Interface Design and Development

3. Interface Design and Development

3.1. Introduction

Overview of the Chapter

This chapter delves into the design of a user-friendly web interface, which encapsulates the comprehensive work conducted in this project. The foremost aim is to present complex modelling and simulation concepts of a flexible spacecraft in an easily digestible manner, even for those without a background in MATLAB or detailed knowledge of spacecraft control systems.

As users navigate through the interface, they are introduced to a thorough theoretical exposition of our thesis, including a discussion on the employed control strategies. The interface is crafted for simplicity, enabling users to execute various control simulations and visualize the results easily. Such results illustrate the flexible system's position, velocity, and behavior under each control scenario, offering an empirical lens through which the spacecraft dynamics are understood.

Importance of Interface in Modern Applications

The interface in modern applications is crucial as it bridges the gap between complex technology and user experience. A well-designed interface ensures that users can easily navigate and utilize an application's features, regardless of their technical expertise. It is the interface that dictates the accessibility of an application, making it an essential aspect of software design that directly impacts user satisfaction and engagement. In essence, the interface is the face of the application, inviting users to interact and explore its capabilities.

3.2. Significance of the Interface

Role of Interface in User Interaction

Our interface is the gateway for users to interact with the complex simulations and control methods detailed in our thesis. It simplifies these advanced concepts into an interactive, user-friendly format, enabling users to actively engage with the system, adjust parameters, and visualize the effects of their inputs.

Impact on User Experience

The interface elevates the user experience by offering a straightforward and intuitive platform to navigate the intricate details of our thesis. It ensures that users of all backgrounds can understand and apply the sophisticated simulations and controls, enhancing their overall experience through a design that emphasizes clarity and ease of use.

Relevance to the Thesis

The interface showcases the practical implementation of our research. It combines the techniques we have explored into a unified, approachable platform, allowing users to execute our developed methods. The interface not only demonstrates our work but also underscores its significance in the realm of spacecraft control systems.

Additionally, the interface provides direct access to downloadable PDF files for each segment of our work. These files offer a detailed description of the methods, their application, and the technologies employed, ensuring that users can easily obtain a comprehensive theoretical understanding of our research in a convenient format.

3.3. Interface Conception and Design

The design and conception of our interface are rooted in the goal of presenting our thesis work in a clear and interactive manner. We employed Figma for UI design, ensuring a cohesive and user-friendly layout. The frontend was developed using HTML and CSS, providing structure and style, while the backend was powered by the Python Django framework, with libraries like Matplotlib, NumPy, and Pandas to handle data processing and visualization.

Our interface is divided into four key sections:

Dashboard Section: The dashboard is the welcoming point of our interface, designed to provide a comprehensive overview of our thesis at a glance. It's structured to guide visitors through the project's highlights, objectives, and key findings, offering a centralized hub for all related information.

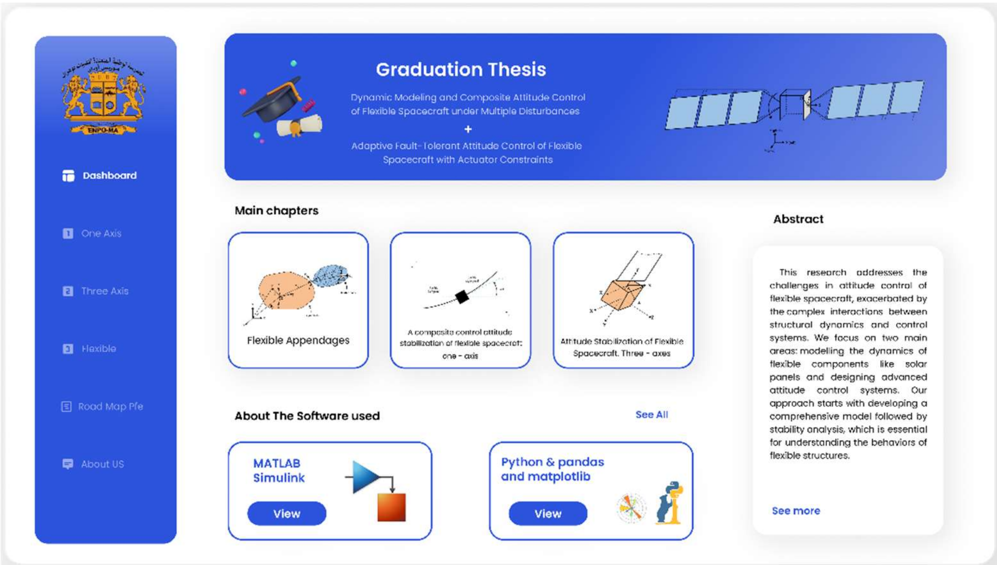

Fig 3.1: The Interface Dashboard page.

Simulation Section: This pivotal section is divided into three distinct parts, each focusing on a different aspect of spacecraft simulation:

One-Axis Simulation, Three-Axis Simulation and Flexible Stability Analysis. Each simulation part is equipped with interactive elements that enable users to run simulations, offering a hands-on experience with the control techniques discussed in our thesis.

About Section: The about section is crafted to communicate the importance of our project and the innovation it brings to the field of spacecraft control. It introduces Our team who have contributed to the project, highlighting their expertise and commitment.

Road map pfe Section: It's our blog section, powered by Notion, serves as a dynamic platform for sharing ongoing updates, insights, and resources related to our thesis work. It's a space where we document our journey, from initial concepts to final implementations, and provide valuable resources such as scripts, tutorials, and theoretical explanations. This section is designed to engage with the community, invite feedback, and foster a collaborative environment.

Each section is crafted to not only reflect the technical precision of our work but also to enhance user engagement and understanding, making our research accessible to a broader audience.

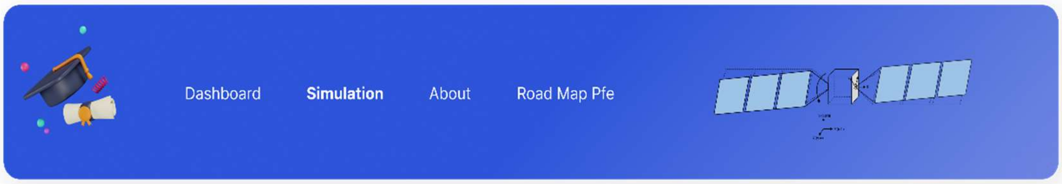

Fig 3.2: The Interface Header page with the four sections.

Fig 3.2 shows the header of the simulation section, which serves as a navigational gateway to the four main sections of our interface.

