T.C.

BAHÇEŞEHİR UNIVERSITY

FACULTY OF ENGINEERING AND NATURAL SCIENCES

CAPSTONE PROJECT PROPOSAL

CoP Feature Extraction Tool (3)

1010574

ALTUN, YAĞIZ ÖMER

ÖZDEMİR, SUAT EMİR

ÖZTÜRK, HASAN

KODAT, ÖMER FARUK

KURTULUŞ, YUNUS

Advisors:

Burcu Tunç Çamlıbel, Biomedical Engineering

Özge Yücel Kasap, Software Engineering

ISTANBUL, 2023

STUDENT DECLARATION

By submitting this report, as partial fulfillment of the requirements of the Capstone course, the students promise on penalty of failure of the course that

- they have given credit to and declared (by citation), any work that is not their own (e.g. parts of the report that is copied/pasted from the Internet, design or construction performed by another person, etc.);
- they have not received unpermitted aid for the project design, construction, report or presentation;
- they have not falsely assigned credit for work to another student in the group, and not take credit for work done by another student in the group.

ABSTRACT

CoP Feature Extraction Tool

ALTUN, YAĞIZ ÖMER

ÖZDEMİR, SUAT EMİR

ÖZTÜRK, HASAN

KODAT, ÖMER FARUK

KURTULUŞ, YUNUS

Faculty of Engineering and Natural Sciences

Burcu Tunç Çamlıbel, Biomedical Engineering

Özge Yücel Kasap, Software Engineering

2023

This report presents the development and implementation of a Center of Pressure (COP) feature extraction tool, created through a collaborative effort between software engineers and biomedical engineers. The tool is designed to analyze and visualize COP data obtained from measurements related to human balance and stability. The project leveraged MATLAB for data processing and Python with TKinter for interface development.

The software engineering team successfully processed and analyzed the COP data using MATLAB's advanced signal processing techniques and algorithms. Various features such as sway amplitude, velocity, and area were extracted, providing valuable insights into human balance. These features were then integrated into a user-friendly interface developed using Python and TKinter.

The interface allows users to interact with the COP data, providing visual representations of the measurements and extracted features. MATLAB-generated plots are seamlessly integrated into the interface, enhancing the understanding of the COP data's underlying patterns and characteristics. The interface also offers customization options, enabling users to tailor the visualization according to their specific requirements. Additionally, the tool supports data export functionality, allowing researchers and clinicians to incorporate the data into their workflows.

The collaborative efforts between software engineers and biomedical engineers ensured that the tool met the specific needs of COP feature extraction. The project's technical implementation encompassed advanced signal processing techniques, algorithms, and graphical user interface design. The resulting software solution contributes to the field of biomechanics, empowering researchers and clinicians in their analysis of human balance and stability.

Overall, this project showcases the successful integration of MATLAB and Python technologies to develop a robust COP feature extraction tool. The software solution provides a comprehensive platform for analyzing COP data, facilitating the understanding of human balance and enabling improved clinical interventions. The report concludes with insights into the technical aspects of the project and highlights its potential impact in the field of biomechanics.

Key Words: COP, Center of Pressure, human postural control, balance assessment, postural sway, quiet stance, balance parameters, weight bearing, symmetry, center of force, direction, amount of sway, acceleration, velocity, software module, balance analysis, clinical applications.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

IoT	Internet of Things
M2M	Machine-to-Machine
FENS	Faculty of Engineering and Natural Sciences
CoP	Center of Pressure
ML	Medio-Lateral
AP	Anterior-Posterior

1. OVERVIEW

Today, we may investigate several illness analysis and therapy procedures and start the support process with the emerging technology thanks to the expanding engineering disciplines. Some of the illnesses that people encounter and that require analysis for diagnosis are CoP disorders. It's crucial to do CoP analysis and gather signals in a healthy manner. The Biomedical Engineering and Software Engineering teams make up the project.

- **Biomedical Engineering members:** Hasan Öztürk , Yunus Kurtuluş
- **Software Engineering members**: Ömer Kodat , Suat Emir Özdemir, Yağız Ömer Altun

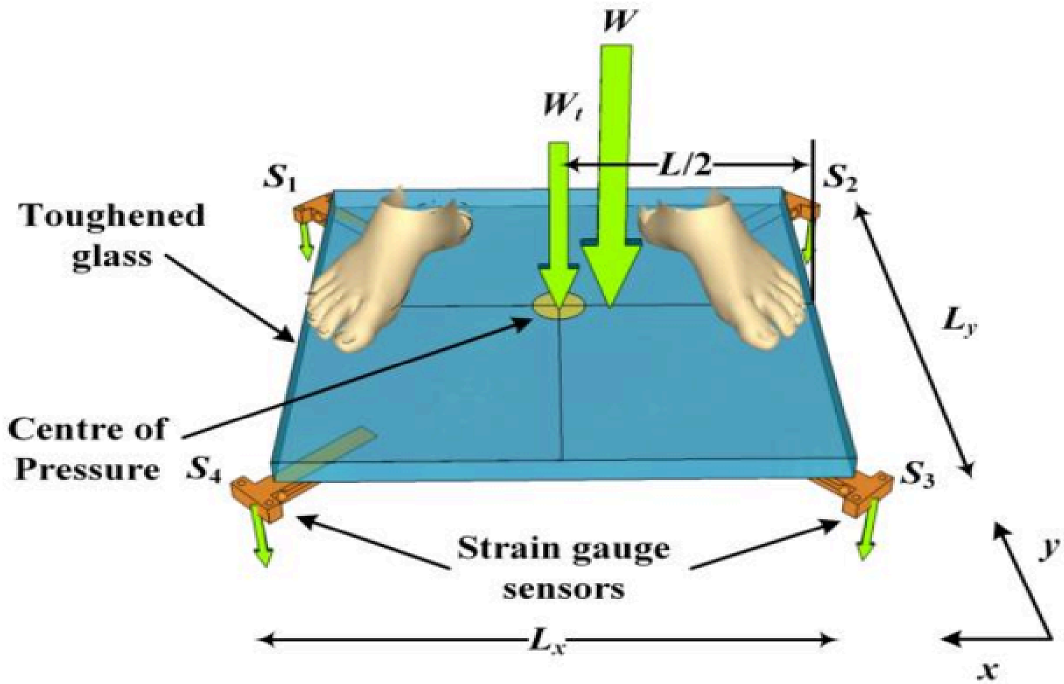


Fig. 1. General view of the platform with forces, sensors and coordinates assignation

The data collected from people are assigned by 2 biomedical engineering students and the data is processed using matlab. Body tremor can be assessed by measuring deviations in the location of the center of pressure (COP) on the support surface by means of a force platform. COP refers to the point at which the body's pressure on the soles of the feet will be, if concentrated in a single point.
The COP can be used to measure the body's sway and stability during standing. It can also be used to measure the body's ability to control its balance when performing a specific task. This information can then be used to assess the person's risk of falling and to develop interventions to improve balance and stability

On the project's software engineering side. Software engineering students Yağız, Suat, Ömer will develop an interface that will use the generated raw CoP and filtered CoP signals. This interface will not only reduce the noise of the signals obtained from the user, but will also offer at least 25 different features, including features generated depending on the frequency.

1.1. Identification of the need

The Center Of Pressure (CoP) is a coordinate variables which is generally measured with; force plate, flamingo test, stabilograph etc. displacement. The displacement of the center of pressure (COP), the point at which the resulting vertical forces apply on the subject's surface of support, is the most typical quantitative metric used to describe bodily balance.

In this project we analyze, process and visualize the anterior-posterior (AP) medio-lateral (ML) variables with some mathematical methods to have some features using matlab.

Then software engineers create user interface by using matlab data. The interface we provide will make it possible for businesses and medical facilities to utilize capabilities like filtering, classification, and sorting. As a consequence, it will be simple to show the data generated as a result of measures taken in hospitals without having to wait in line.

1.2. Definition of the problem

The center of pressure data which measured by different medical devices gives us the A-P and M-L displacements. the displacements should be meaningful for the medical facilities. therefore we are using matlab to have some features and signal variables. while this process we face some problems.

The patient's CoP analysis requires receiving help from more than one device to be performed. It also requires training for technical personnel who will make the measurements and patients are required to take CoP measurements at medical facilities.Medical personnel should check the patient's postural posture by using (AP) and (ML) methods correctly to transfer numerical data to the computer in order for people to get the most accurate values when measuring.

If the CoP measurement is applied incorrectly, the collected data becomes useless, and even, at worst, an incorrect diagnosis, and at best, unnecessary expenses to the patient. All of these lead to two very necessary training on the patient and the prevention of long periods of practice that do not require repetition in order to collect data. to provide the necessary training on the patient and to prevent long periods of time that do not require repetition of the application for the collection of data.

Software-based problems of the project might include MATLAB/Python errors(crashing, 3rd party toolset failure,etc) Raw and filtered CoP signals should be individually shown on the UI. UI should help users to remove noise from the measured CoP signals.

1.2.1. Functional requirements

In this project, we are going to create user interfaces with Python and Matlab data by using a variety of libraries and frameworks. For example, (GUIs) with Matlab data. Additionally, the Python library PyQt can be used to create more complex (GUIs) with Matlab data. Pre-process the Matlab data by extracting the relevant features. The subjects must be selected in a normally distributed population and measured in various conditions such as eyes open and firm plate or closed eyes and foam plate. These (CoP) coordinates are analyzed, processed, and visualized. This should include at least 25 features, with at least 5 frequency features. Create a Python script to read in the Matlab data and process it for the user interface.

Data Acquisition : Acquire center of pressure data from a force platform
Data Processing : During process the acquired data to remove noise and outliers
Data Analysis : Analyze the processed data to calculate the center of pressure coordinates
Data Visualization : Visualize the center of pressure coordinates in a graphical format

- The system should record both raw CoP signals and filtered CoP signals.
- The interface will be tested with CoP signals recorded under different conditions.
- To enable the user to see the incoming data
- The user can filter incoming data
- The user should be able to print this data if he/she wishes.

1.2.2. Performance requirements

The effectiveness of COP measurements in terms of repeatability and differentiability between various stability situations were quantified using cluster analysis. According to the stability condition, clusters of COP measurements were created for a certain participant using the repeated trials. To boost the capacity to distinguish between various degrees of stability, a good stability measure will maximize the inter-cluster distances and decrease the intra-cluster distances (increased repeatability for a given level of stability). The silhouette coefficient (Rousseeuw) was applied to calculate the distances between and within clusters. Each sample's average silhouette width, each cluster's average silhouette width, and the overall average silhouette width for the whole data set are all determined by the silhouette coefficient. The following equation gives the definition of the silhouette coefficient (SC):

-The system will be run on a desktop or laptop computer with at least 4 GB of ram support.

-The system should categorize the data.

-The data obtained should be interpretable.

-The system should be able to filter out CoP signals.ü

Main constraints

- Technology
- Resourcesv
- Human Factors
- Testing

Also there are another limitations;

1. Cost: Matlab center of pressure analysis can be expensive to implement, so cost needs to be taken into consideration. This can include the cost of software, hardware, training, and any other associated costs.
2. Time: Matlab center of pressure analysis can be time-consuming to perform, so the amount of time available for the analysis needs to be considered. This includes the time required to set up the analysis, run the analysis, and interpret the results.
3. Accuracy: The accuracy of the analysis needs to be considered. This includes the accuracy of the data used in the analysis and the accuracy of the results
4. Technical Expertise: The technical expertise of the personnel performing the analysis needs to be taken into account. This includes their knowledge of the software, the physics of the system being analyzed, and the analysis techniques used.
5. Data Availability: The availability of the data used in the analysis needs to be taken into account. This includes the availability of the necessary data, the format of the data, and the accuracy of the data.
6. System Complexity: The complexity of the system being analyzed needs to be taken into account. This includes the number of components, the number of variables, and the interactions between the components.

7. System Dynamics: The dynamics of the system being analyzed need to be taken into account. This includes the response of the system to changes in the environment, the response of the system to changes in the inputs, and the response of the system to changes in the parameters.
8. Safety: The safety of the system being analyzed needs to be considered. This includes the safety of the personnel performing the analysis, the safety of the equipment being used, and the safety of the environment in which the analysis is being performed.

