POST Explorer: A Design Space Exploration Tool for POST2

R. Anthony Williams[1]
NASA Langley Research Center, Hampton, Virginia, 23681, USA

James A. Hoffman[2]
Analytical Mechanics Associates, Inc., Hampton, Virginia, 23666, USA

Rafael A. Lugo[3]
NASA Langley Research Center, Hampton, Virginia, 23681, USA

Recent improvements for the Program to Optimize Simulated Trajectories II (POST2) have included the development of an application programming interface (API). This API allows POST2 simulation inputs to be directly manipulated from other applications (such as MATLAB or Python), and the outputs from POST2 are streamed directly to the external application that enables visualization, data manipulation, etc. Through this framework, a new tool called POST Explorer is being developed that provides a user the capability to modify the simulation inputs and interrogate the outputs within the same application, with raw data inspection and visualization embedded. This tool can be leveraged for multiple types of analyses, such as parametric sweeps and sensitivity studies, and will be available with a future release of the POST2 software.

I. Introduction

The Program to Optimize Simulated Trajectories II (POST2) is a tool used for trajectory performance analysis for missions from pre-phase A through active flight operations. POST2 is developed at NASA's Langley Research Center, and it is utilized across the NASA agency, industry, and academia [1]. A recent improvement to the tool is the development of an application programming interface (API) [2]. The POST2 API allows for external tools to manipulate settings within POST2 directly, as well as receive data output without the need to write information to disk. Efficient and accurate methods of interfacing with POST2 provide new or improved ways to solve trajectory propagation and optimization problems [3].

While the API itself can be powerful when leveraged by an external framework that a user is familiar with, e.g., MATLAB or Python, the strength of the API is apparent when tools are built utilizing it for a particular purpose. These tools can enable flight mechanics analysts new ways to interface with POST2. POST2 is inherently an expert system that requires knowledge of not only the physics being simulated to propagate the vehicle's trajectory, but also the simulation settings that enable the correct calculations. This can be a daunting task for many users, and thus it was desired to have a framework that allowed usage of the expert system without full knowledge of the underlying tool. While the goal is to develop an interface for non-expert POST2 users to work with a simulation, there is still a need for a set of inputs to be created, referred to as an input deck, by a POST2 expert user. This allows the expert POST2 user to ensure the simulation runs as expected, while providing a framework where non-expert users can directly modify the inputs and interrogate the effects on the behavior of the trajectory.

In this work, the design of the POST Explorer tool will be discussed in Section II. The design will cover the framework that was leveraged for development, as well as key aspects of the tool. Potential use cases of the tool in its

[1] Research Computer Scientist, Atmospheric Flight and Entry Systems Branch.
[2] Software Development Senior Manager, Atmospheric Flight and Entry Systems Branch.
[3] Aerospace Engineer, Atmospheric Flight and Entry Systems Branch, AIAA Senior Member.

current state will be discussed in Section III. This list of use cases is not exhaustive, but it is meant as a starting point of ways POST Explorer can be utilized. The final section will summarize the potential impacts that a tool such as POST Explorer could have on a project.

II. Design of POST Explorer

The design of POST Explorer was driven by a simple top-level requirement: provide a method for a non-expert POST2 user to modify, execute, and interrogate a POST2 simulation directly. Execution of these simulations still require an expert POST2 user to set up the simulation settings in an input deck, but POST Explorer should allow for the non-expert user to complete multiple types of analyses without the expert user in the loop.

Development of the POST Explorer tool was based on a few main principles. Firstly, considering this was a new tool development that would potentially require programming languages and frameworks not used for POST2 before, attempt to maximize maintainability of the software for future development and future upgrades. Second, leverage as much existing software as possible, thus minimizing the overhead needed to lay the foundation for the tool. Lastly, the goal was to minimize code duplication as much as possible. Minimal code duplication aids in maintaining the software, but it also minimizes a potential error source that can occur if code changes are not implemented properly.