3.3.1. Objectives of Interface Sections

Section 01 - Dashboard: The dashboard is designed to be the central hub of our interface, providing easy and total access to all its parts. It features an abstract of our thesis work and clickable links to pages describing the technologies used in the simulations, such as MATLAB, SIMULINK, and Python libraries for the backend. The main chapters of our thesis are also accessible here, with simple clickable options to download PDFs for each chapter. On the left side of the dashboard, users can find and navigate to the different sections of the interface, all consolidated on one convenient page.

Section 02 - Simulation: This section's objective is to enable users to run the simulations discussed in our thesis in a user-friendly manner. It is divided into three parts, each representing a different simulation that our interface provides:

- Part 01: Focuses on one-axis attitude control, offering controller designs like PD, PD+DOBC, or PD+ESO, along with disturbance plotting.

- Part 02: Deals with three-axis attitude control, featuring four different control methods including PD controller, sliding mode controller with three modes (sat, sign, tanh), and two methods of adaptive sliding mode.
- Part 03: Conducts stability analysis using a flexible part model and a damping controller model to examine the behavior of flexible stability, complemented by Bode and Nicholas charts for visualization.

Section 03 - About Section: The about section provides a straightforward page that offers information about the thesis work and the team behind it. It also includes a downloadable PDF file that discusses the perspectives and overview of the work, giving visitors a deeper understanding of the project's significance.

Fig 3.3: The about section (Upper part).

Fig 3.3 shows the Upper part of the simulation section, which outlines the objectives of our thesis work and provides detailed information about it.

Section 04 – Road map pfe: The blog page is a Notion page integrated into our interface and shared on the web. It acts as a platform where users can comment, and we can share the roadmap of our thesis work and the various resources associated with it. This section is dedicated to fostering a community around our thesis, encouraging engagement and knowledge sharing.

3.3.2. UI/UX Design

In the realm of web interfaces, the significance of UI/UX design cannot be overstated. It serves as the bridge between humans and machines, facilitating not just the interaction but also the user’s journey through complex processes. For applications that deal with intricate simulations, like those akin to MATLAB Simulink, the UI/UX design becomes even more pivotal. It transforms the daunting task of navigating through dense computational operations into a seamless experience. By prioritizing user-friendliness, the design ensures that even users with minimal background in simulation software can engage with the system effectively, fostering an inclusive environment for learning and exploration.

The objective of this particular design is to make complex spacecraft simulations easy to use and understand. The interface uses simple controls like toggle bars and helpful diagrams to guide users through the simulation process. Users can easily start simulations, see the results as clear graphs, and even compare different methods from the thesis. The design lets users change settings to select the methods they want to use and the graphs they want to see. This makes it possible for anyone to perform advanced simulations without needing to know a lot about how they work.

To achieve the objectives of our user-friendly design, we utilized Figma for the UI design process. Figma is a web-based application specifically crafted for designing user interfaces (UI). It stands out in the design world due to its collaborative nature, allowing multiple users to work on the same project simultaneously, which is particularly beneficial for team-based projects. This feature aligns perfectly with the academic norm of collaborative research and development.

Figma’s advantages are manifold. Its ease of use is one of its most significant benefits; it offers a straightforward interface with no complex menus or toolbars, making it accessible even to those new to design tools1. Moreover, Figma operates on a freemium model, providing core features at no cost, which is ideal for academic settings where budget constraints are common2. The tool also supports real-time collaboration, enabling seamless teamwork and instant feedback, which is crucial when working on a thesis that may involve multiple stakeholders.

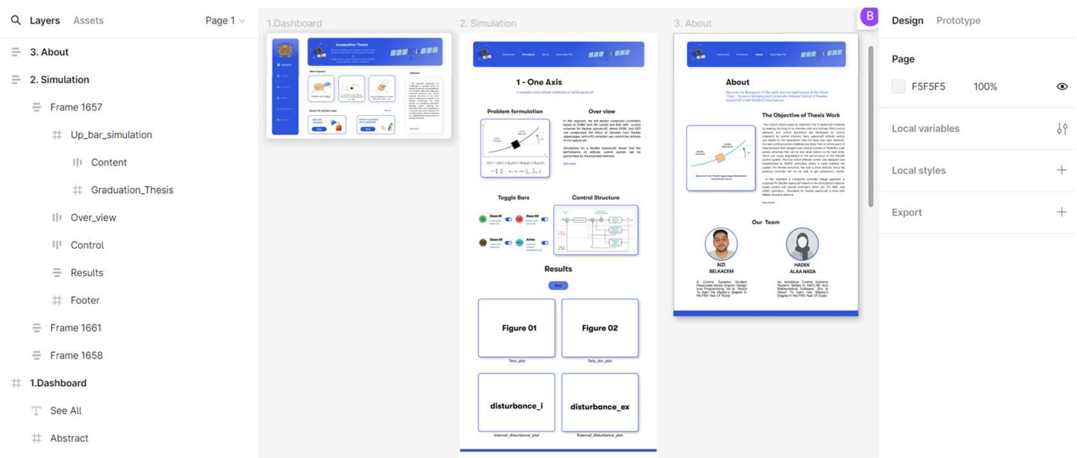

Fig 3.4: Figma design interface.

Fig 3.4 shows the Figma design file for our interface. Which present the design of each section of the interface.

The user interface of our simulation section is thoughtfully divided into three distinct parts, each serving a specific purpose to enhance the user experience and provide a seamless flow from information gathering to simulation execution and results analysis.

- **Information Section**

The topmost section is dedicated to information. Its primary objective is to introduce and define the control approaches used in the simulations. It serves as an educational resource where users can familiarize themselves with the theoretical underpinnings of the control methods implemented. To support this learning process, we can download PDF files detailing our control designs and offer an overview of the work presented. This feature is particularly beneficial for users seeking a deeper understanding of the control strategies before running simulations.

