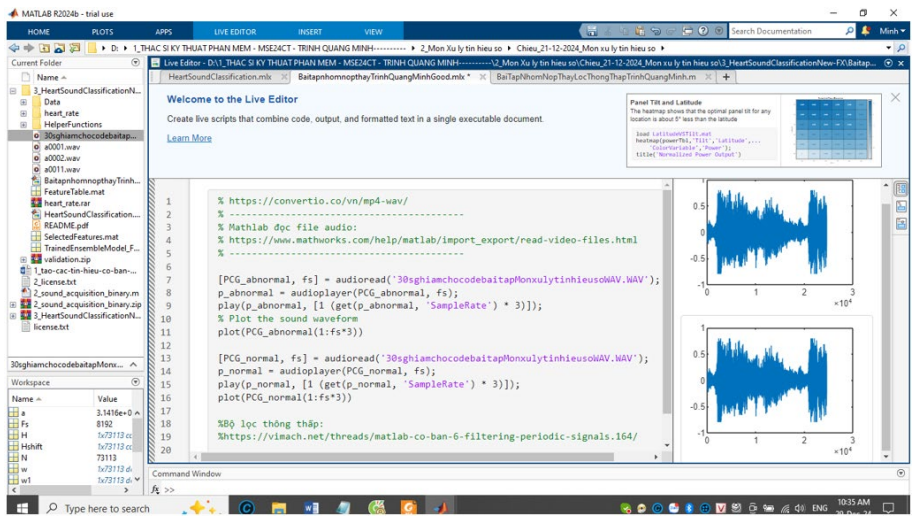
*Fast Fourier Transform (FFT) analysis*: FFT converts the signal from the time domain to the frequency domain. The result is the amplitude values corresponding to each frequency. Low frequencies (below 300 Hz) often contain noise or bass. The voice is mainly in the range of 300 Hz to 3 kHz.

*Frequency components explained*: Frequency components between 300 Hz and 3 kHz are highlighted, as this is the important region of the speech signal. Higher frequencies may represent ambient sounds or noise.

*Frequency spectrum illustration*: The "Magnitude vs. Frequency" plot shows the magnitude of each frequency. A region (300 Hz - 3 kHz) is highlighted on the plot to illustrate the speech region. The significance of the frequency domain is to help identify noise or unwanted components (such as background noise). This analysis is particularly useful in applications such as speech recognition or audio signal processing.

1. **Present the audio signal recorded from the mobile phone using recording software (*.MP4) and convert to WAV and send to the computer for programming:**

```
% https://convertio.co/vn/mp4-wav/

% ----------------------------------------

% Matlab read audio file:

% https://www.mathworks.com/help/matlab/import_export/read-video-files.html

% ----------------------------------------

[PCG_abnormal, fs] = audioread('30sghiamchocodebaitapMonxulytinhieusoWAV.WAV');

p_abnormal = audioplayer(PCG_abnormal, fs);

play(p_abnormal, [1 (get(p_abnormal, 'SampleRate') * 3)]);

% Plot the sound waveform

plot(PCG_abnormal(1:fs*3))

[PCG_normal, fs] = audioread('30sghiamchocodebaitapMonxulytinhieusoWAV.WAV');

p_normal = audioplayer(PCG_normal, fs);

play(p_normal, [1 (get(p_normal, 'SampleRate') * 3)]);

plot(PCG_normal(1:fs*3))

% Low pass filter:

% https://vimach.net/threads/matlab-co-ban-6-filtering-periodic-signals.164/
```




**Convert code from Mathlab to Python with the support of https://copilot.microsoft.com and demonstrate running the code online at** https://www.kaggle.com/code/trnhquangminh140/software-engineering-major-python-programming

*# .\venv\Scripts\activate  # On Windows*

***# pip install simpleaudio***

*# Reinstall simpleaudio: If the module is already installed but still not found, try reinstalling it:*