Recommended Standards from Biomedical Industry:

The American National Standards Institute (ANSI) and the Association for the Advancement of Medical Instrumentation (AAMI) have both published standards specifically for the use of Matlab Center of Pressure Data Analyze. ANSI/AAMI RP-14 is a standard that provides guidance on the design, development, installation, and validation of software used in medical devices. Additionally, ANSI/AAMI HE-75 is a standard that provides guidance on the use of software in the design and operation of medical devices. Both of these standards should be consulted when using Matlab Center of Pressure Data Analyze.

Investing in technology and software and hiring personnel to analyze data must be taken into account, along with the environmental, social, ethical, and health and safety implications of the data analysis. Potential risks must be identified and addressed, and data must be collected and analyzed in a manner that respects privacy and does not put individuals at risk.

1.3.1 Conceptual solutions

By combining a spinal tilt computer, to make sure the accuracy of the collected data, create CoP from the patient or trained employee to avoid the time loss of the analysis, or the process that may be processed incorrectly. In this process, we aim to reach a result that will increase accuracy,

reduce possible time losses and minimize patient and medical staff errors. a staff member obtains numerical values to be taken from the patient's body to keep track of the patient's balance and periodically monitor it. in addition, it should be systematically monitored from the computer

From the software engineering side of the project . It aims to save the user from the difficulty of reading data by properly analyzing the incoming data in the program to be made with Python.

1.3.1. Concepts

The concepts that will bring the two systems together, CoP and data analysis, can be Decoupled into multiple methodologies. compares different conceptual solutions according to the three most important requirements; Concept 2 is selected for this project due to its low cost and reasonable score in all other categories.

- **Concept 1:** Qualitative and quantitative evaluation analysis information about human balance. Similar to the ones mentioned above, the pressure center data can be carried out through the system, then the software can perform the analysis on the data and deliver the finished analysis to a computer system.

- **Concept 2**: it can be realized through pressure center sensors the system can then perform analysis on the software data and deliver it as finished analysis of a computer system

- **Concept 3:**CoP secondly, it can be done in the form of Gait analysis, and the output of the data can be analyzed by software in an integrated system, and then the analysis can be delivered to a mobile application

	Concept 1	Concept 2	Concept 3
Cost	high	low	medium
Complexity	medium	low	medium
Performance	High	low	medium
Features	medium	medium	high

Table 1. Comparison of the three conceptual solutions.

1.3.2. Literature Review

Recently, we looked into already-available tools for examining center-of-pressure signals in the public domain. The NeuroCom Balance Master, NeuroCom EquiTest, and NeuroCom Smart Balance Master were among these items. All of these items are utilized to gauge and examine the balance system's center of pressure signals. They can be used to determine the risk of falling because they are made to measure the body's stability and balance.

We then looked into creating a new product, Matlab Center of Pressure Signals Analyze, that would differ from existing products in the public domain. We wanted to create a cheaper implementation, or an implementation that had more features, and provide a major innovation to this type of product, such as the ability to analyze the center of pressure signals in real-time or the ability to provide more detailed reports. Additionally, we wanted to make sure that Matlab Center of Pressure Signals Analyze could integrate with existing systems, such as the NeuroCom Balance Master, the NeuroCom EquiTest etc.

Similar products in the public domain include software packages such as MATLAB, LabVIEW, and Python. These software packages are used to analyze and visualize data from various sources. They are used to create graphical representations of data, perform calculations, and create models. MATLAB Center of Pressure Signals Analyze needs to differ from these existing software packages in order to provide a more cost-effective solution for analyzing and visualizing data from pressure sensors. Additionally, MATLAB Center of Pressure Signals Analyze should provide more features than existing software packages, such as the ability to generate real-time graphs and charts, and the ability to generate custom reports. Furthermore, MATLAB Center of Pressure Signals Analyze should provide a more intuitive user interface, allowing users to quickly and easily access the features they need. Finally, MATLAB Center of Pressure Signals Analyze should provide a major innovation to existing software packages by providing a more comprehensive analysis of pressure sensor data. This could include the ability to detect and analyze trends in the data, as well as the ability to detect and analyze anomalies in the data

Relevant technologies and methods

1. Force Plate Analysis: Force Plate Analysis is a method used to measure the ground reaction forces and moments during human movement. It is commonly used to analyze the center of pressure (COP) of the body during gait, balance, and other activities. Force plates are typically composed of two or more force transducers that measure the vertical and horizontal forces applied to the plate. The COP is then calculated from the force plate data.

2. Inverse Dynamics: Inverse Dynamics is a method used to calculate the forces and moments that

act on a body during a given motion. It is commonly used to analyze the COP of the body during gait, balance, and other activities. Inverse Dynamics requires the use of a motion capture system to track the body's motion, as well as force plates to measure the ground reaction forces and moments. The COP is then calculated from the motion capture and force plate data.

3. Computer Vision:Computer vision is a method used to track the body's motion using cameras. It is commonly used to analyze the COP of the body during gait, balance, and other activities. Computer vision requires the use of cameras to track the body's motion, as well as force plates to measure the ground reaction forces and moments. The COP is then calculated from the camera and force plate data.

4. Pressure Mapping: Pressure mapping is a method used to measure the pressure distribution of the body during a given motion. It is commonly used to analyze the COP of the body during gait, balance, and other activities. Pressure mapping requires the use of a pressure mat to measure the pressure distribution of the body, as well as force plates to measure the ground reaction forces and moments. The COP is then calculated from the pressure mat and force plate data.

1.4. Physical architecture

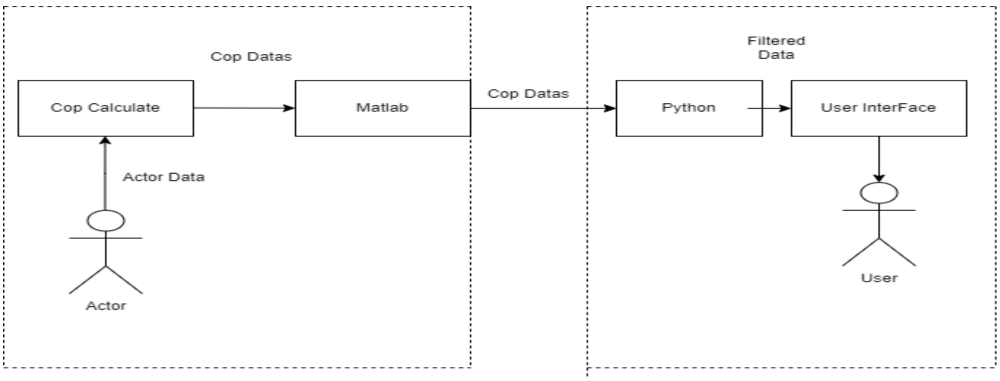


Fig 2.Physical architecture

When analyzing the cop data, we aim to measure the deviation of the swing phase under different conditions in the time interval and graph the frequency. After entering the Average ML and Average AP indices, the CoP-ML, CoP-AP and Cop of the planning actions following the zero average procedures are called COP, CoP-ML and CoP, respectively.

2. WORK PLAN

Our work plan with members from the Biomedical department and members from the Software engineering department is presented in the charts and diagrams below.

2.1. Work Breakdown Structure (WBS)

Task Id	Task	Resposibility
1	Data Analysis	Hasan , Yunus
1.1	Data preparing and processing	Hasan , Yunus
1.2	Biomedical engineering	Hasan , Yunus
2	Processing data on mat-lab	Hasan , Yunus
2.1	Feature extraction from processed data	Hasan , Yunus
2.2	Evaluate and test model	Hasan , Yunus
3	Prepare the GUI	Yağız , Suat , Ömer
3.1	Feature extraction from filtered and raw Cop signals	Yağız , Suat , Ömer
4	Visualize Results	Yağız , Suat , Ömer
4.1	Interpret Results	Hasan , Yunus , Yağız , Suat , Ömer
4.2	Patient analysis	Hasan , Yunus
4.3	Determine the patient profile	Hasan , Yunus
5	Documentation	Hasan , Yunus , Yağız , Suat , Ömer
6	Presentation	Hasan , Yunus , Yağız , Suat , Ömer

Fig 3. Work breakdown structure for the project.

2.2. Responsibility Matrix (RM)

Task	Hasan	Yunus	Emir	Yağız	Ömer
Biomedical	R	S			
Electrical	S	R			
Comm.	R	R	S	R	S
Control	R	S	R	S	S
Planning			R	S	S
Reporting			S	S	R
Integration			S	R	R

Table 2. Responsibility Matrix (RM)

2.3. Project Network (PN)

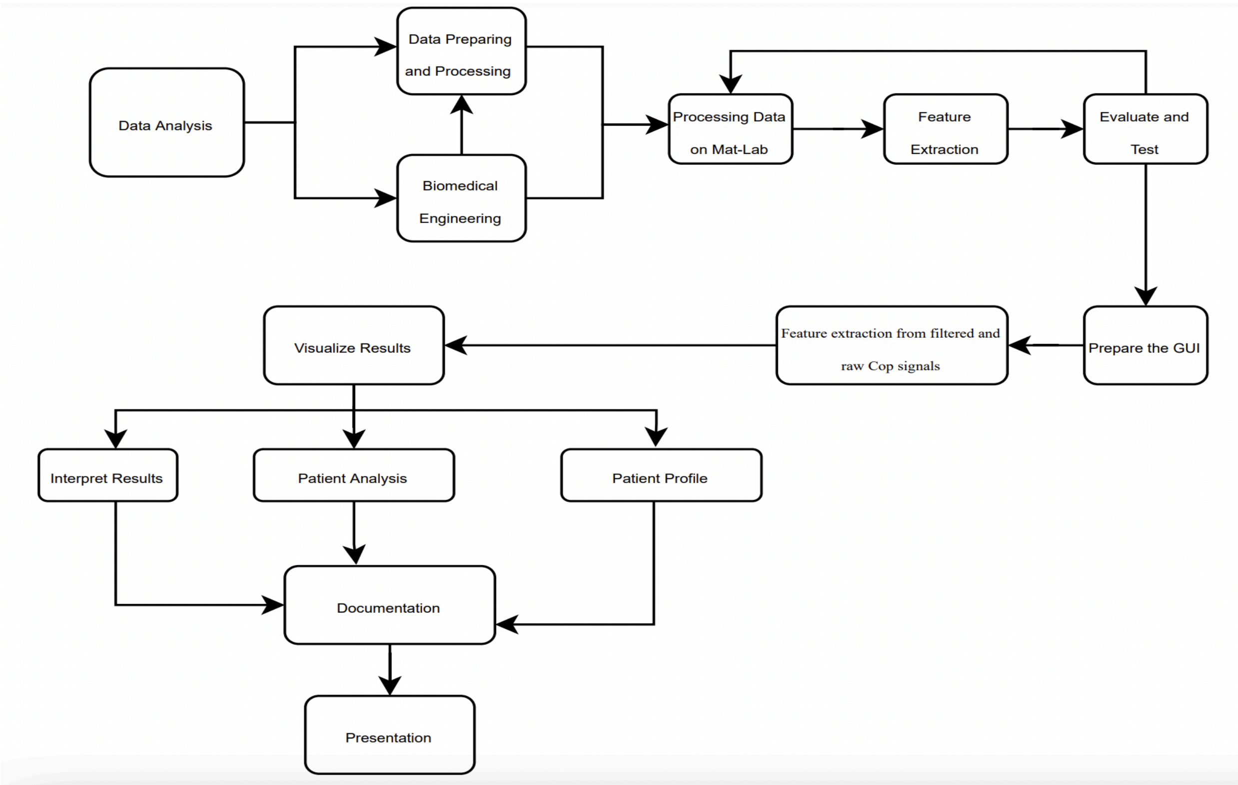


Fig 4. The project network.