A. Framework

Building on the work demonstrated in [2], the programming language chosen for POST Explorer development was Python. This decision was largely motivated by the maintainability of the software, as well as leveraging existing libraries. Additionally, using Python allows for a more platform-agnostic software, and all information that is used for both the graphical user interface (GUI) and the trajectory simulation reside within the Python memory space – making for easier, more efficient communication between the simulation and the GUI. The design of POST Explorer is intended to be lightweight, similar to the design of the POST2 API. While lightweight design may result in simpler looking interfaces for the user, the software will also be simpler to maintain and upgrade as the needs for the tool evolve.

To minimize code needed to be written from scratch, the Qt framework was selected to develop the POST Explorer GUI. This decision was made since Qt is widely used across multiple platforms, and there exists multiple Python modules with bindings to Qt. Additionally, it is well documented and examples and tutorials are readily available [4]. More specifically, the PySide6 module was utilized for the Python bindings to the Qt framework, as PySide6 is the official module from the Qt for Python project [5].

The overall functionality of the GUI is written using PySide6; however, this library does not support data visualization as well as some others do. Thus, for visualizations of data as the trajectory simulations complete, the PyQtGraph library was selected [6]. There are many built options that come standard with the PyQtGraph library that made it attractive, such as log-log plots and Fourier transforms. Additionally, it leverages numpy for all computations where possible, so it is efficient in terms of memory and compute time [6].

The development of POST Explorer would be impossible without the work completed to implement the POST2 API. POST Explorer calls the POST2 API directly to modify settings, query the value of data, and leverage the POST2 dictionary. Inputs that can be modified and outputs that can be tracked are pulled from the dictionary, which populates a large amount of information that is used by the GUI (variable name, units, description, etc.). The POST2 dictionary is a list of human-readable variable names paired with pertinent information about that variable, such as where in a data structure it can be found, units, dimensions, etc. Additionally, this information can be populated from POST2 input deck settings called aliases, where an expert POST2 user may choose to rename a POST2 dictionary entry to something more verbose (e.g. rename gamma to flight_path_angle_degrees, or to use project-specific nomenclature). An option exists to leverage either the entire POST2 dictionary and all aliases, or to just populate information and settings using aliases only. This was done to simplify things further for the user by limiting the number of parameters they can modify and eliminating some POST2-specific knowledge that is associated with the dictionary names.

POST Explorer is divided into three tabs that cover the tasks that can be completed using the tool. A user will analyze a simulation by modifying input settings and parameters, adjusting data from POST2 they are concerned with, and executing one or more simulations. The user may then *explore* the data that comes from POST2 visually, comparing multiple runs against each other or interrogating a single run further. The final tab displays *data* from the completed simulations in a raw table format.

B. Analyze Tab

The first tab that appears in POST Explorer is the *Analyze* tab. This tab will be used to set up any analysis to be completed by defining how the simulation will execute differently from the settings laid out in the input deck set up

by the expert user. The expert POST2 user can prepopulate suggested inputs and settings to assist the non-expert user, which allows the non-expert user to complete meaningful analyses that are informed by the expert user.

POST2 is an event driven simulation that has the capability of simulating multiple vehicles simultaneously [1]; the settings that are modified within the Analyze tab have the option to specify for which event to modify a parameter, e.g. decrease time step size starting at the main parachute deployment event. Trajectories will be executed by defining a few key pieces of information described below.