*# pip uninstall simpleaudio*

*# pip install simpleaudio*

***# First, install the resampy library if you haven't already:***

***# pip install resampy***

import soundfile as sf

import matplotlib.pyplot as plt

import numpy as np

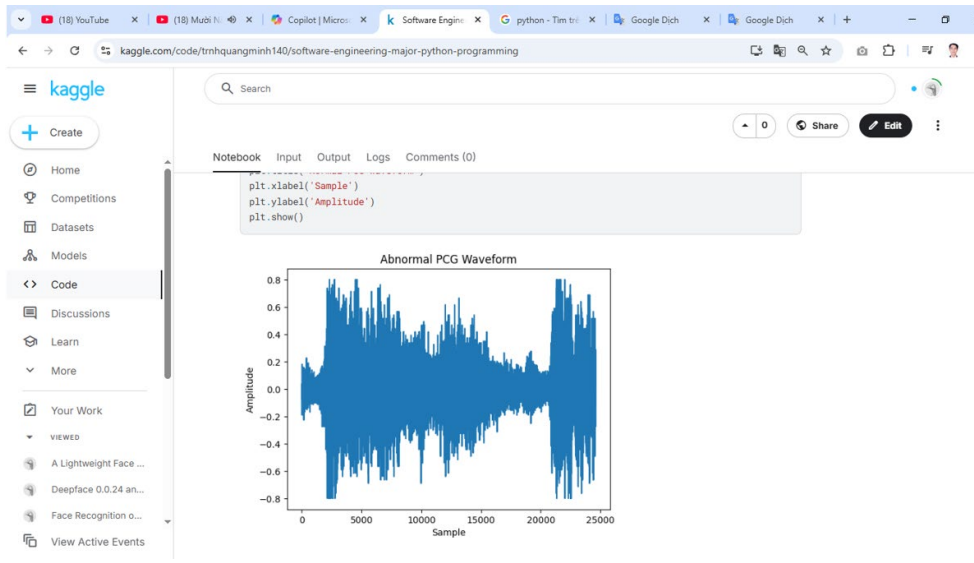
from pydub import AudioSegment

from pydub.playback import play

***# Read the abnormal audio file***

```
PCG_abnormal, fs =
sf.read('/kaggle/input/ghiamwav/30sghiamchocodebaitapMonxulytinhieusoWAV.wav')

# Convert to AudioSegment and play

PCG_abnormal_segment = AudioSegment(

  (PCG_abnormal * 32767).astype(np.int16).tobytes(),

  frame_rate=fs,

  sample_width=2,

  channels=1

)

play(PCG_abnormal_segment)

# Plot the sound waveform for the first 3 seconds

plt.plot(PCG_abnormal[:fs*3])

plt.title('Abnormal PCG Waveform')

plt.xlabel('Sample')

plt.ylabel('Amplitude')

plt.show()

# Read the normal audio file

PCG_normal, fs =
sf.read('/kaggle/input/ghiamwav/30sghiamchocodebaitapMonxulytinhieusoWAV.wav')

# Convert to AudioSegment and play

PCG_normal_segment = AudioSegment(

  (PCG_normal * 32767).astype(np.int16).tobytes(),

  frame_rate=fs,

  sample_width=2,

  channels=1

)

play(PCG_normal_segment)
```