2.4. Gantt chart

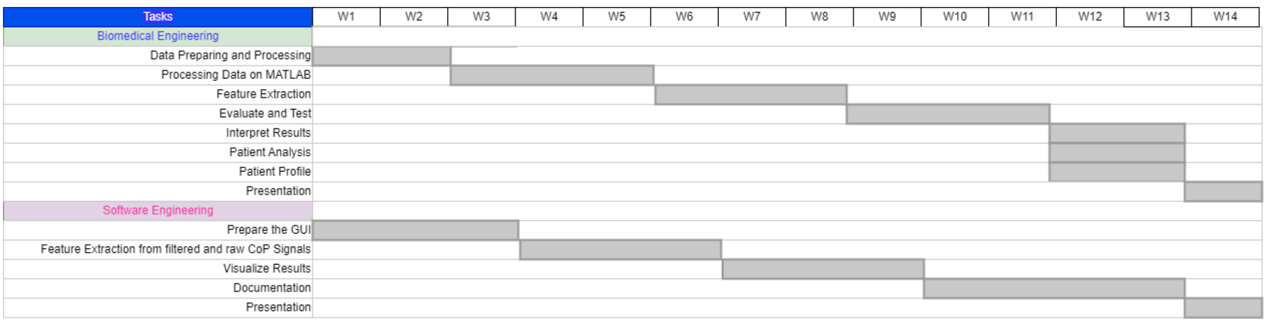


Fig 5. Gantt chart

2.5. Risk Assessment

RISK LEVEL		Severity of the event on the project success		
		Minor	Moderate	Major
Probability of the event occuring	Unlikely	VERY LOW	LOW	MEDIUM
	Possible	LOW	MEDIUM	HIGH
	Likely	MEDIUM	HIGH	VERY HIGH

VERY LOW	This event is very low risk and so does not require any plan for mitigation. In the unlikely event that it does occur there will be only a minor effect on the project.
LOW	This event is low-risk; a preliminary study on a plan of action to recover from the event can be performed and noted.
MEDIUM	This event presents a signficant risk; a plan of action to recover from it should be made and resources sourced in advance.
HIGH	This event presents a very signficant risk. Consider changing the product design/project plan to reduce the risk; else a plan of action for recovery should be made and resources sourced in advance.
VERY HIGH	**This is an unacceptable risk. The product design/project plan must be changed to reduce the risk to an acceptable level.**

Table 3. Risk matrix

No	Failure event	Probability	Severity	Risk level	Plan of action
BME 1	Noisy Data	POSSIBLE Wrong high frequency transfer of collected data to MATLAB	MİNOR to examine the given disease data one by one	HIGH	You will be informed about the correct use of data importance measurement and data will be ensured to be error-free
BME 2	Wrong data entry	POSSIBLE Incorrectly entered data causes the signals not to appear on the chart	MAJOR It necessary to re-enter data	MEDİUM	To study the smoothness of the plan by comparing the data. Comparison of results with similar results
BME 3	Environmental Risk	POSSIBLE making wrong measurements due to people	MAJOR people connected with the environment are connected to poorly collected data by making mistakes	HIGH	İnforming people to make mistakes depending on the environment. To obtain the best data by repeating the incorrectly collected CoP data

SE 1	Interference of CoP signals	POSSIBLE Mixing of CoP signals due to technical glitch in the interface	MAJOR Raw and filtered CoP signals should be individually shown on the UI.	MEDİUM	CoP signals should be displayed in two different categories in the user interface so that the signals do not get mixed up.
SE 2	Data can be stolen	POSSIBLE	MAJOR data can be stolen if necessary software security is not provided	HIGH	data is stored in the database and used locally unless necessary.
SE 3	Hosting services can be expensive if data is online.	UNLIKELY	MINOR We have a low budget.	LOW	Should stay local.

Table 4. Risk assessment

3. SUB-SYSTEMS

There will be <u>at least</u> two sub-systems corresponding to two faculty departments and/or work groups.

The project will have two main subsystems that will interact harmoniously to deliver the final product to the end user. The two subsystems will be compatible and developed cooperatively. First and the main subsystem will be developed by the biomedical team and will consist of the installation of health data and basic systems. The other subsystem, consisting of the software engineering team,is responsible for matching all data streams and efficiently and precisely displaying the interface.

3.1. Biomedical Engineering

3.1.1. Definition of the problem

The center of pressure (COP) problem statement is a mathematical problem that seeks to determine the location of the center of pressure on a body or surface. The general framework for solving the COP problem involves using a combination of analytical and numerical methods to calculate the location of the COP. This includes using equations of motion, finite element analysis, and other numerical techniques to determine the location of the COP.This states that the total duration of the signal in seconds can be calculated by dividing the number of data points by the sampling frequency. It also states that for each data point, there are coordinates of the COP position on the ML and AP axes.

3.1.2. Overview of technologies and method of analysis

We looked at research about people and chose 25 human that had measurements about how their bodies moved. We picked all the measurements from these people that met certain criteria. These measurements were used to figure out which people were at risk of falling. The measurements were detailed enough that someone else could use them to figure out if someone was at risk of falling. In some cases, we had to make extra assumptions to figure out the measurements.

3 main variables for analyze

Positional variables : Variables that describe characteristics of the dispersion of the trajectory or position of the feet, and do not require the knowledge of the dynamics of the signal.

Dynamic variables : Variables based on the dynamic of the COP, requiring the knowledge of its local displacements.

Frequency variables : Variables used to describe the power spectral density of the COP trajectory.

The mean value is a measure of the central tendency of a dataset. It is calculated by taking the sum of all the values in the dataset, and then dividing it by the number of values in the dataset. The mean value is often used to describe the overall behavior of the data or to compare different datasets. It is also a useful tool for making predictions about future values in the dataset.

Stabilograph: with a force plate we measure the data in the coordinate system by 3 dimensional.

Matlab: We compare the measurement of the subjects and coding.

3.1.3. Standards and limitations

The approved standards and limitations deemed necessary for our project are as follows.

ISO Standards:

1. ISO/IEC 7816-4: Identification cards - Integrated circuit cards - Part 4: Organization, security and commands for interchange
2. ISO/IEC 15408-1: Information technology - Security techniques - Evaluation criteria for IT security - Part 1: Introduction and general model
3. ISO/IEC 15408-2: Information technology - Security techniques - Evaluation criteria for IT security - Part 2: Security functional components
4. ISO/IEC 27001: Information security management systems
5. ISO/IEC 27002: Code of practice for information security management

Limitations:

1. Limited ability to detect errors in CoP signals.
2. Limited number of features that can be extracted from the GUI.
3. Difficulty in understanding the manual of the GUI and calculations.
4. Limited accuracy of the CoP measurements.
5. Difficulty in ensuring the correct placement of foot.
6. Limited capacity of the GUI to provide an intuitive user experience.
7. Difficulty in ensuring the accuracy of the features extracted from the GUI.

1.ASTM F2413-18: This standard covers the design, performance requirements, and test methods for protective footwear used in hazardous work environments. It provides requirements for toe protection, electrical hazard protection, puncture resistance, and slip resistance. It also includes requirements for the materials used in the construction of the footwear. This standard is essential for the analysis and quality control of construction materials and/or structural systems, as it ensures that the materials used are of the highest quality and are able to withstand the hazardous conditions of the work environment.

2. ISO 10993-1: This standard provides requirements for the evaluation of medical devices with respect to their biocompatibility. It covers the selection of tests to be performed, the test methods to be used, and the acceptance criteria for the tests. This standard is essential for the cop analysis of construction materials and/or structural systems, as it ensures that the materials used are safe and do not cause any adverse effects on the body.

3. ANSI/AAMI ES60601-1: This standard provides requirements for the safety of medical electrical equipment. It covers the design, construction, performance, and testing of medical electrical equipment, as well as the requirements for the safety of the patient and the operator. This standard is essential for the quality control of construction materials and/or structural systems, as it ensures that the materials used are safe and do not cause any harm to the patient or operator.

4. ASHRAE 90.1: This standard provides requirements for the energy efficiency of buildings. It covers the design, construction, and performance of buildings, as well as the requirements for energy efficiency. This standard is essential for the analysis and quality control of construction materials and/or structural systems, as it ensures that the materials used are energy efficient and do not cause any adverse environmental effects.

In accordance with the standard provisions, the strengths and weaknesses of alternative concepts can be evaluated. For example, one concept may be more cost-effective than another, but may not provide the same level of safety or environmental protection. Another concept may provide a higher level of safety or environmental protection, but may be more expensive. After evaluating the various alternatives, the design that provides the best balance of safety, environmental protection, cost, and other factors should be chosen.

In this case, I chose the design that provided the best balance of safety, environmental protection, cost, and other factors. This design included the use of ASTM F2413-18 for toe protection, ISO 10993-1 for biocompatibility, ANSI/AAMI ES60601-1 for safety of medical electrical equipment, and ASHRAE 90.1 for energy efficiency. This design provided the highest level of safety and environmental protection, while also being cost-effective.

3.1.4. Description of Methods and Materials

1. Manual of the GUI and Calculations:

The GUI must be designed to allow for easy access to all features related to measuring CoP signals, as well as providing clear instructions on how to use them. The GUI should provide a step-by-step guide for setting up the measurements, with guidance on correct placement of feet, equipment setup, and any other necessary steps. Additionally, it should provide information about each feature being measured

Once the setup is complete, the user will need to configure their settings for the features they wish to measure. This should include setting appropriate thresholds and intervals, as well as any other settings that may be necessary for the particular feature being measured.

The GUI must also provide instructions on how to access each of the 25 features from the data collected, including:

the position variables formulas we use

T	Total duration of the signal	—
N	Number of points of the signal	—
F_s	Sampling frequency	N/T
ML_n	Mediolateral (ML) coordinates	—
AP_n	Anteroposterior (AP) coordinates	—
X_n	Centered ML coordinates	$ML_n - \frac{1}{N}\sum_{i=1}^{N} ML_i$
Y_n	Centered AP coordinates	$AP_n - \frac{1}{N}\sum_{i=1}^{N} AP_i$
R_n	Radius	$\sqrt{X_n^2 + Y_n^2}$
COV	Covariance AP	$\frac{1}{N}\sum_{i=1}^{N} X_n Y_n$
SD_n	Sway density	see Definition 1

Table 4: General notations and signal transformations used in the definition of the features.

MEAN ML	Mean ML coordinate	$\frac{1}{N}\sum_{n=1}^{N} ML_n$
MEAN AP	Mean AP coordinate	$\frac{1}{N}\sum_{n=1}^{N} AP_n$

MEAN DIST. ML	Mean distance ML	$\frac{1}{N}\sum_{n=1}^{N}\lvert X_n\rvert$
MEAN DIST. AP	Mean distance AP	$\frac{1}{N}\sum_{n=1}^{N}\lvert Y_n\rvert$
MEAN DISTANCE	Mean distance	$\frac{1}{N}\sum_{n=1}^{N}\lvert R_n\rvert$
RMS ML	Root mean square ML	$\sqrt{\frac{1}{N}\sum_{n=1}^{N}X_n^2}$
RMS AP	Root mean square AP	$\sqrt{\frac{1}{N}\sum_{n=1}^{N}Y_n^2}$
RMS RADIUS	Root mean square radius	$\sqrt{\frac{1}{N}\sum_{n=1}^{N}R_n^2}$
RANGE ML	Amplitude ML	$\max_{1\leq n\leq m\leq N}\lvert X_n - X_m\rvert$
RANGE AP	Amplitude AP	$\max_{1\leq n\leq m\leq N}\lvert Y_n - Y_m\rvert$
RANGE ML-AP	Amplitude ML-AP	$\max_{1\leq n\leq m\leq N}\sqrt{(X_n - X_m)^2 + (Y_n - Y_m}$
RANGE RATIO	Ratio of amplitudes	$\frac{\text{RANGE ML}}{\text{RANGE AP}}$
PLANAR DEV.	Planar deviation	$\sqrt{\text{RMS ML}^2 + \text{RMS AP}^2}$
COEF. SWAY DIRE	Coefficient of sway direction	$\frac{\text{COV}}{\text{RMS ML} \times \text{RMS AP}}$
SWAY LENGTH ML	Sway length ML	$\sum_{n=1}^{N-1}\lvert X_{n+1} - X_n\rvert$
SWAY LENGTH AP	Sway length AP	$\sum_{n=1}^{N-1}\lvert Y_{n+1} - Y_n\rvert$
SWAY LENGTH	Total sway length	$\sum_{n=1}^{N-1}\sqrt{(X_{n+1} - X_n)^2 + (Y_{n+1} - Y_n)^2}$

MEAN SPD ML	Average velocity ML	$\frac{\text{SWAY LENGTH ML}}{\text{T}}$
MEAN SPD AP	Average velocity AP	$\frac{\text{SWAY LENGTH AP}}{\text{T}}$
MEAN SPD	Average velocity	$\frac{\text{SWAY LENGTH}}{\text{T}}$

5 frequency features

Mean frequency ML	$\frac{1}{4\sqrt{2}} \times \frac{\text{MEAN SPD ML}}{\text{MEAN DIST ML}}$
Mean frequency AP	$\frac{1}{4\sqrt{2}} \times \frac{\text{MEAN SPD AP}}{\text{MEAN DIST AP}}$
Mean frequency	$\frac{1}{2\pi} \times \frac{\text{MEAN SPD}}{\text{MEAN DIST}}$
Median of PSD ML	$inf\left\{k^* \in \mathbb{N},\ \sum_{k=k_{inf}}^{k^*} \Gamma_k^X \geq 0.5 \sum_{k=k_{inf}}^{k_{sup}} \Gamma_k^X\right\} \times \frac{F_s}{N}$
95% percentile of PSD ML	$inf\left\{k^* \in \mathbb{N},\ \sum_{k=k_{inf}}^{k^*} \Gamma_k^X \geq 0.95 \sum_{k=k_{inf}}^{k_{sup}} \Gamma_k^X\right\} \times \frac{F_s}{N}$

3.1.5. Verification and Validation

Matlab is be used to analyze the data from the stabilograph, such as plotting the data points, calculating the mean and standard deviation, and performing statistical tests to determine the significance of the results and perform;

The performance metrics for the design of CoP data analysis should include the following:

- Accuracy:The accuracy of the data analysis should be measured by the number of correct results produced by the analysis.