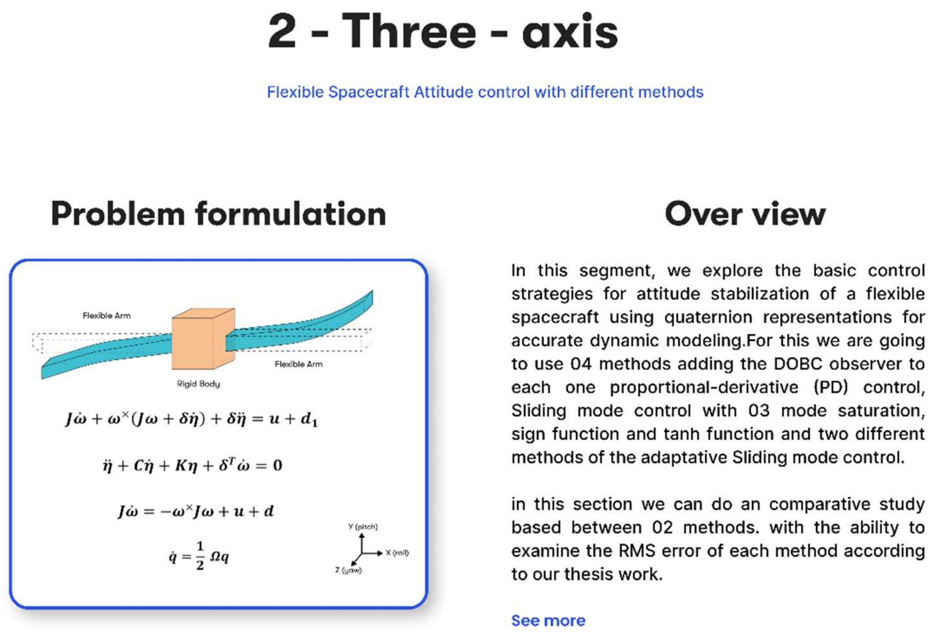

Fig 3.5: The simulation section” three axis”, (information section).

- **Simulation Settings Section**

This middle section is the interactive core of the simulation process, where users can customize their simulation experience. Here, they are presented with various settings that control the outcome possibilities. Users can select control methods, decide whether to include disturbances and choose other parameters relevant to the simulation. This section empowers users to tailor the simulation to their specific needs or research questions, providing a hands-on approach to exploring the behavior of flexible spacecraft under different control scenarios.

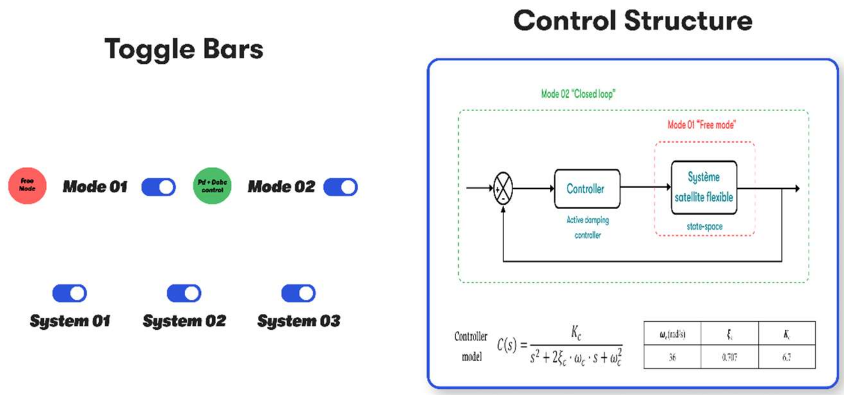

Fig 3.6: The simulation section" Flexible analysis", (simulation settings section).

- **Results Section**

The final section is where the outcomes of the simulations are displayed. After the user runs a simulation with their chosen settings, this area populates with plots generated by Matplotlib, showcasing the results in a clear and interpretable manner. The design of this section prioritizes ease of comparison, allowing users to run different simulations side by side. Whether it's position, velocity, or error metrics, the results are organized to facilitate a comprehensive comparative study, turning complex data into actionable insights.

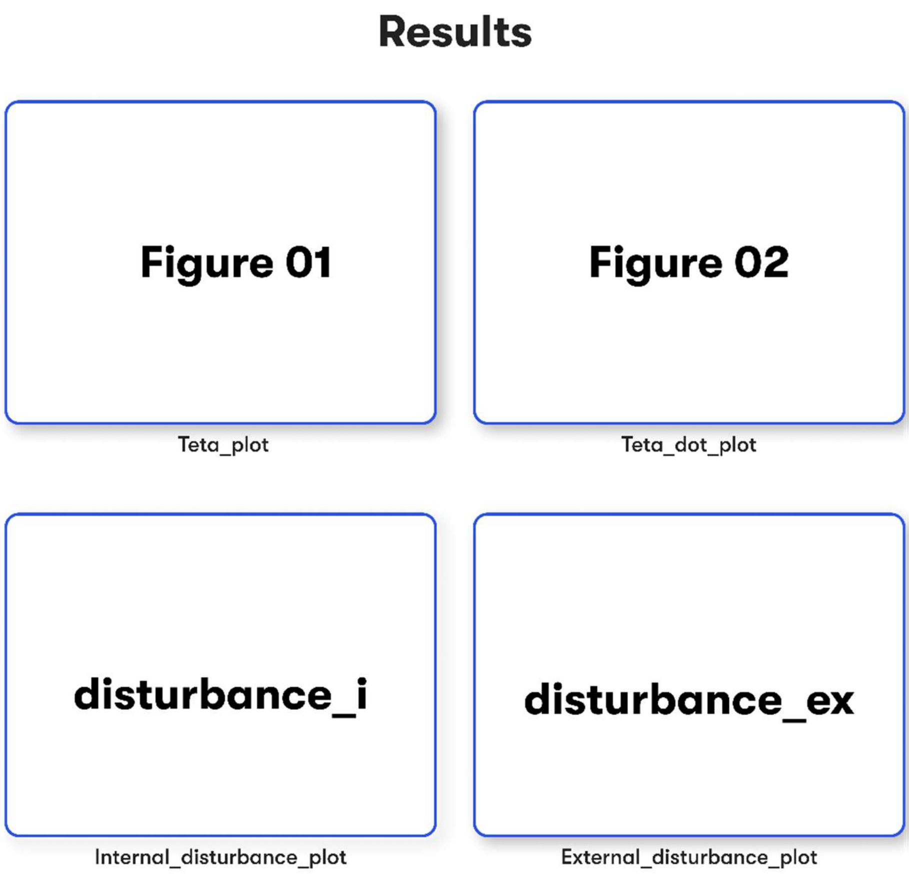

Fig 3.7: The simulation section" one axis", (results section)

Each section is integral to the user interface, collectively providing a structured pathway from learning about control methods to visualizing their effects on spacecraft behavior. The division not only aids in organization but also aligns with the pedagogical goal of making advanced simulations accessible to all users, regardless of their prior experience with such system.

Design Principles

In the design of our web interface, we embraced the principles of simplicity, consistency, and navigability. The color palette was carefully selected to enhance user engagement and cognitive ease. The primary color, a soothing shade of blue (hex code #2D54DD), transitions gracefully into white, creating a gradient that symbolizes clarity and trust. Black text against this backdrop ensures optimal readability, while the accent color (hex code #F2AE77) adds a distinctive touch to our interface's identity. The generous use of white space around rectangular shapes with rounded edges offers a clean and organized layout, making it easy for users to locate information and navigate the interface without feeling overwhelmed.