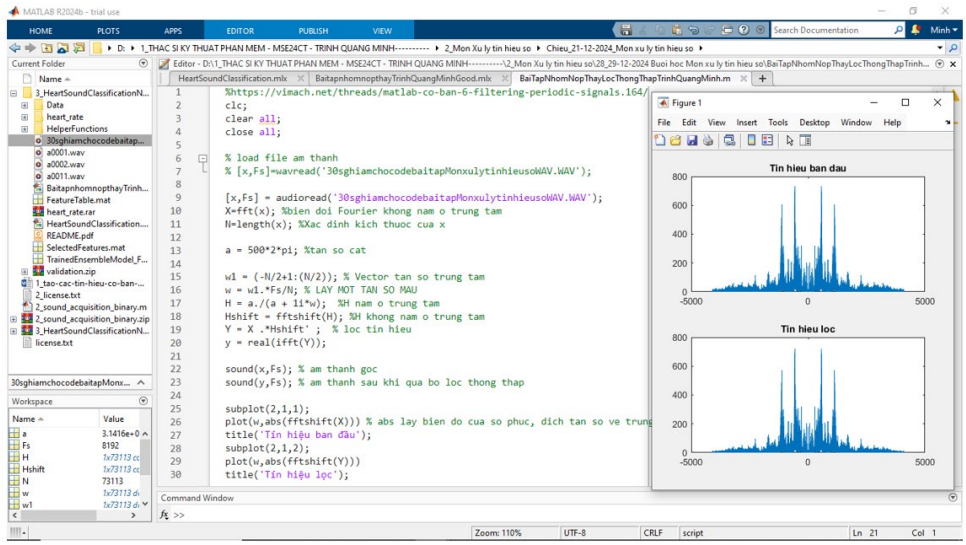
```
# Plot the sound waveform for the first 3 seconds

plt.plot(PCG_normal[:fs*3])

plt.title('Normal PCG Waveform')

plt.xlabel('Sample')

plt.ylabel('Amplitude')

plt.show()
```

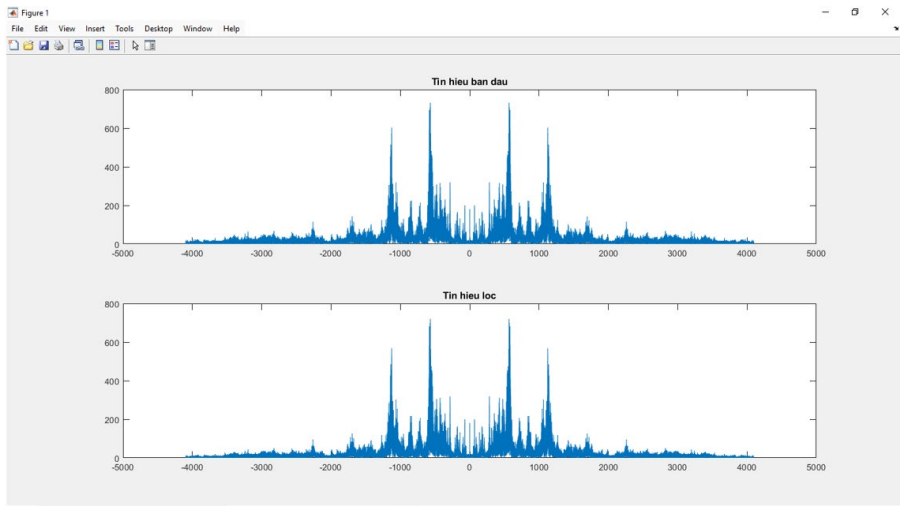



2. **The original signal and the filtered signal are represented in the graph.**

```
% https://vimach.net/threads/matlab-co-ban-6-filtering-periodic-signals.164/

clc;

clear all;

close all;

% load audio file

% [x,Fs]=wavread('30sghiamchocodebaitapMonxulytinhieusoWAV.WAV');

[x,Fs] = audioread('30sghiamchocodebaitapMonxulytinhieusoWAV.WAV');

X=fft(x);                                    % Non-centered Fourier transform
```

```
N=length(x);                                % Determine the size of x

a = 500*2*pi;                               % cut-off frequency

w1 = (-N/2+1:(N/2));                        % Center Frequency Vector

w = w1.*Fs/N;                               % SAMPLING FREQUENCY

H = a./(a + 1i*w);                          % H is in the center

Hshift = fftshift(H);                       % H is not in the center

Y = X .*Hshift' ;                           % Signal filter

y = real(ifft(Y));

sound(x,Fs);                                % Original sound

sound(y,Fs);                                % Of audio after low pass filter

subplot(2,1,1);

plot(w,abs(fftshift(X)))      % abs takes the amplitude of the complex number, shifts the frequency to the
centertitle(' Initial signal ');

subplot(2,1,2);

plot(w,abs(fftshift(Y)))

title(' Filter signal ');
```