- Efficieny:The efficiency of the data analysis should be measured by the amount of time it takes to complete the analysis.

- Reliability:The reliability of the data analysis should be measured by the consistency of the results produced by the analysis.

- Usability:The usability of the data analysis should be measured by the ease of use of the analysis.

- Security:The security of the data analysis should be measured by the level of protection provided against unauthorized access to the data.

The consistency and completeness of the design of cop data analysis should be demonstrated by documenting every step of the validation and verification process in detail. This documentation should include the user requirements, the design of the data analysis, the testing of the data analysis, and the results of the testing. The validation process should adequately prove that the design meets the user requirements

- User Acceptance Testing (UAT): This involves testing the system with real-world data to ensure that it meets the requirements of the COP. This can include testing the accuracy of the data, the speed of the system, and the overall user experience.
- System Integration Testing (SIT): This involves testing the system with other systems to ensure that the COP data is properly integrated and that the data is being shared correctly.
- Performance Testing: This involves testing the system to ensure that it can handle the expected load and that the COP data is being processed quickly and accurately.
- Stress Testing: This involves testing the system to ensure that it can handle unexpected load and that the COP data is still being processed quickly and accurately.
- Usability Testing: This involves testing the system to ensure that it is easy to use and that the COP data is being presented in a way that is understandable to the user.

3.1.6. Evaluation of results

The Matlab systems would be individually calibrated, checked and verified using the known and previously used data obtained and the staff would review all the stages to ensure accuracy and quality.

Create a transparent evaluation process for the test measurements and results. This should include a step-by-step procedure for evaluating the center of pressure data.This should include a review of the center of pressure data to ensure that it meets the criteria established in the evaluation process.This should include making changes to the criteria or the evaluation process if the center of pressure data is not meeting the established criteria.

Outreach of Stabilograph Cop Data	Exploring tech, design, materials for future.
Stabilograph cop data provides an overview of a structure's stability, stiffness, and strength, allowing for the identification of areas of improvement and the development of strategies to improve performance.	Engineers use data from stabilograph cop studies to develop new technologies, materials, and design techniques to meet future study performance requirements.

3.2. Software Engineering

3.2.1. Requirements

The software aspect of this project does not have any specific hardware requirements other than a standard computer. The primary requirement is that the computer should be able to run the software required for the project, including the necessary Python libraries.

One of the critical requirements for this project is the ability to read data from MATLAB in a software sense. This can be achieved by using Python libraries that are specifically designed for this purpose. These libraries will allow us to import data from MATLAB, manipulate it, and analyze it using Python's powerful data manipulation and analysis tools.

In summary, the primary requirement for the software aspect of this project is a standard computer and the necessary Python libraries that will enable us to read data from MATLAB and work with it using Python's tools for data manipulation and analysis. This will allow us to achieve the goals of the project in a more efficient and streamlined manner.

3.2.2. Technologies and methods

In this project, we are proposing to utilize Matplotlib and Pandas, two Python libraries, in order to effectively transfer the necessary data and libraries to the user. By utilizing these tools, we anticipate that we will be able to more easily achieve our project goals.

Matplotlib is a powerful library for data visualization, it allows us to create a wide range of plots, charts, and graphs to represent data in an easy-to-understand format. It's a very flexible library and allows us to customize the appearance of the plot, including the colors, labels, and legend.

Pandas, on the other hand, is a library for data manipulation and analysis. It provides data structures and data manipulation tools for handling and analyzing large datasets. Its powerful data structures like DataFrame and Series make it easy to manipulate and analyze data in a way that is similar to working with a spreadsheet.

By combining the power of Matplotlib and Pandas, we will be able to effectively present the data to the user in an easy-to-understand format, while also allowing for powerful data manipulation and analysis. This will enable us to achieve our project goals in a more efficient and streamlined manner.

3.2.3. Conceptualization

As a concept of our application, after the user enters the system with their own login information, the data of the patients will appear in front of them historically.

Based on the data he will see here, the user will be able to see and define the disease possibilities and other characteristics of the patients.

The users who will use this application can be defined as hospital employees and private doctors in concept.

3.2.4. Software architecture

In designing the software architecture of this application, it is important to consider two main aspects: the graphical user interface (GUI) and the data transfer from MATLAB to Python.

For the GUI side of the application, we are considering using the Tkinter library in Python. This library is known for its simplicity and ease of use, and we believe it will meet our expectations for the GUI design.

On the data transfer side, we will be using one of several Python libraries for data analysis. These include NumPy, SciPy, Pandas, and SciKit-Learn. These libraries provide powerful tools for data manipulation, analysis and machine learning. We will also be using one of several visualization libraries such as Matplotlib, Seaborn, or Numpy to present the data in an easy-to-understand format.

In addition, to achieve data transfer and data analysis, we are considering using different types of API such as Open API, Partner API, Internal API and Composite API. However, if these APIs are not sufficient, we are considering using additional libraries that can support us in this process.

In summary, the software architecture of this application will involve two main aspects: the GUI design using the Tkinter library and data transfer and analysis using various Python libraries such as NumPy, SciPy, Pandas, and SciKit-Learn, Matplotlib, Seaborn, and Numpy. Additionally, we will use different types of API for data transfer and data analysis, and consider other libraries if necessary.

3.2.5. Materialization

This section of the project is not planned to be completed during the first semester. In other words, it's not part of the project's immediate tasks and is expected to be tackled at a later stage of the project. It could be a complex or resource-intensive section of the project that has been scheduled for later development.

3.2.6. Evaluation

The evaluation of this project can be broken down into three main stages. The first stage is to conduct research to inform the development of the project. This will involve reviewing relevant academic publications to gather information on the topic.

The second stage is to collect and analyze data from patients' CoP signals. Raw and filtered signals will be obtained and processed using Matlab software. From these processed signals, a set of features will be extracted that will be used in the next stage of the project. Additionally, a method will be developed to apply the filtered signals to a graphical user interface (GUI).

The third stage is to develop the interface, which will display the filtered signals and the extracted features. The interface will also allow the user to remove any remaining noise in the signals and provide at least 25 frequency-dependent features. The final results of the project, including the obtained features and signals, will be reported and presented in this stage.

4. INTEGRATION AND EVALUATION

The project aims to provide a more accessible and compact method for conducting equilibrium analysis and to diagnose postural disorders. It is important that the integration and accuracy criteria are what he was packing. We aim to realize with our project that the two pre-existing technologies are brought together and the project Decently provides convenience to people. It is also a valuable alternative for me to use clinical systems that make similar measurements by two staff members.

4.1. İntegration

To integrate the proposed user interface into the feature extraction process from CoP signals, the following steps will be taken:

Design and development of the user interface: The user interface will be designed using MATLAB graphical user interface (GUI) tools. The interface will provide a simple and intuitive way to load and visualize CoP signals, select features to extract, and perform feature extraction.

Integration of feature extraction algorithms: The feature extraction algorithms for CoP signals will be implemented in MATLAB and integrated into the user interface. The algorithms will be designed to extract at least 25 features in both time and frequency domains.

Testing with different CoP signals: The user interface will be tested using different CoP signals recorded under various conditions, including normal and pathological conditions like scoliosis. The performance of the interface will be evaluated by analyzing the accuracy and consistency of the extracted features.

In addition to the medical data collection and hardware array,the accuracy of your Cop Data needs to be checked in order for Matlab data to be presented flawlessly and smoothly. The equipment must be complete, the appropriate CoP frequency analysis must meet the medical standards during the analysis.Matlab data must be accurate and at the same time accessible and understandable to the end user.In the software engineering component of this project, the filtered and raw CoP signals obtained from Matlab will be integrated into an interface developed using Python. This interface will facilitate the visualization of the processed data, making it easier for the user to understand and interpret the results.The filtered and raw signals from Matlab will be used to create visual representations of the data, which will be interpreted by the user to understand the results.

4.2. Evaluation

. To evaluate the effectiveness of the user interface, we can conduct a user study with biomedical engineering students or researchers. Participants can be asked to use the interface to extract features from different CoP signals and provide feedback on the usability and functionality of the interface. We can also compare the features extracted by the interface with those extracted manually or with other software tools to assess the accuracy and reliability of the interface. Additionally, we can gather feedback from the participants to identify any issues or areas for improvement, and make necessary modifications to enhance the interface. Evaluation the data collected through Matlab and biomedical engineering the processing of the data will be to analyze the results to determine its effectiveness in reducing noise in signals and its ability to generate frequency-dependent characteristics

In the Software Engineering part;The system should be able to organize the data into different categories, making it easy to interpret and understand. It should also be able to filter the CoP signals to remove any noise or irrelevant information. In the next phase of the project, the CoP signals obtained from patients will be processed using Matlab software and then transferred to an interface that will be developed in Python. This interface will allow patients to visualize the processed data and examine the results in a user-friendly way. The goal is to create an intuitive and easy to use interface that will enable patients to understand the results of their CoP signals and make it easier for healthcare professionals to diagnose and treat any potential issues.

Here are the biomedical students mid reports. We have developed some codes using formulas to visualize the features, and our goal is to create algorithms to evaluate balance. However, these codes are only half of our project. We plan to further refine them into a final version within the given timeline.

Here are the mid-reports of software engineering students. To develop the GUI, we decided to use the tkinter lib, which we mentioned before in our report, and to develop it over it. Even though the GUI side of the project is still under development, our aim will be to call the scripts prepared by biomedical students with certain functions and visualize them in our ui. you can see the code part

5. SUMMARY AND CONCLUSION

During the aging process, many changes are observed in the human body. Posture, balance and the walking parameters are also among those that have Decayed

In summary, we are creating a system in which we combine both electromyography and Matlab data.based balance analysis to create an integrated system in which balance disorders can be easily diagnosed,and to make such postural balance diagnostic tools more accessible and more streamlined.simple and straightforward GUI to help technicians analyze balance disorders.

In conclusion, our software engineering team has successfully developed a comprehensive Center of Pressure (COP) feature extraction tool as part of our capstone project, collaborating closely with biomedical engineers to meet the project objectives. Leveraging the capabilities of MATLAB, we effectively processed and analyzed the data obtained from the measurements provided by our biomedical engineering colleagues. Through the use of advanced signal processing techniques and algorithms, we were able to extract crucial COP features, such as sway amplitude, velocity, and area, which are essential for assessing human balance and stability.

To enhance the usability and visual representation of the extracted COP features, we designed and implemented a user-friendly interface using Python in conjunction with the TKinter library. The interface provides a seamless integration of the MATLAB-processed data, allowing users to conveniently visualize and interpret the extracted features. Through interactive plots and graphical representations, users can gain deeper insights into the underlying patterns and characteristics of the COP data.

Our software solution utilizes the strength of MATLAB's extensive mathematical and plotting functionalities, enabling us to generate accurate and visually appealing representations of the COP measurements. By leveraging MATLAB's capabilities, we were able to implement advanced data processing techniques, including filtering, feature extraction algorithms, and statistical analysis, ensuring the reliability and robustness of the extracted COP features.