The user-friendly approach is particularly crucial in the context of spacecraft control interfaces, where precision and ease of use can significantly impact the success of simulations. Our interface features intuitive toggle bars that allow users to effortlessly adjust control methods and visualize the results through plots. This interactive element not only simplifies the complex process of spacecraft modeling but also encourages users to experiment with different scenarios, fostering a deeper understanding of the simulation outcomes. By streamlining the user's journey through the interface, we ensure that the focus remains on the learning and analytical aspects of spacecraft control rather than the mechanics of the simulation software.

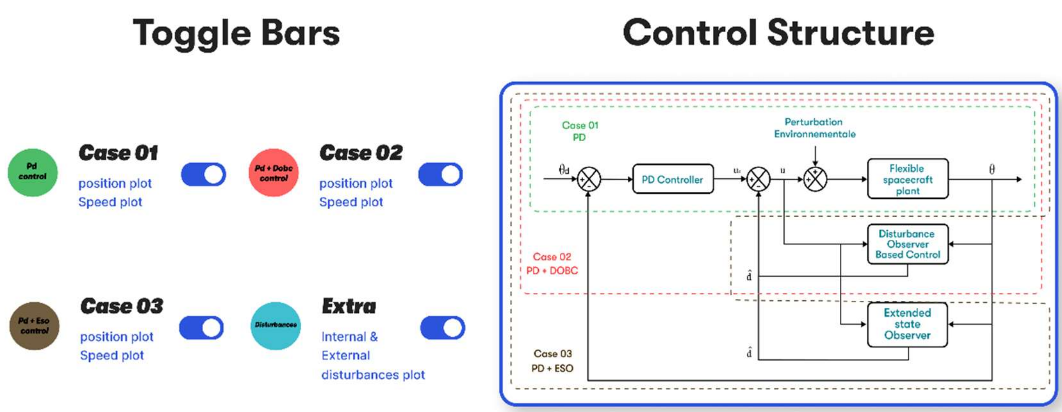

Fig 3.8: The Toggle Bars for the Control of part 02 Simulation Section.

3.3.3. Front-end Technologies

In this project, we employed HTML to structure the web interface, defining elements that present our thesis work and simulations in a clear, browser-friendly manner. CSS was used to style the interface, ensuring it was visually appealing; by combining the structural power of HTML with the styling capabilities of CSS, we were able to create an interface that was not only informative but also intuitive and appealing.

Definitions

HTML (HyperText Markup Language) is the standard markup language for creating web pages. It structures the content on a webpage, defining elements like headings, paragraphs, links, and images. HTML elements are the building blocks of web pages, telling the browser how to display the content.

CSS (Cascading Style Sheets) is a stylesheet language used to describe the presentation of documents written in HTML or XML. CSS controls the layout, colors, fonts, and overall visual appearance of web pages. It allows developers to create engaging and responsive designs, ensuring a consistent look and feel across different devices and screen sizes.

We use HTML and CSS together in web interfaces because they serve complementary purposes. HTML provides the structure and content of a web page, while CSS enhances the presentation and layout. This combination allows for the creation of web pages that are both functional and aesthetically pleasing. Using HTML, we can build the foundation of a webpage, and with CSS, we can style it to be visually appealing and user-friendly. This synergy is essential for developing modern web interfaces that deliver a positive user experience.

The importance of HTML and CSS lies in their roles as the foundational technologies for web development. HTML sets up the structure and content of web pages, while CSS enhances the layout and style, ensuring that websites are both functional and visually appealing1. Together, they form the essential building blocks for creating user-friendly and engaging online experiences.

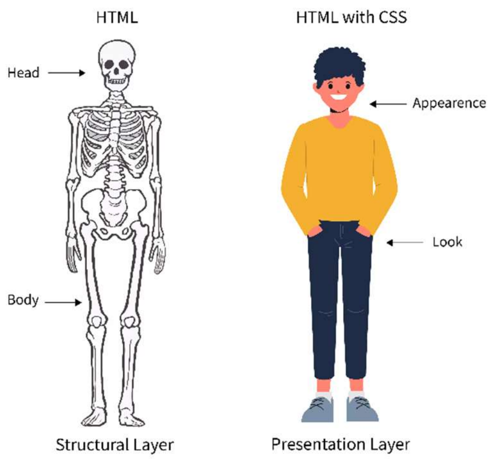

Fig 3.9: HTML and CSS importance.

Fig 3.9, an illustration, shows the importance and the different roles of HTML and CSS in interface development.

3.3.4. Back-end Technologies

The backbone of our web interface is a robust backend system designed to handle complex computations and data management with ease. This system's core is the Django Python framework, a high-level tool that accelerates development while maintaining a clean and pragmatic design. This framework lays the groundwork for a scalable and secure web application, providing essential components that streamline the development process.

- **Django Framework**

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. It provides a robust foundation for building scalable and secure web applications.

In our interface, Django serves as the backbone, simplifying the development process by offering ready-to-use components such as authentication, database management, and routing. It allows us to focus on the unique aspects of our application, enhancing the backbend's efficiency and security.

- **Matplotlib**

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It excels at generating plots, histograms, bar charts, and other types of graphs.

Within our interface, Matplotlib plays a crucial role in visualizing data. It enables us to create clear and interpretable charts and graphs that are embedded in the web interface, providing users with immediate insights into the simulation results.

- **NumPy**

NumPy is an essential package for scientific computing in Python, supporting arrays, matrices, and a broad spectrum of mathematical functions.

NumPy is employed for numerical operations and data manipulation in our backend. It is indispensable for managing large datasets and performing complex calculations, forming the backbone of our application's computational logic.

- **Pandas**

Pandas is a powerful data manipulation and analysis library for Python, offering Data Frames to store and manipulate data efficiently.

Pandas is utilized to handle and process data within our application. It simplifies tasks like data cleaning, transformation, and analysis, proving to be an invaluable asset for backend data operations.

- **Control.py**

Control.py is a Python library tailored for control system design and analysis, providing tools for modeling, simulating, and analyzing dynamic systems.

In our backend, Control.py is essential for implementing control system algorithms. It is vital for applications involving feedback control, system stability analysis, and dynamic system simulation, ensuring accurate and reliable analysis of flexible spacecraft models.

Each of these technologies has been carefully selected to ensure that our backend is not only powerful but also intuitive, allowing for seamless integration with the frontend to deliver a user experience that is both engaging and informative.