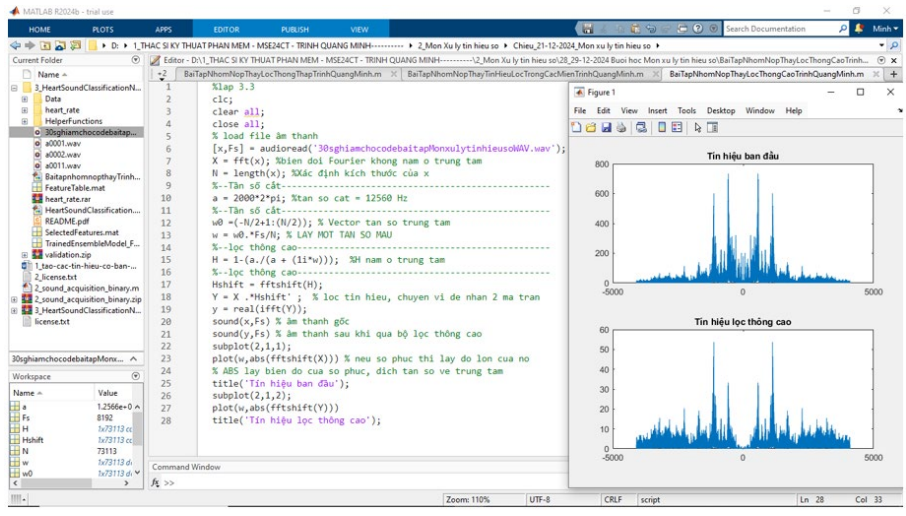


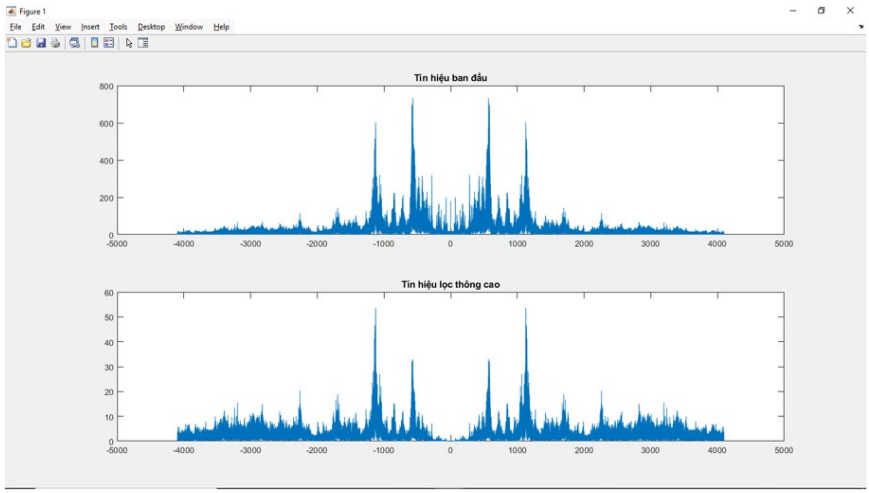



3. **A high pass filter is a filter that only allows signals to operate from its cutoff frequency to infinity. Low frequency signals below the cutoff frequency are filtered out. An ideal high pass filter completely rejects low frequencies.**

```
%lap 3.3

clc;

clear all;

close all;

% load audio file

[x,Fs] = audioread('30sghiamchocodebaitapMonxulytinhieusoWAV.wav');

X = fft(x);                                  % of Fourier transforms are not centered

N = length(x);                               % Determine the size of x

%-- Cutoff frequency ----------------------------------------------

a = 2000*2*pi;                               % Cut-off frequency = 12560 Hz

%-- Cutoff frequency ----------------------------------------------

w0 =(-N/2+1:(N/2));                          % Center Frequency Vector

w = w0.*Fs/N;                                % SAMPLING FREQUENCY

%-- high pass filter ----------------------------------------------
```

```
H = 1-(a./(a + (1i*w)));                                    % H is in the center

%-- high pass filter ----------------------------------------------

Hshift = fftshift(H);

Y = X .*Hshift' ;                                % signal filter, transpose to get 2 matrices

y = real(ifft(Y));

sound(x,Fs)                                                 % original sound

sound(y,Fs)                                                 % of audio after high pass filter

subplot(2,1,1);

plot(w,abs(fftshift(X)))                        % if complex number then take its magnitude

% ABS takes the amplitude of the complex number, shifts the frequency to the center

title(' Initial signal'');

subplot(2,1,2);

plot(w,abs(fftshift(Y)))

title(' High pass filter signal ');
```