Furthermore, the integration of Python with TKinter facilitated the development of a responsive and intuitive user interface. Through the use of TKinter's GUI framework, we designed a visually appealing interface with interactive elements, such as buttons, sliders, and dropdown menus, enabling users to customize and control the visualization of COP features according to their specific requirements. The interface also provides options for exporting the plotted measurements and extracted COP features in various formats, allowing users to easily incorporate the data into their research or clinical workflows.

Our collaborative efforts with the biomedical engineering team ensured that the software solution met the domain-specific requirements and addressed the challenges associated with COP feature extraction. By combining the expertise of software and biomedical engineers, we successfully developed a robust and efficient tool that contributes to the field of biomechanics. This tool empowers researchers and clinicians by providing them with a powerful platform for analyzing human balance and stability, enabling them to make informed decisions and develop effective interventions for patients.

In summary, our capstone project's software engineering team has accomplished the development of a COP feature extraction tool that seamlessly integrates MATLAB for data processing and Python with TKinter for an intuitive user interface. The technical implementation of advanced signal processing techniques, algorithms, and visualization capabilities has resulted in a reliable, flexible, and user-friendly software solution. This tool is expected to have a significant impact on the field of biomechanics, supporting researchers and clinicians in their efforts to analyze COP data and improve the understanding of human balance and stability.

REFERENCES

1. Mishra, P., Kumar, A., Rodrigues, V., Shukla, A. K., & Sundaresan, V. (2016). Feasibility of nuclear ribosomal region ITS1 over ITS2 in barcoding taxonomically challenging genera of subtribe Cassiinae (Fabaceae). In PeerJ (Vol. 4, p. e2638). PeerJ. https://doi.org/10.7717/peerj.2638

2.https://www.researchgate.net/figure/Center-of-pressure-standard-deviation-CoP-SD-in-the-medialateral-A-C-and_fig2_51507192

3.https://www.jstage.jst.go.jp/article/ijshs/5/0/5_0_71/_pdf

4.https://www.tekscan.com/resources/ebook/pressure-measurement-in-gait-lab-why-you-need-it?utm_source=google&utm_medium=cpc&utm_term=gait+analysis&utm_content=ra2&utm_campaign=medical&gclid=CjwKCAiAwomeBhBWEiwAM43YILdrl_Yg21R6gZpHepBpG9qOYbgtLdo7ozk7MAbCR8m_8zSId3pnWBoC8oAQAvD_BwE

5..https://unfccc.int/process/bodies/supreme-bodies/conference-of-the-parties-cop

6.https://www.researchgate.net/figure/A-exemplary-plots-of-the-center-of-pressure-COP-at-the-anterior-posterior-ap_fig3_318838624

7.https://www.sciencedirect.com/science/article/abs/pii/S036054422102990X

8.https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8623280/

9.https://www.econstor.eu/bitstream/10419/173238/1/full-05.pdf

10. https://www.sciencedirect.com/science/article/abs/pii/S0966636208000301

11. https://www.sciencedirect.com/science/article/abs/pii/S0966636213007030

12.https://interestingliterature.com/2022/07/o-henry-cop-anthem-summary-analysis/

13.https://valdperformance.com/forcedecks/?utm_term=force%20plate%20analysis&utm_campaign=Force+Plate+-+ASIA&utm_source=adwords&utm_medium=ppc&hsa_acc=5184783684&hsa_cam=13597454580&hsa_grp=126399894600&hsa_ad=528815912370&hsa_src=g&hsa_tgt=kwd-501804593712&hsa_kw=force%20plate%20analysis&hsa_mt=e&hsa_net=adwords&hsa_ver=3&gclid=CjwKCAiAwomeBhBWEiwAM43YIBLenz2KOT-QJGH8uRVvwYSLWc2hOFWxN50sSVstya4cZdfH57eLDxoCicMQAvD_BwE

14. https://belgelendirme.ctr.com.tr/iso-13485-nedir.html

15.https://tse.org.tr/Icerik/DuyuruDetay?DuyuruID=4313

16. https://www.sciencedirect.com/journal/gait-and-posture

17.http://tip.baskent.edu.tr/kw/upload/464/dosyalar/cg/sempozyum/ogrsmpzsnm13/13.P14.pdf

18.http://www.kidssportsacademy.com/?pnum=16&pt=Flamingo+Denge+Testi

19.http://www.ee.hacettepe.edu.tr/~solen/Matlab/Coskun%20Tasdemir'den/Matlab%20Turkce%20kullanma%20kilavuzu.pdf

20. https://erdincuzun.com/ileri-python/tkinter-gui/

21.https://docs.python.org/3/library/tkinter.html

22.https://realpython.com/python-gui-tkinter/

23.https://www.pythontutorial.net/tkinter/tkinter-matplotlib/

24.https://pythonprogramming.net/how-to-embed-matplotlib-graph-tkinter-gui/

25.https://pythonprogramming.net/tkinter-depth-tutorial-making-actual-program/

26.https://datatofish.com/matplotlib-charts-tkinter-gui/

27.https://www.mdpi.com/1424-8220/19/13/2974

1 fig : Walendziuk, W. (2014). Measurement Uncertainty Analysis of the Strain Gauge Based Stabilographic Platform. Acta Mechanica et Automatica, 8, 74 - 78.

Fig.1.3.2.1 Gageler, William & Wearing, Scott & James, Daniel. (2015). Automatic jump detection method for athlete monitoring and performance in volleyball. International Journal of Performance Analysis in Sport. 15. 10.1080/24748668.2015.11868793.

Fig.1.3.2.2 Yazji, Mouna & Raison, Maxime & Aubin, Carl-Eric & Labelle, Hubert & Detrembleur, Christine & Mahaudens, Philippe & Mousny, Marilyne. (2015). Are the mediolateral joint forces in the lower limbs different between scoliotic and healthy subjects during gait?. Scoliosis. 10. S3. 10.1186/1748-7161-10-S2-S3.

Fig.1.3.2.3 Huang, T.-C., Huang, H.-P., Wu, K.-W., Pao, J.-L., Chen, C.-K., Wang, T.-M., & Lu, T.-W. (2022). Body's Center of Mass Motion Relative to the Center of Pressure during Gait, and Its Correlation with Standing Balance in Patients with Lumbar Spondylosis. Applied Sciences, 12(24), 12915. https://doi.org/10.3390/app122412915

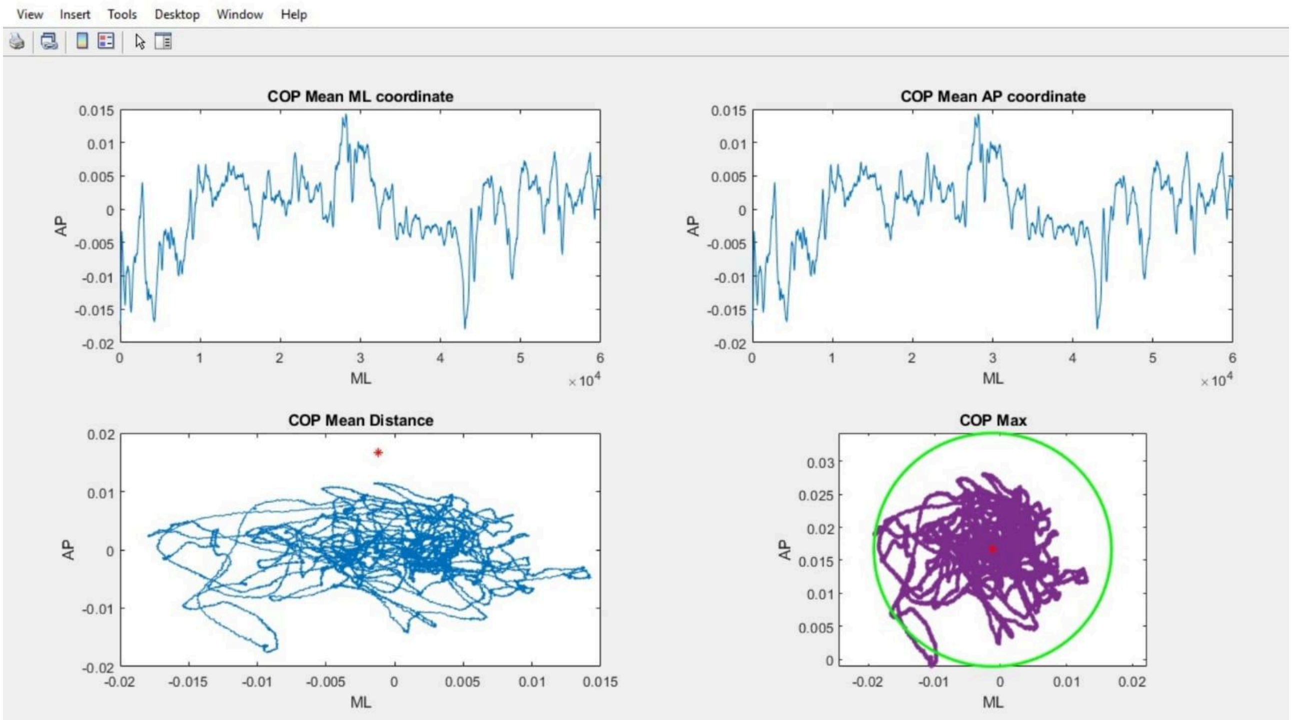
Fig. 1.3.2.4: Tao, Juan & Dong, Ming & Li, Li & Wang, Chunfeng & Li, Jing & Liu, Yue & Bao, Rongrong & Pan, Caofeng. (2020). Real-time pressure mapping smart insole system based on a controllable vertical pore dielectric layer. Microsystems & Nanoengineering. 6. 10.1038/s41378-020-0171-1.

Use the **IEEE style** when listing references. Try to add MINIMUM 10 references.
A good guide can be found here: http://libguides.murdoch.edu.au/IEEE/,
and many examples here: https://libguides.murdoch.edu.au/IEEE/all

Appendix

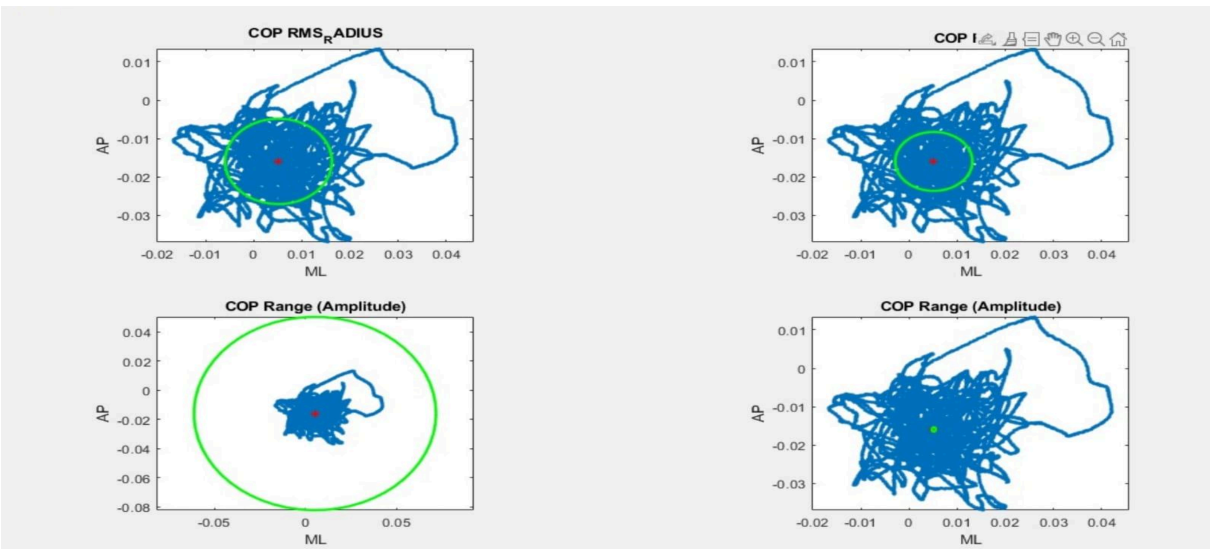
MATLAB Code :

```
% Step 1 and 2: COP Mean ML coordinates
load matlab.mat
ML= COPx;
AP= COPy;
Xn1 = ML - mean(ML);
Xn2 = ML - (sum(ML) / length(ML));
subplot(2,2,1)
plot(Xn1);
xlabel('ML');
ylabel('AP');
title('COP Mean ML coordinate');
subplot(2,2,2)
plot(Xn2);
xlabel('ML');
ylabel('AP');
title('COP Mean AP coordinate');
% STEP 3
Yn = AP - mean(AP);
Xn = ML - mean(ML);
COV = sum(Xn.*Yn) / length(Yn);
Rn = sqrt(Xn.^2 + Yn.^2);
subplot(2, 2, 3)
plot(Xn, Yn);
hold on;
plot(mean(ML), mean(AP), 'r*');
theta = linspace(0, 2*pi, 100);
x = COV * cos(theta) + mean(ML);
y = COV * sin(theta) + mean(AP);
plot(x, y, 'g', 'LineWidth', 2);
xlabel('ML');
ylabel('AP');
title('COP Mean Distance');
% Step 4: COP Max
mean_ML = mean(ML);
mean_AP = mean(AP);
max_ML = max(abs(ML - mean_ML));
max_AP = max(abs(AP - mean_AP));
subplot(2, 2, 4)
plot(ML, AP, '.');
hold on;
plot(mean_ML, mean_AP, 'r*');
theta = linspace(0, 2*pi, 100);
x = mean_ML + max_ML * cos(theta);
y = mean_AP + max_AP * sin(theta);
plot(x, y, 'g', 'LineWidth', 2);
axis equal;
xlabel('ML');
ylabel('AP');
title('COP Max')
```