3.3.5. Blog and Notion Integration

Notion is a powerful tool for bloggers, offering a simple way to manage content and publish online. Its content management features allow for easy organization and scheduling of articles. The addition of a public API means bloggers can customize their experience and streamline their workflow. With tools like Feather, Notion pages can be turned into SEO-friendly blogs, providing a structured online presence and eliminating the need for coding knowledge. This integration offers bloggers a hosted link, making it easier to share their work with a broader audience.

- **Purpose and Functionality**

The blog which is the Notion parts are integrated into the interface to provide a roadmap and valuable resources for website visitors and partners.

The blog serves as platform to share updates, insights, and the progress of the thesis work. Our Notion page “Pfe” acts as a detailed roadmap that outlines the process of the thesis and offers organized resources for students and partners. This integration ensures that all stakeholders have access to necessary information and can follow the project's development.

- **Design and Implementation**

For our web interface’s blog, we used Notion’s tools to create a clear database with a progress tracker and resource links. We easily hosted and linked the blog to our site using Notion’s sharing features, making it accessible and well-integrated.

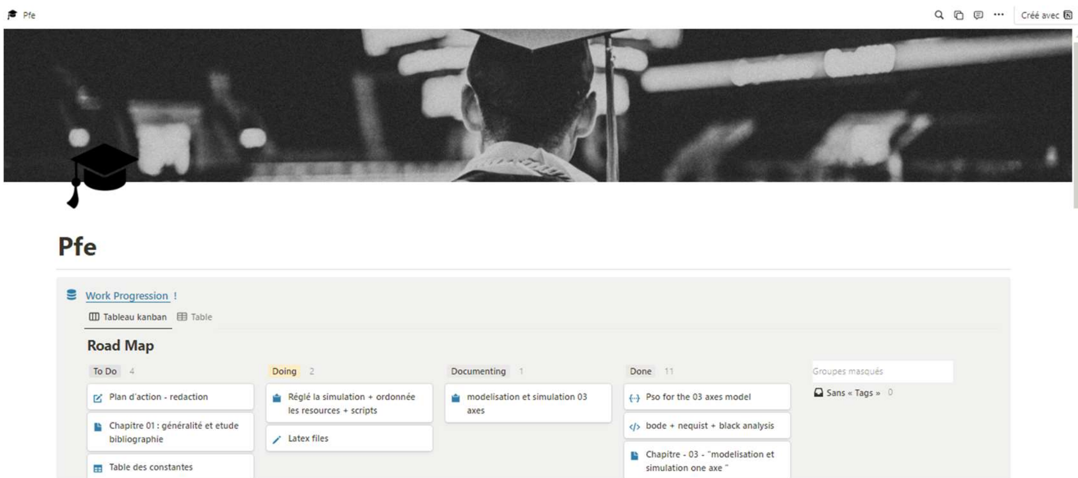

Fig 3.10: The notion Blog page.

Fig 3.10 presents the frontend of our blog as it appears to visitors when accessed through the provided web link.

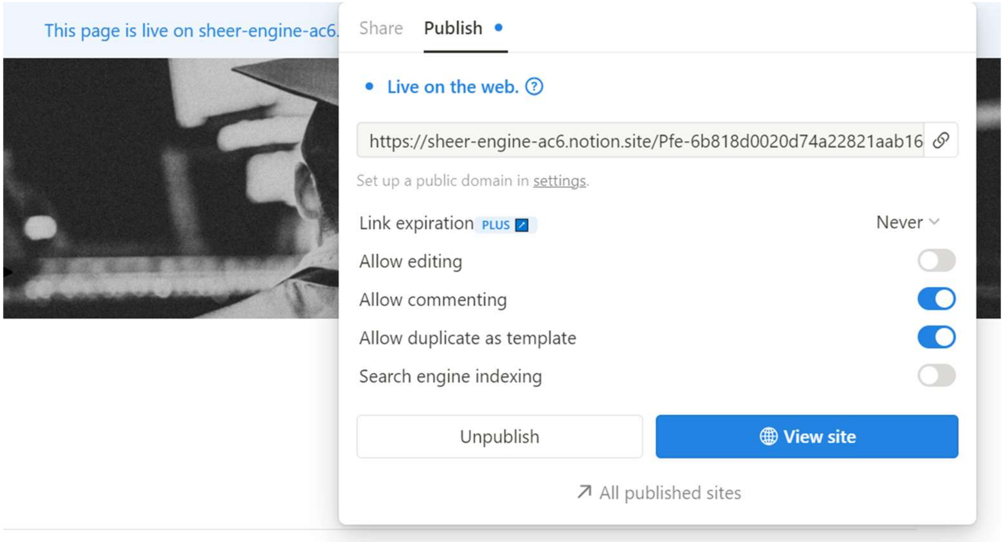

Fig 3.11: The Share to Web Notion functionality.

Fig 3.11 Shows the functionality used to share the blog page as a web link, and for our purpose, we allow comments and template duplicates as a Publish setting.

3.4. Transition from MATLAB to Python

The transition of data from MATLAB Simulink to the Python environment is a critical step in our interface's functionality. Initially, the simulation data is extracted from Simulink using the 'to workspace' function, which saves the data in the .m file format through a dedicated script. Once we have the data in the .m format, we execute our Python script to begin the data processing phase. Here, the pandas library comes into play, renowned for its data manipulation prowess. It allows us to precisely extract the desired data vectors from the .m files. Subsequently, these vectors are converted into a Data Frame structure, a pandas-centric data format that facilitates efficient handling and analysis of tabular data. Finally, the data is shaped into its ultimate form and stored as .csv files, which are then seamlessly integrated into the web interface. This .csv format is particularly advantageous for its compatibility and ease of use within web technologies, ensuring that the data is ready for visualization and further interaction in the user interface.