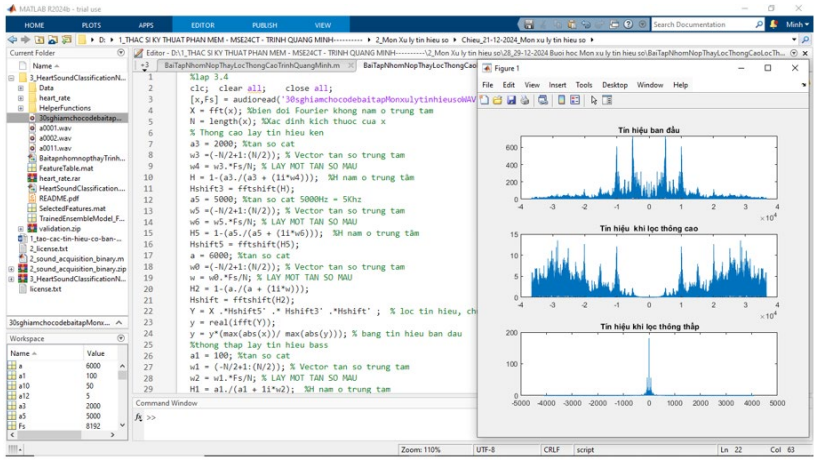


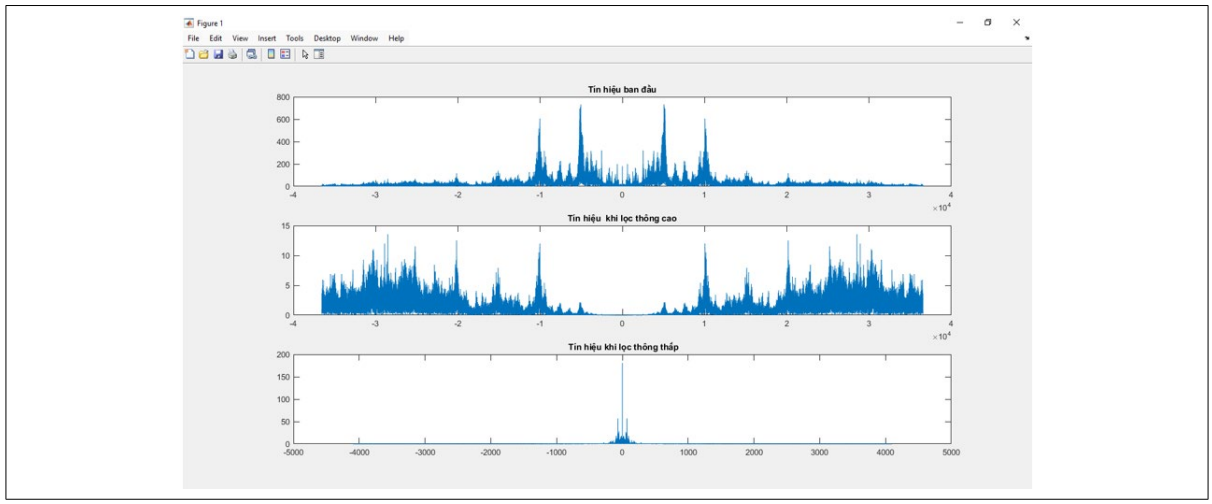

4. **High pass and low pass filtering**

```
clc;  clear all;    close all;

[x,Fs] = audioread('30sghiamchocodebaitapMonxulytinhieusoWAV.wav'); % load audio file

X = fft(x);                                    % of Fourier transforms are not centered

N = length(x);                                 % Determine the size of x

% Horn signal announcement

%----------Cutoff frequency 2000Hz=2KHz----------

a3 = 2000;                                     % cut-off frequency

%----------Cutoff frequency 2000Hz=2KHz----------

w3 =(-N/2+1:(N/2));                            % Center Frequency Vector

w4 = w3.*Fs/N;                                 % SAMPLING FREQUENCY

H = 1-(a3./(a3 + (1i*w4)));                    % H is in the center

Hshift3 = fftshift(H);

%---------- Cutoff frequency 5000Hz=5KHz----------

a5 = 5000;                                     % cut-off frequency 5000Hz = 5Khz

%---------- Cutoff frequency 5000Hz=5KHz----------
```

```
w5 =(-N/2+1:(N/2));                          % Center Frequency Vector

w6 = w5.*Fs/N;                               % SAMPLING FREQUENCY

H5 = 1-(a5./(a5 + (1i*w6)));                 % H is in the center

Hshift5 = fftshift(H5);

a = 6000;                                    % cut-off frequency

w0 =(-N/2+1:(N/2));                          % Center Frequency Vector

w = w0.*Fs/N;

                                             % SAMPLING FREQUENCY

H2 = 1-(a./(a + (1i*w)));

Hshift = fftshift(H2);

Y = X .*Hshift5' .* Hshift3' .*Hshift' ;     % signal filter, transpose to get 2 matrices

y = real(ifft(Y));

y = y*(max(abs(x))/ max(abs(y)));            % of original signal

                                             % low pass bass signal