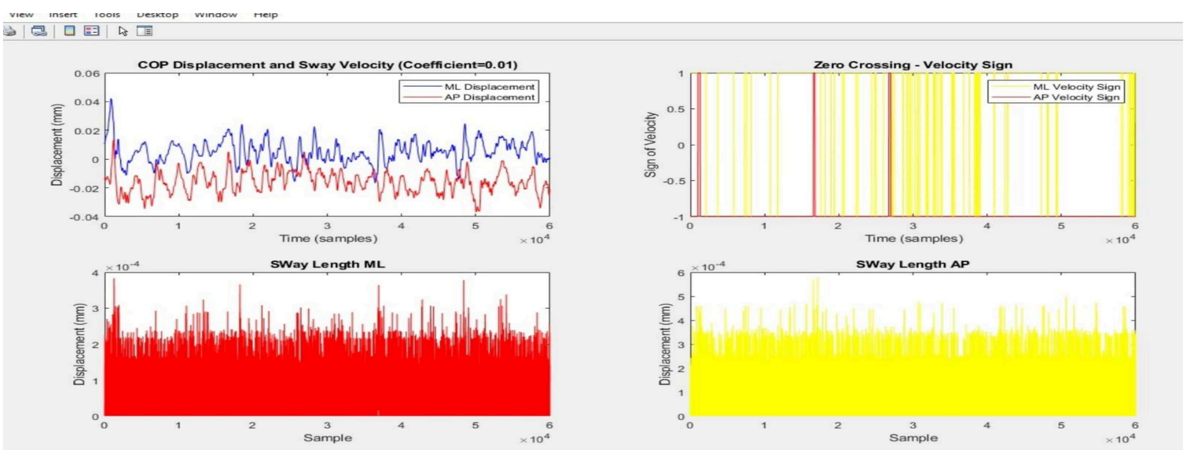

```
% Step 5: COP RMS
% COPx and COPy data from
ML = COPx;
AP = COPy;
% ML and AP
mean_ML = mean(ML);
mean_AP = mean(AP);
% RMS (Root Mean Square) values in ML and AP directions
RMS_ML = sqrt(mean((ML - mean_ML).^2));
RMS_AP = sqrt(mean((AP - mean_AP).^2));
% RMS value of the radius
RMS_RADIUS = sqrt(mean((sqrt((ML - mean_ML).^2 + (AP - mean_AP).^2)).^2));

% Plot COP RMS RADIUS
subplot(2,2,1)
plot(ML, AP, '.');
hold on;
plot(mean_ML, mean_AP, 'r*');
theta = linspace(0, 2*pi, 100);
x = mean_ML + RMS_RADIUS* cos(theta);
y = mean_AP + RMS_RADIUS* sin(theta);
plot(x, y, 'g', 'LineWidth', 2);
axis equal;
xlabel('ML');
ylabel('AP');
title('COP RMS_RADIUS');

% Plot COP RMS
subplot(2,2,2)
plot(ML, AP, '.');
hold on;
plot(mean_ML, mean_AP, 'r*');
theta = linspace(0, 2*pi, 100);
x = mean_ML + RMS_ML * cos(theta);
y = mean_AP + RMS_AP * sin(theta);
```

```
plot(x, y, 'g', 'LineWidth', 2);
axis equal;
xlabel('ML');
ylabel('AP');
title('COP RMS');
% Step 6: COP Range (Amplitude)
mean_ML = mean(COPx);
mean_AP = mean(COPy);
max_radius = max(abs(COPx - mean_ML)) + max(abs(COPy - mean_AP));
subplot(2,2,3)
plot(COPx, COPy, '.');
hold on;
plot(mean_ML, mean_AP, 'r*');
theta = linspace(0, 2*pi, 60);
x = mean_ML + max_radius * cos(theta);
y = mean_AP + max_radius * sin(theta);
plot(x, y, 'g', 'LineWidth', 2);
axis equal;
xlabel('ML');
ylabel('AP');
title('COP Range (Amplitude)');
% step 8: COP Range (Amplitude)
ML= COPx;
AP= COPy;
mean_ML = mean(ML);
mean_AP = mean(AP);
max_diff_ML = max(abs(diff(COPx)));
max_diff_AP = max(abs(diff(COPy)));
subplot(2,2,4)
plot(ML, AP, '.');
hold on;
plot(mean_ML, mean_AP, 'r*');
theta = linspace(0, 2*pi, 100);
x = mean_ML + max_diff_ML * cos(theta);
y = mean_AP + max_diff_AP * sin(theta);
plot(x, y, 'g', 'LineWidth', 2);
axis equal;
xlabel('ML');
ylabel('AP');
title('COP Range (Amplitude)')
```

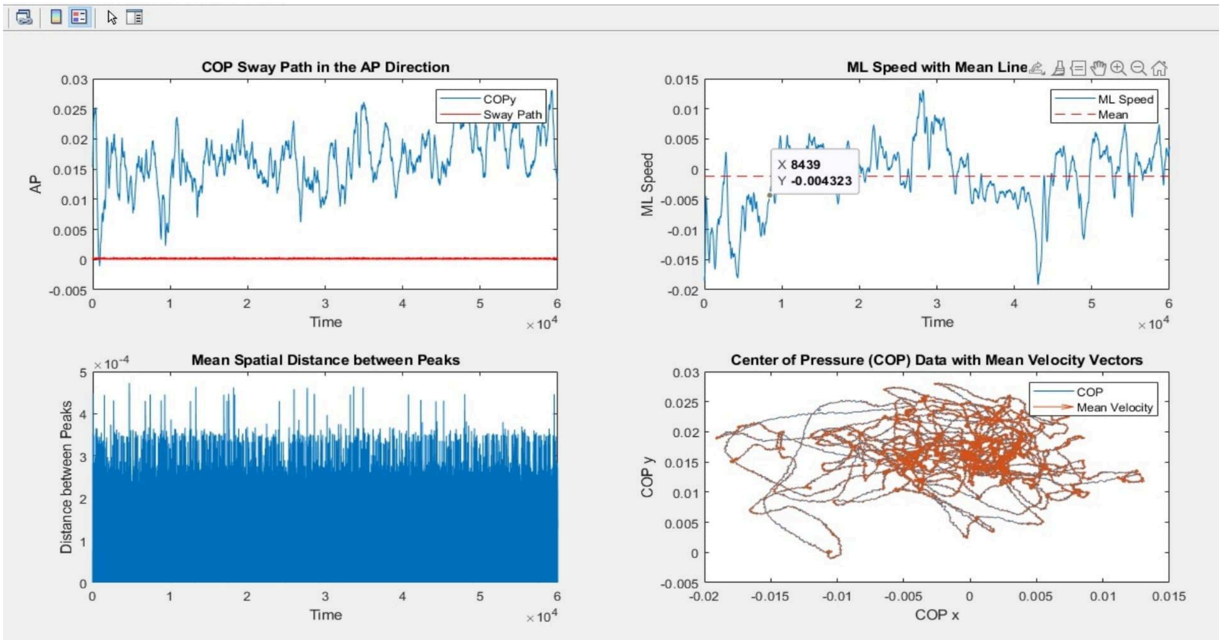



```
ML = COPx;
```

```
AP = COPy;
var_vel_AP = var(diff(COPx));
var_vel_ML = var(diff(COPy));
mean_AP = mean(abs(COPy));
sway_vel_coeff = sqrt((var_vel_ML + var_vel_AP) / (mean_AP^2));
subplot(2,2,1);
plot(1:length(COPx), COPx, 'b');
hold on;
plot(1:length(COPy), COPy, 'r');
xlabel('Time (samples)');
ylabel('Displacement (mm)');
title(sprintf('COP Displacement and Sway Velocity (Coefficient=%.2f)',
sway_vel_coeff));
legend('ML Displacement', 'AP Displacement');
% Zero Crossing ML
zero_cross_ML = sum(diff(sign(ML)) ~= 0);
% Zero Crossing AP
zero_cross_AP = sum(diff(sign(AP)) ~= 0);
subplot(2,2,2);
plot(sign(ML), 'y');
hold on;
plot(sign(AP), 'r');
xlabel('Time (samples)');
ylabel('Sign of Velocity');
title('Zero Crossing - Velocity Sign');
legend('ML Velocity Sign', 'AP Velocity Sign');
% SWay Length ML
sway_length_ML = sum(abs(diff(COPx)));
% SWay Length AP
sway_length_AP = sum(abs(diff(COPy)));
subplot(2,2,3);
plot(1:length(ML)-1, abs(diff(ML)), 'r');
xlabel('Sample');
ylabel('Displacement (mm)');
title('SWay Length ML');
subplot(2,2,4);
plot(1:length(AP)-1, abs(diff(AP)), 'y');
xlabel('Sample');
ylabel('Displacement (mm)');
title('SWay Length AP');
```

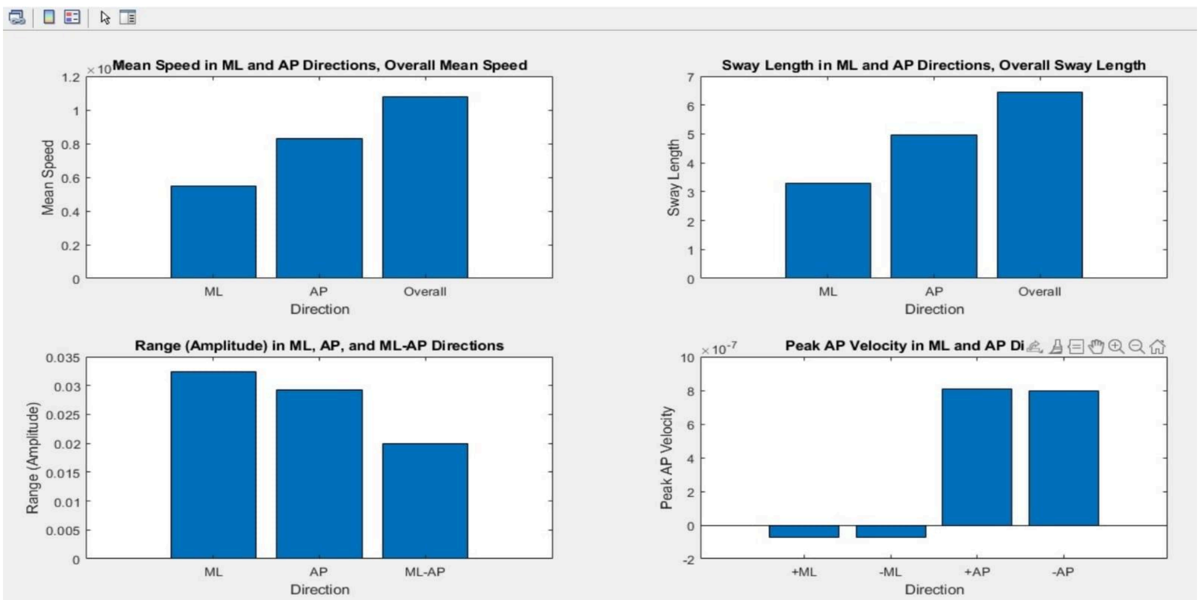


```
% STEP 13 sway length
SWAY_LENGTH = sum(sqrt(diff(COPx).^2 + diff(COPy).^2));
% COPy sway path in the AP direction
subplot(2,2,1)
plot(COPy);
hold on;
plot(1:length(COPy)-1, abs(diff(COPy)), 'r');
xlabel('Time');
ylabel('AP');
title('COP Sway Path in the AP Direction');
legend('COPy', 'Sway Path');
% STEP 14
ML_STD_SPD = sqrt(mean((COPx - mean(COPx)).^2));
% Plot ML speed
subplot(2,2,2)
plot(COPx);
hold on;
plot(1:length(COPx), mean(COPx)*ones(size(COPx)), 'r--');
xlabel('Time');
ylabel('ML Speed');
title('ML Speed with Mean Line');
legend('ML Speed', 'Mean');
% Step 15:
% Calculation of mean spatial distance between peaks
dist_peak_SD = sqrt((diff(COPx).^2) + (diff(COPy).^2));
mean_dist_peak_SD = mean(dist_peak_SD);
subplot(2,2,3)
plot(dist_peak_SD);
xlabel('Time');
ylabel('Distance between Peaks');
title('Mean Spatial Distance between Peaks');
% velocity in the ML direction
% Step 16
velML = diff(COPx);
meanVelML = mean(velML);
% velocity in the AP direction
velAP = diff(COPy);
meanVelAP = mean(velAP);
meanVel = sqrt(meanVelML^2 + meanVelAP^2);

% COP mean velocity vectors
subplot(2,2,4)
plot(COPx, COPy);
hold on;
quiver(COPx(1:end-1), COPy(1:end-1), velML, velAP);
hold off;
xlabel('COP x');
ylabel('COP y');
title('Center of Pressure (COP) Data with Mean Velocity Vectors');
legend('COP', 'Mean Velocity');
```