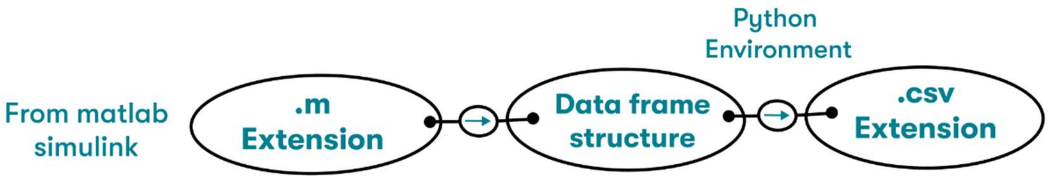

Fig 3.12: Data transition process.

```
clc;
% Define the model name
system1 = 'system_with_PD';
% system2 = 'system_with_PD_DOBC';
% system3 = 'system_with_PD_ESO';
% Load the Simulink models
sim(system1);
%------------------------------------------------------------------------------%
% for case 01 :
save('for_the_interface/one_axis/Pd_control/teta_pd_1x.mat', 'teta_pd_1x');
save('for_the_interface/one_axis/Pd_control/teta_p_pd_1x.mat', 'teta_p_pd_1x');
save('for_the_interface/one_axis/Pd_control/w_pd_1x.mat', 'w_pd_1x');
save('for_the_interface/one_axis/Pd_control/d0_pd_1x.mat', 'd0_pd_1x');
save('for_the_interface/one_axis/Pd_control/t_pd_1x.mat', 't_pd_1x');
%------------------------------------------------------------------------------%
sim(system2);
% for case 02 :
save('for_the_interface/one_axis/Dobc_control/teta_do_1x.mat', 'teta_do_1x');
save('for_the_interface/one_axis/Dobc_control/teta_p_do_1x.mat', 'teta_p_do_1x');
save('for_the_interface/one_axis/Dobc_control/w_do_1x.mat', 'w_do_1x');
save('for_the_interface/one_axis/Dobc_control/d0_do_1x.mat', 'd0_do_1x');
save('for_the_interface/one_axis/Dobc_control/t_do_1x.mat', 't_do_1x');
%------------------------------------------------------------------------------%
```

Fig 3.13: Matlab Script-Data Extraction from the Simulink (part 01).

Fig 3.13 shows a part of the MATLAB script used for data extraction, The script retrieves data from the Matlab workspace and saves it as a .m file, each labeled with a unique name for easy identification and organization.

```
# load the case 01 data #
teta_es_1x=scipy.io.loadmat('for_the_interface/one_axis/Eso_control/teta_es_1x.mat')
teta_p_es_1x=scipy.io.loadmat('for_the_interface/one_axis/ESo_control/teta_p_es_1x.mat')
w_es_1x=scipy.io.loadmat('for_the_interface/one_axis/Eso_control/w_es_1x.mat')
d0_es_1x=scipy.io.loadmat('for_the_interface/one_axis/Eso_control/d0_es_1x.mat')
t_es_1x=scipy.io.loadmat('for_the_interface/one_axis/Eso_control/t_es_1x.mat')
#------------------------------------------------------------------------------------------
# adjust the case 01 data #
# value raw !!
teta_pd_1x=teta_pd_1x["teta_pd_1x"]
teta_p_pd_1x=teta_p_pd_1x["teta_p_pd_1x"]
w_pd_1x=w_pd_1x["w_pd_1x"]
d0_pd_1x=d0_pd_1x["d0_pd_1x"]
t_pd_1x=t_pd_1x["t_pd_1x"]
# to datafram extention #
teta_pd_1x=pd.DataFrame(teta_pd_1x)
teta_p_pd_1x=pd.DataFrame(teta_p_pd_1x)
w_pd_1x=pd.DataFrame(w_pd_1x)
d0_pd_1x=pd.DataFrame(d0_pd_1x)
t_pd_1x=pd.DataFrame(t_pd_1x)
```

Fig 3.14: Python Script-Data transition into data frame (part 01).

Fig 3.14 shows a section of the Python script responsible for transforming data into a DataFrame format. This script efficiently extracts data from the .m files stored in the designated directory, converting them into DataFrame variables. Each variable is systematically named to facilitate straightforward identification and subsequent organization within the interface.

After we process all the simulation data we need, we end up with all the data stored as variables in Python. We can then use Matplotlib to create graphs and other libraries to work with the data for our interface.

3.5. Interface Simulation Process

Navigating through the simulation process using our interface is designed to be intuitive and informative, allowing users to engage with various control methods and analyze their outcomes. Here's a detailed description of how users can interact with the simulation section:

- **One-Axis Simulation**

In this section, users can explore the one-axis flexible spacecraft model controlled by a PD controller. We offer three control designs: PD, PD+DOBC, and PD+ESO. Users can simulate and obtain plots for position and velocity. Additionally, we provide plots for disturbances, both internal and external, offering a comprehensive set of combinations for comparative studies. Users select at least one control method and decide whether to display disturbances before running the simulation.

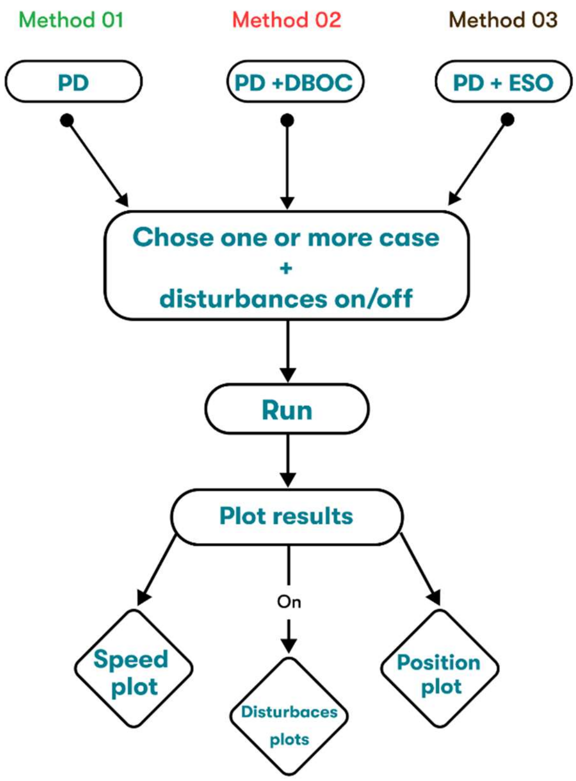

Fig 3.15: Flowchart of the simulation process "one axis simulation".

Fig 3.15 shows the flowchart describing the logic of the outcomes and the incomes in this section of the simulation. "Part one"

Fig 3.16: Settings area as toggle bars for the "one axis simulation".

- **Three-Axis Simulation**

Here, four different control methods are available: PD, Sliding Mode with three variations, and two Adaptive Sliding Mode methods. Users have the ability to run simulations for two methods as a comparative study. They must choose two out of the four methods and decide on displaying disturbances. If sliding mode is one of the chosen methods, they must select one of its modes. The simulation results include position and speed plots, along with the RMSE (Root Mean Square Error) for the two selected methods.