a1 = 100;                                    % cut-off frequency
 w1 = (-N/2+1:(N/2));                        % Center Frequency Vector

 w2 = w1.*Fs/N;                              % SAMPLING FREQUENCY

 H1 = a1./(a1 + 1i*w2);                      % H is in the center

 Hshift1 = fftshift(H1);                     % H is not in the center

 a10 = 50;                                   % cut-off frequency

 w10 = (-N/2+1:(N/2));                       % Center Frequency Vector

 w20 = w10.*Fs/N;                            % SAMPLING FREQUENCY

 H10 = a10./(a10 + 1i*w20);                  % H is in the center

 Hshift10 = fftshift(H10);                   % H is not in the center

 a12 = 5;                                    % cut-off frequency

 w12 = (-N/2+1:(N/2));                       % Center Frequency Vector
```

```
w22 = w12.*Fs/N;                                % SAMPLING FREQUENCY

H12 = a10./(a10 + 1i*w22);                      % H is in the center

Hshift12 = fftshift(H12);                       % H is not in the center

Y1 = X .*Hshift1' .*Hshift10' .*Hshift12';      % signal filter

y1 = real(ifft(Y1));

y1 = y1*(max(abs(x))/ max(abs(y1)));            % of original signal

sound(x,Fs)                                     % original sound

sound(y1,Fs)                                    %sound after low pass filter

sound(y,Fs)                                     %sound after high pass filter

subplot(3,1,1);

plot(w5,abs(fftshift(X)))          % if complex number then take its magnitude

title(' Initial signal ');

subplot(3,1,2);
 plot(w5,abs(fftshift(Y)))

 title(' Signal when high-pass filtered ');

 subplot(3,1,3);

 plot(w2,abs(fftshift(Y1)))

 title('  Signal when low-pass filtered ');
```