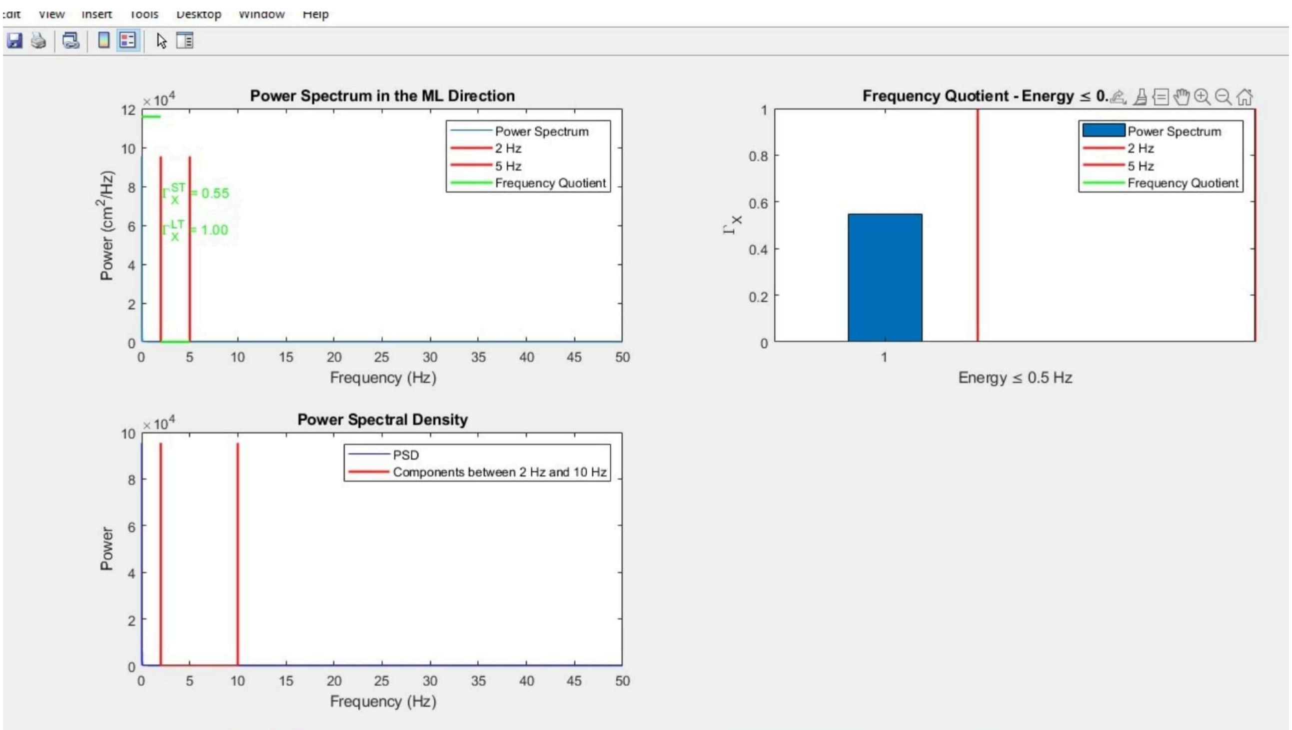

```
% 17 step
% speed in ML direction
MEAN_SPD_ML = mean(abs(diff(COPx)));
% speed in AP direction
MEAN_SPD_AP = mean(abs(diff(COPy)));
% mean speed
MEAN_SPD = mean(sqrt(diff(COPx).^2 + diff(COPy).^2));
% Create a bar plot for mean speed
subplot(2, 2, 1);
bar([MEAN_SPD_ML, MEAN_SPD_AP, MEAN_SPD]);
xticks(1:3);
xticklabels({'ML', 'AP', 'Overall'});
xlabel('Direction');
ylabel('Mean Speed');
title('Mean Speed in ML and AP Directions, Overall Mean Speed');
% step 20
% sway length in ML direction
SWAY_LENGTH_ML = sum(abs(diff(COPx)));
% sway length in AP direction
SWAY_LENGTH_AP = sum(abs(diff(COPy)));
% sway length
SWAY_LENGTH = sum(sqrt(diff(COPx).^2 + diff(COPy).^2));
% plot for sway length
subplot(2, 2, 2);
bar([SWAY_LENGTH_ML, SWAY_LENGTH_AP, SWAY_LENGTH]);
xticks(1:3);
xticklabels({'ML', 'AP', 'Overall'});
xlabel('Direction');
ylabel('Sway Length');
title('Sway Length in ML and AP Directions, Overall Sway Length');
hold on;
% step 21
% Range (Amplitude) in ML direction
RANGE_ML = max(COPx) - min(COPx);
% Range (Amplitude) in AP direction
RANGE_AP = max(COPy) - min(COPy);
% Range (Amplitude) in ML-AP direction
```

```
RANGE_ML_AP = max(sqrt((COPx - mean(COPx)).^2 + (COPy - mean(COPy)).^2));
% Create a bar plot for range (amplitude)
subplot(2, 2, 3);
bar([RANGE_ML, RANGE_AP, RANGE_ML_AP]);
xticks(1:3);
xticklabels({'ML', 'AP', 'ML-AP'});
xlabel('Direction');
ylabel('Range (Amplitude)');
title('Range (Amplitude) in ML, AP, and ML-AP Directions');
% step 20
% Positive peaks in ML axis (displacements to the right)
positive_peaks_ML = find(diff(COPx) > 0);
% Calculate the peak AP velocity in the +ML direction
peak_AP_vel_positive_ML = mean(diff(COPy(positive_peaks_ML)));
% N peaks in ML axis (displacements to the left)
negative_peaks_ML = find(diff(COPx) < 0);
% peak AP velocity in the -ML direction
peak_AP_vel_negative_ML = mean(diff(COPy(negative_peaks_ML)));
% P peaks in AP axis (displacements forward)
positive_peaks_AP = find(diff(COPy) > 0);
% peak AP velocity in the +AP direction
peak_AP_vel_positive_AP = mean(diff(COPx(positive_peaks_AP)));
% Negative peaks in AP axis (displacements backward)
negative_peaks_AP = find(diff(COPy) < 0);
% the peak AP velocity in the -AP direction
peak_AP_vel_negative_AP = mean(diff(COPx(negative_peaks_AP)));
% bar plot for peak AP velocity
subplot(2,2,4)
bar([peak_AP_vel_positive_ML, peak_AP_vel_negative_ML, peak_AP_vel_positive_AP,
peak_AP_vel_negative_AP]);
xticks(1:4);
xticklabels({'+ML', '-ML', '+AP', '-AP'});
xlabel('Direction');
ylabel('Peak AP Velocity');
title('Peak AP Velocity in ML and AP Directions');
```



```
%step 21
% Define the frequency bands
f_inf = 0;
f_2 = 2;
f_5 = 5;
Fs = 100;
ML_fft = fft(COPx);
ML_power = abs(ML_fft).^2;
ML_power_half = ML_power(1:floor(length(ML_power)/2)+1);
ML_freq_half = linspace(0, Fs/2, length(ML_power_half));
Delta_t_c = 20;
ML_ST_power = ML_power_half(ML_freq_half <= f_2);
ML_LT_power = ML_power_half(ML_freq_half > f_2);
ML_LT_power = ML_LT_power(ML_freq_half(ML_freq_half > f_2) <= f_5);
Hs = sum(ML_ST_power);
Hl = sum(ML_LT_power);
Gamma_X_ST = sum(ML_power_half(f_inf < ML_freq_half & ML_freq_half <= f_2)) / Hs;
Gamma_X_LT = sum(ML_power_half(f_2 < ML_freq_half & ML_freq_half <= f_5)) / Hl;
% Plot the power spectrum in the ML direction
subplot(2,2,1);
plot(ML_freq_half, ML_power_half);
xlabel('Frequency (Hz)');
ylabel('Power (cm^2/Hz)');
title('Power Spectrum in the ML Direction');
hold on;
line([f_2 f_2], [0 max(ML_power_half)], 'Color', 'r', 'LineWidth', 1.5);
line([f_5 f_5], [0 max(ML_power_half)], 'Color', 'r', 'LineWidth', 1.5);
line([f_inf f_2], [Gamma_X_ST*Hs Gamma_X_ST*Hs], 'Color', 'g', 'LineWidth', 1.5);
line([f_2 f_5], [Gamma_X_LT*Hl Gamma_X_LT*Hl], 'Color', 'g', 'LineWidth', 1.5);
text(f_2+0.1, max(ML_power_half)*0.8, sprintf('\\Gamma_X^{ST} =
%.2f',Gamma_X_ST),'Color', 'g');
text(f_2+0.1, max(ML_power_half)*0.6, sprintf('\\Gamma_X^{LT} =
%.2f',Gamma_X_LT),'Color', 'g');
legend('Power Spectrum', '2 Hz', '5 Hz', 'Frequency Quotient');

% Step 22: Energy ≤ 0.5 Hz ML
finf = 0;
fsup = 0.5;
indices = find(ML_freq_half > finf & ML_freq_half <= fsup);
Gamma_X_0_5 = sum(ML_power_half(indices)) / sum(ML_power_half);
% Plot the energy ≤ 0.5 Hz in the ML direction
subplot(2,2,2);
bar(1, Gamma_X_0_5);
ylim([0 1]);
xlabel('Energy ≤ 0.5 Hz');
ylabel('\Gamma_X');
title('Frequency Quotient - Energy ≤ 0.5 Hz ML');
hold on;
line([f_2 f_2], [0 max(ML_power_half)], 'Color', 'r', 'LineWidth', 1.5);
line([f_5 f_5], [0 max(ML_power_half)], 'Color', 'r', 'LineWidth', 1.5);
line([f_inf f_2], [Gamma_X_ST * Hs Gamma_X_ST * Hs], 'Color', 'g', 'LineWidth',
1.5);
line([f_2 f_5], [Gamma_X_LT * Hl Gamma_X_LT * Hl], 'Color', 'g', 'LineWidth',
1.5);
text(f_2+0.1, max(ML_power_half)*0.8, sprintf('\\Gamma_X^{ST} =
%.2f',Gamma_X_ST),'Color', 'g');
text(f_2+0.1, max(ML_power_half)*0.6, sprintf('\\Gamma_X^{LT} =
%.2f',Gamma_X_LT),'Color', 'g');
```

```
legend('Power Spectrum', '2 Hz', '5 Hz', 'Frequency Quotient');

%step 23
Fs = 100;
ML_fft = fft(COPx);
ML_power = abs(ML_fft).^2;
ML_power_half = ML_power(1:floor(length(ML_power)/2)+1);
ML_freq_half = linspace(0, Fs/2, length(ML_power_half));
f_inf = 2;
f_sup = 10;
indices = find(ML_freq_half > f_inf & ML_freq_half <= f_sup);
Gamma_X_0_5 = sum(ML_power_half(indices)) / sum(ML_power_half);
subplot(2,2,3)
plot(ML_freq_half, ML_power_half, 'b');
xlabel('Frequency (Hz)');
ylabel('Power');
title('Power Spectral Density');
hold on;
line([f_inf f_inf], [0 max(ML_power_half)], 'Color', 'r', 'LineWidth', 1.5);
line([f_sup f_sup], [0 max(ML_power_half)], 'Color', 'r', 'LineWidth', 1.5);
plot(ML_freq_half(indices), ML_power_half(indices), 'r');
legend('PSD', 'Components between 2 Hz and 10 Hz');
```