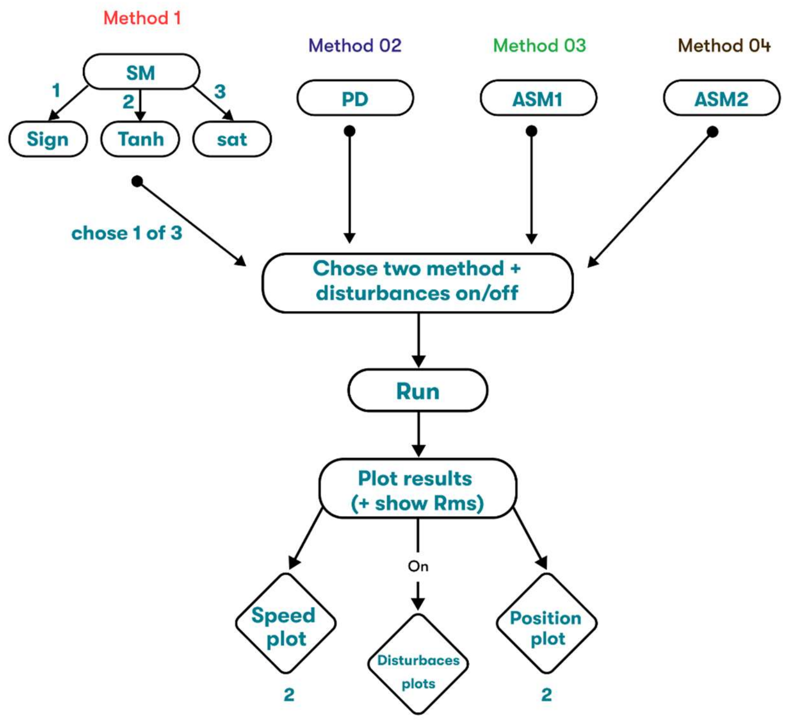

Fig 3.17: Flowchart of the simulation process "Three axis simulation".

Fig 3.17 shows the flowchart describing the logic of the outcomes and the incomes in this section of the simulation. “Part Two”.

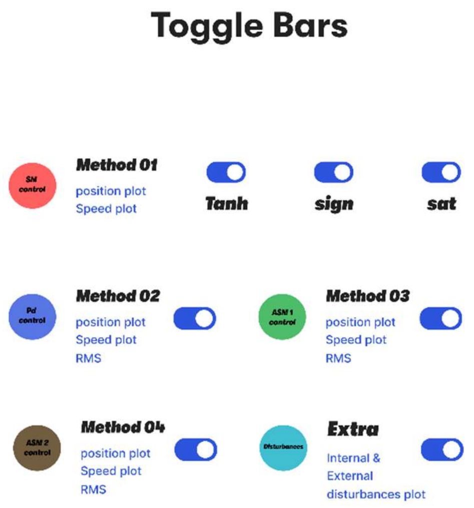

Fig 3.18: Settings area as toggle bars for the "Three axis simulation".

- **Flexible Stability Analysis**

This section offers two simulation models, Free Mode, and Closed Loop Mode, for three different flexible appendage systems. Users must select at least one simulation mode and one of the three systems provided. Upon setting these parameters, they run the simulation to obtain the results in the form of Bode and Nicholas charts.

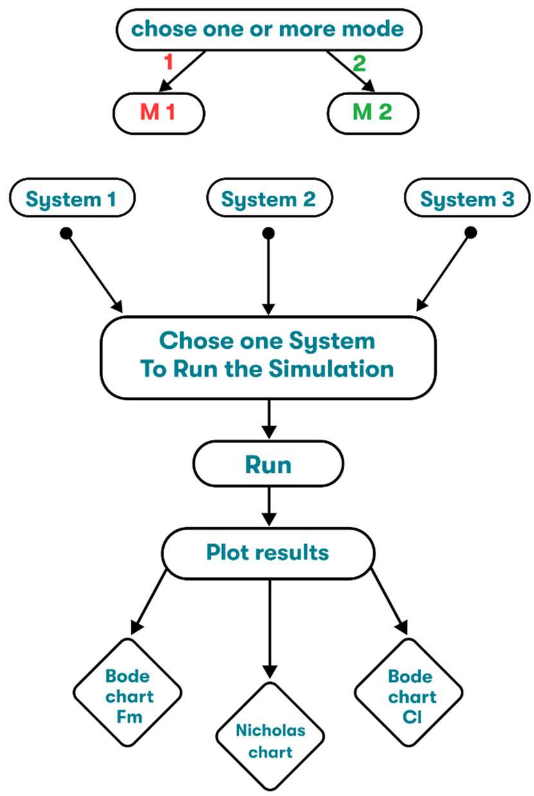

Fig 3.19: Flowchart of the simulation process "Flexible analysis".

Fig 3.19 shows the flowchart describing the logic of the outcomes and the incomes in this simulation section. “Part Three”.

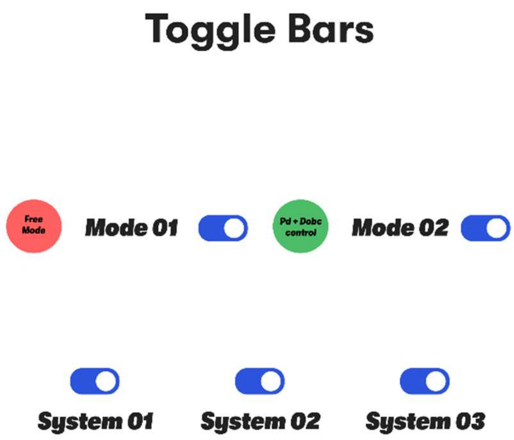

Fig 3.20: Settings area as toggle bars for the "Flexible analysis".

Each part of the simulation section is logically structured to guide users through the different methods our work provides, ensuring a user-friendly experience while delivering complex simulation analysis.

- **Results**

After setting up the simulation, the interface will show the results as clear charts and graphs. These visuals are created using Matplotlib, a tool that turns our simulation data into easy-to-understand images. The results section of the interface neatly arranges these images, making it simple for users to see and compare the outcomes of different simulations.