NoSQL (Not Only SQL) databases are a type of database that does not use the traditional relational model. Instead, they are designed to handle large, unstructured or semi-structured data sets efficiently. Store data as JSON, BSON, or XML documents (e.g. MongoDB, CouchDB). Key-Value Stores store data as key-value pairs (e.g. Redis, DynamoDB). Wide-Column Stores store data in tables, but each row can have a different number of columns (e.g. Cassandra, HBase). Graph Databases store data as a graph, with nodes and edges (e.g. Neo4j, ArangoDB). Advantages of NoSQL Databases with scalability: NoSQL databases can scale horizontally easily by adding more servers. High performance is optimized for fast read and write operations. Flexible and does not require fixed schema, easy to change data structure. Limitation in complex queries is not good support for complex queries like SQL. Practical application with Web Applications is using NoSQL databases to handle requests quickly and manage spikes in traffic. E-commerce with managing large data volumes and spikes during shopping season. NoSQL databases are a popular choice for modern applications that require high speed and scalability.

## Conclusion

Software Engineering is a broad and diverse field that encompasses various aspects of software development, maintenance, and management. Switching from Matlab programming to Python in the context of university studies not only helps students master the basic and advanced concepts of programming but also equips them with the necessary skills to apply in real-world projects. Software Engineering is a diverse and rich field that provides students with the skills and knowledge necessary to succeed in the modern software industry. Learning and applying Python in Software Engineering not only helps students develop programming skills but also opens up many career opportunities in fields such as data science, machine learning, and software project management. Using ChatGPT to convert code from MATLAB to Python can bring many benefits such as saving time: ChatGPT can automatically convert code segments, saving you time compared to having to rewrite them from scratch. Minimize errors with automated conversions that can reduce human errors when rewriting code. Learn more about programming experience as you can learn how to write Python code from converted MATLAB code, helping to improve your programming skills. Easy integration is that Python has many powerful and popular libraries, making it easy to integrate converted code into larger projects. As for the complete code conversion rate, this depends on the complexity of the original MATLAB code. ChatGPT can convert most basic and average code, but may have difficulty with complex code or code that uses specialized MATLAB libraries. However, you can expect a fairly high conversion rate, usually above 80%, and manual processing of software engineering.

## Conflict of interest

Microsoft Copilot can help you write a scientific research paper through many useful features: Features that support writing a research paper such as planning and outlining by Copilot help you create an outline and organize your thoughts logically. Editing and formatting content, you can ask Copilot to edit, rewrite, or improve the grammar and tone of the text. In-depth research such as information retrieval, Copilot can help you find and synthesize information from many different sources. Data analysis, Copilot has the ability to turn raw data into spreadsheets and run Python code to analyze data. Support for writing long texts, text editing, Copilot helps you compose long and complex paragraphs, from initial ideas to complete drafts. How to use Copilot, you can use Copilot on many different platforms, including PC, Mac, mobile devices, and directly on the web at copilot.microsoft.com.

## Acknowledgements

We would like to thank Tay Do University for providing time and specialized laboratories for software engineering and FPT University's FSB Institute of Management & Technology for offering the Master of Software Engineering (MSE) program. This program is designed to meet the growing needs of the domestic and foreign software market, especially in the leadership and management roles of large international software projects.

## References

1. Copilot. Microsoft Copilot is your companion to inform, entertain, and inspire. Get advice, feedback, and straightforward answers. Microsoft 365 Copilot (2025).
2. Coursera. Online Python Courses. Retrieved from Coursera partners with more than 350 leading universities and companies to bring flexible, affordable, job-relevant online learning to individuals and organizations worldwide (2025).
3. Fsb.edu.vn. Training Program. FSB Institute of Management & Technology. FSB Institute of Management & Technology, FPT University (2025).
4. MathWorks. R2024b Release Highlights – MATLAB and Simulink. Retrieved from © 1994-2025 The MathWorks, Inc. - MathWorks is the leading developer of mathematical computing software for engineers and scientists (2025).
5. University TD. Tay Do University. Retrieved from Tay Do University (2023).