Python Code:

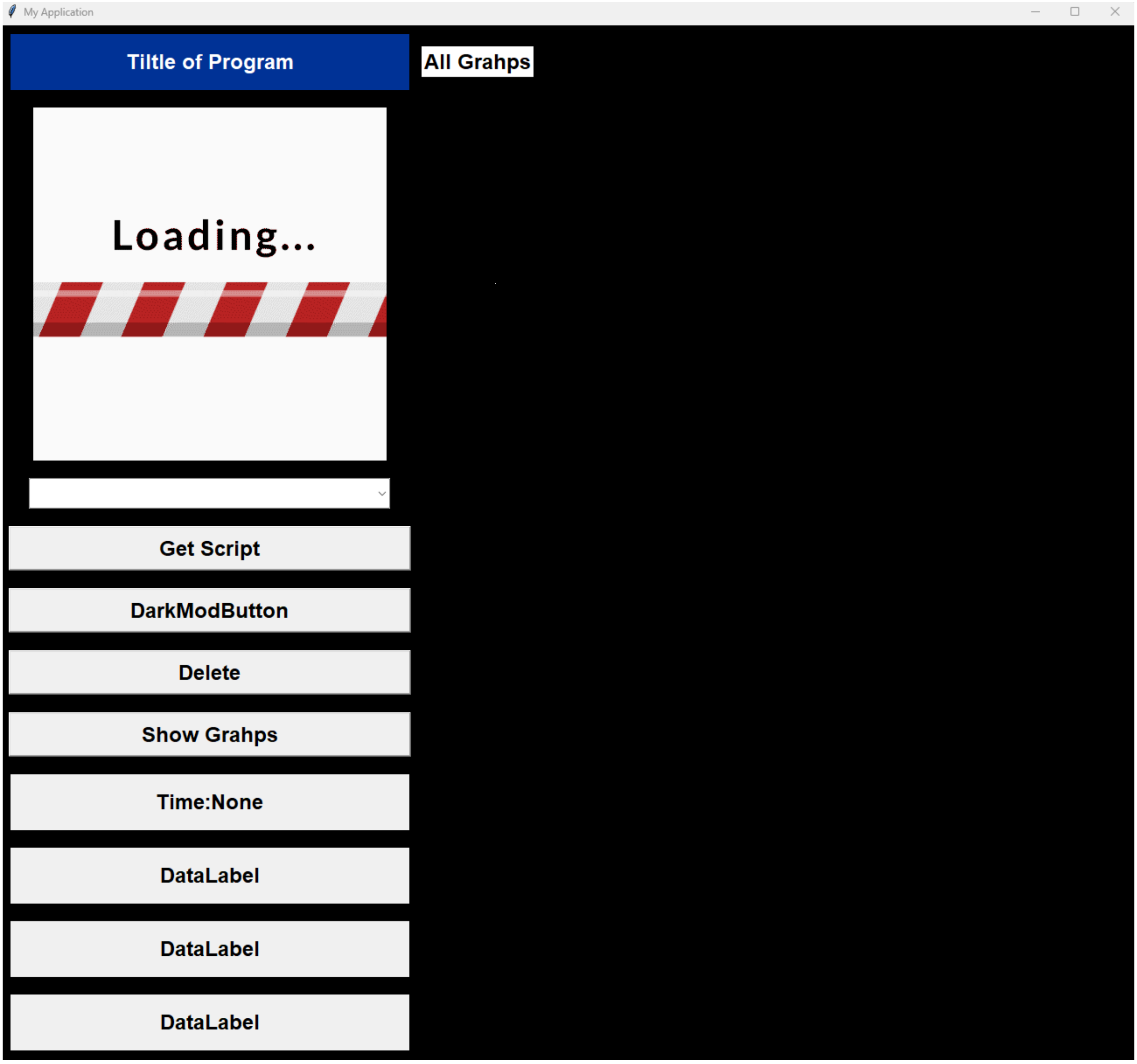
```python
import tkinter as tk
import os
from tkinter import ttk
import matlab.engine
import matplotlib.pyplot as plt
from PIL import Image, ImageTk
import tkinter.messagebox
import time
#Colors
primary_color = "#003399"
secondary_color = "#66B2FF"
light_gray_color = "#E6E6E6"
white_color = "#FFFFFF"
light_green_color = "#66CC99"
accent_color = "#FF9900"
dark_gray_color = "#666666"
error_color = "#CC0000"


mode=False
x=0
maxData=None
minData=None
meanData=None
start_time=None
end_time=None
execution_time=None
eng = matlab.engine.start_matlab()
image_paths = []
def addArray():
    global image_paths
    print('Resimler Eklendi')
    klasor_yolu = 'C:\\Users\\yagiz\\OneDrive\\Documents\\GitHub\\Python'
    dosyalar = os.listdir(klasor_yolu)
    for dosya in dosyalar:
     dosya_yolu = os.path.join(klasor_yolu, dosya)
     if dosya.endswith(( '.jpeg', '.png', '.gif')):
        if dosya in image_paths:
```

```python
        if dosya in image_paths:
            print('Exist')
        else:
            image_paths.append(dosya)
            print(f"{dosya} Eklendi.")
    print(image_paths)
    row = 0
    column = 0
    for image_path in image_paths:
     image = Image.open(image_path)
     image = image.resize((250, 250))
     photo = ImageTk.PhotoImage(image)

     label = tk.Label(image_frame, image=photo)
     label.image = photo
     label.grid(row=row, column=column)

     column += 1
     if column == 3:
        column = 0
        row += 1
def removeImage():
    klasor_yolu = 'C:\\Users\\yagiz\\OneDrive\\Documents\\GitHub\\Python'
    dosyalar = os.listdir(klasor_yolu)
    for dosya in dosyalar:
     dosya_yolu = os.path.join(klasor_yolu, dosya)
     if dosya.endswith(('.jpg', '.jpeg', '.png', '.gif')):
        os.remove(dosya_yolu)
        print(f"{dosya} Deleted.")
class Option:
    def __init__(self, value, label):
        self.value = value
        self.label = label

    def __str__(self):
        return self.label
```

```
def on_select(event):
    global x
    selected_item = combobox.get()
    option = next((opt for opt in options if str(opt) == selected_item), None)
    if option:
        print("Seçilen öğe:", option.value)
        x=option.value
        print(x)
    else:
        print("Null")

def darkmodSwicth():
   print('Theme Change')
   global mode
   if mode==False:
    myTiltle.config(fg='white',bg='#003399')
    form.configure(bg='black')
    mode=True
   else:
    myTiltle.config(fg='black',bg=primary_color)
    form.configure(bg='white')
    mode=False

def getScript():
      global start_time
      global end_time
      global execution_time
      global maxData
      start_time=time.time()
      if x == 0:
          tkinter.messagebox.showinfo("Alert.",  "Script not selected")
      elif x==1:
          script_path = "script1.png"
          eng.eval("run('step1.m')", nargout=0)
          maxData=eng.workspace['MaxCopX']
          meanData=eng.workspace['MaxCopY']
          if not os.path.exists(script_path):
```

```
options = [
    Option(1, "COP Mean ML Coordinate"),
    Option(2, "COP Mean AP Coordinate"),
    Option(3, "COP Mean Distance"),
    Option(4, "COP Max"),
    Option(5, "COP RMS"),
    Option(6, "COP Range (Amplitude)"),
    Option(7, "COP and Sway Area"),
    Option(8, "Sway Direction Coeff"),
    Option(9, "Maximum Displacement"),
    Option(10, "COP Displacement and Sway Velocity"),
    Option(11, "COP Max Signal"),
    Option(12, "Sway Length ML"),
    Option(13, "Sway Length AP"),
    Option(14, "ML Time Series"),
    Option(15, "Mean Spatial Distance between Peaks"),
    Option(16, "Center of Pressure (COP) Data with Mean Velocity Vectors"),
    Option(17, "Cop"),
    Option(18, "Peak AP Velocity in Different Directions"),
    Option(19, "Mean Speed in ML and AP Directions"),
    Option(10, "Sway Length in ML and AP Directions,Overall Sway Length"),
    Option(21, "Center of Pressure (COP)"),
    Option(22, "Power Spectrum in the ML Direction"),
    Option(23, "Frequency Quotient - Energy ≤ 0.5 Hz ML"),
    Option(24, "Power Spectral Density"),
]
```

```
# GUI
form = tk.Tk()
form.title("My Application")
form.configure(bg=light_gray_color)
form.geometry('1300x1200')
image_frame = tk.Frame(form)
image = tk.PhotoImage(file='.idea/loading_gif.gif')
myTiltle=tk.Label(form,text="Tiltle of Program",width=30,height=2,font='Helvetica 18 bold',fg='Black',bg=primary_color)
timeLabel=tk.Label(form,text='Time:'+str(execution_time),width=30,height=2,font='Helvetica 18 bold',fg='Black')
maxLabel=tk.Label(form,text= 'DataLabel' ,width=30,height=2,font='Helvetica 18 bold',fg='Black')
minLabel=tk.Label(form,text= 'DataLabel', width=30,height=2,font='Helvetica 18 bold',fg='Black')
meanLabel=tk.Label(form,text= 'DataLabel',width=30,height=2,font='Helvetica 18 bold',fg='Black')
myButton_1= tk.Button(form,text='Get Script',width=30,height=1,font='Helvetica 18 bold',fg='Black',command=getScript)
myButton_3= tk.Button(form,text='Delete',width=30,height=1,font='Helvetica 18 bold',fg='Black',command=removeImage)
myButton_4= tk.Button(form,text='Show Grahps',width=30,height=1,font='Helvetica 18 bold',fg='Black',command=addArray)
myButton_2= tk.Button(form,text='DarkModButton',width=30,height=1,font='Helvetica 18 bold',fg='Black',command=darkmodSwicth)
combobox = ttk.Combobox(form, values=[str(opt) for opt in options],width=30,height=20,font='Helvetica 18 bold')
resultLabel = tk.Label(form, image=image,width=400,height=400)
tiltleLabel= tk.Label(form,text='All Grahps',font='Helvetica 18 bold',bg=white_color)
tiltleLabel.grid(column=3,row=0,columnspan=3,pady=10)
myTiltle.grid(column=0,row=0,columnspan=3,pady=10)
timeLabel.grid(column=0,row=7 ,columnspan=3, pady=10)
maxLabel.grid(column=0,row=8 ,columnspan=3, pady=10)
minLabel.grid(column=0,row=9 ,columnspan=3, pady=10)
meanLabel.grid(column=0,row=10 ,columnspan=3, pady=10)
myButton_2.grid(column=0,row=4 ,columnspan=3, pady=10)
myButton_3.grid(column=0,row=5 ,columnspan=3, pady=10)
myButton_4.grid(column=0,row=6 ,columnspan=3, pady=10)
resultLabel.grid(column=0,row=1,columnspan=3,pady=10,padx=40 )
myButton_1.grid(column=0,row=3 ,columnspan=3, pady=10)
image_frame.grid(column=4,row=1,columnspan=3,pady=10)
combobox.grid(column=0,columnspan=3,row=2 , pady=10)
combobox.bind("<<ComboboxSelected>>",  on_select)
# Pencereyi göster
tk.mainloop()
eng.quit()
```


My Application
Tiltle of Program
All Grahps
Loading...
Get Script
DarkModButton
Delete
Show Grahps
Time:None
DataLabel
DataLabel
DataLabel