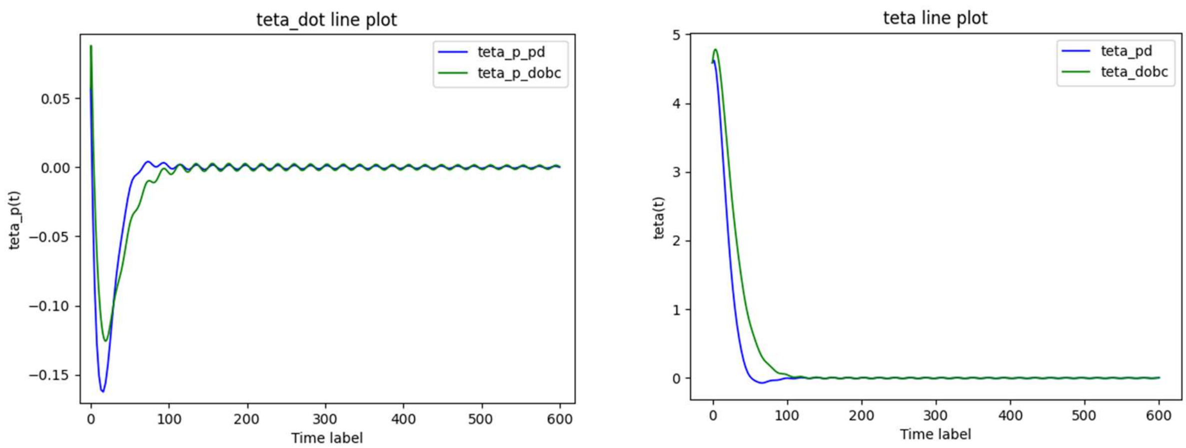

Fig 3.21: The position and velocity plots for the “one axis simulation”.

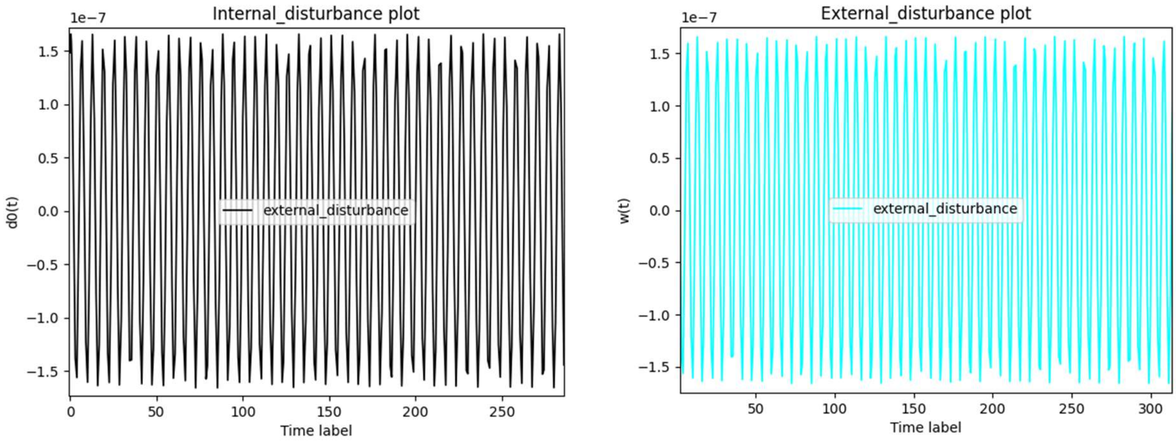

Fig 3.22: Internal and external disturbances plot for the “one axis simulation”.

Fig 3.21 and Fig 3.22 illustrate the outcomes of the one-axis simulation with PD and PD+DOBC methods selected, PD+ESO deselected, and disturbances enabled. These figures are strategically placed in the ‘Results’ section of Part One of the interface. The generation and placement of these plots are handled as backend processes, ensuring that the frontend remains streamlined and user-focused.

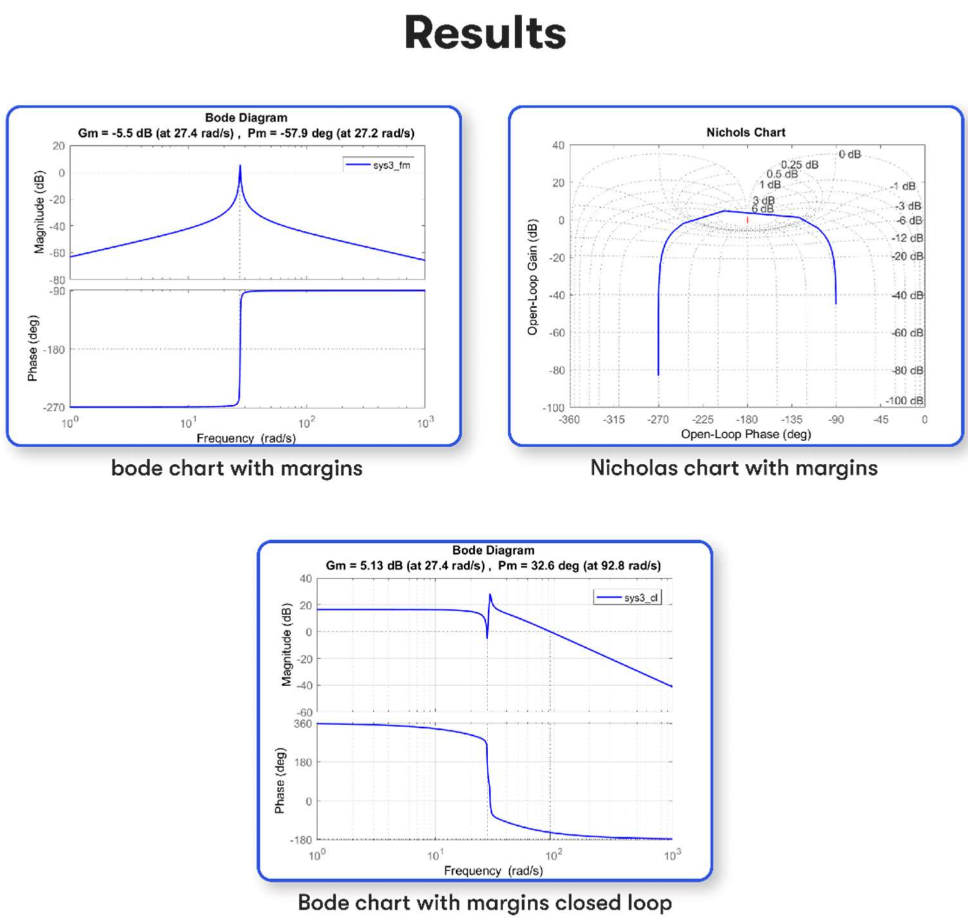

Fig 3.23: Results outcomes for the “flexible analysis”.

Fig 3.23 Shows the outcomes of the Flexible analysis with the two modes selected and the second system selected. These figures are strategically placed in the 'Results' section of Part Three of the interface.

3.6. Conclusion

Our interface is a bridge between advanced simulation technology and users. It is simple to use and helps people learn about spacecraft dynamics through hands-on experience. We have ensured that it is not just for experts but everyone curious about space technology.

In the journey of our interface's conception and design, we have successfully achieved our objective of creating an interface that simplifies the intricate processes involved in spacecraft simulation. This accomplishment reflects our technical abilities and our dedication to making scientific research more approachable.