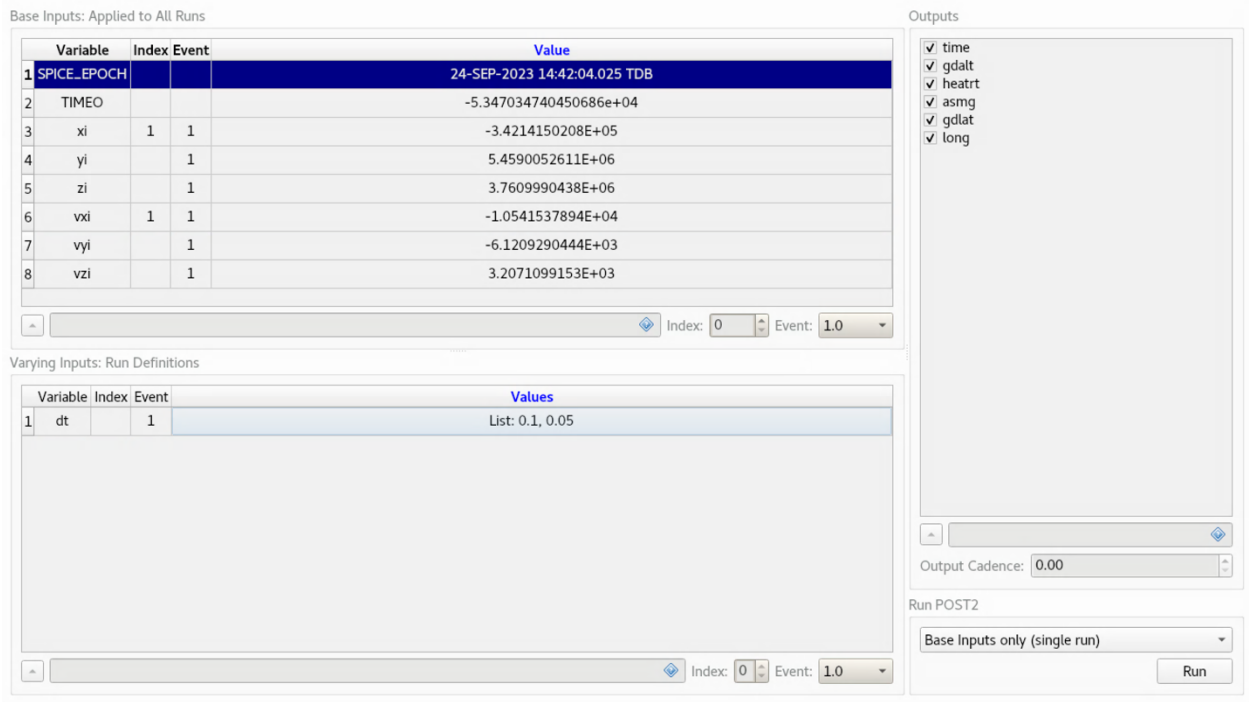

Fig. 1 An example of a populated Analyze tab, which is modifying vehicle initial states with a sweep across multiple defined time step sizes. Variable names are taken from the POST2 dictionary.

1. *Base Inputs*

Base inputs define settings that will be applied to each trajectory that is simulated. An example might be that for an Earth reentry trajectory, the simulation will finish at main parachute deploy as opposed to propagating to touchdown. Modifications made to settings in the *Base Inputs* table should be used for parameters that only have a single value to incorporate into the simulation, as opposed to a list of multiple values in a parametric sweep.

2. *Varying Inputs*

Varying inputs define how the individual trajectories differ from each other. The varied input values can be a list of discrete options, or a parametric sweep can be defined using a start, stop, and step size value. Each value of the varied input(s) defines a separate trajectory to be simulated. The varied inputs are applied to the simulation on top of the modifications made in the *Base Inputs*. For instance, in Fig. 1, the parameter that is being varied is time step size (dt), where the user can investigate how the simulation time step size affects the behavior of the trajectory. Two discrete time step sizes define two separate trajectories to be propagated, where both trajectories will start at the initial vehicle state defined in the *Base Inputs*.

3. *Outputs*

The outputs define what data the user is concerned with interrogating the value of when the simulation has completed. Time is the only default value that is populated for the user, which is utilized for plotting purposes to generate time histories. Data will not be collected from POST2 without requesting them in the *Outputs* table. This is one way that the POST2 API is lightweight since it will update all data within the POST2 simulation, but it will only collect time histories of the requested parameters. An additional option the user can control is what rate the output data will be collected, where the default rate is the simulation time step. The memory needed to store the time histories from all runs can be decreased if the data is requested at a lower rate.

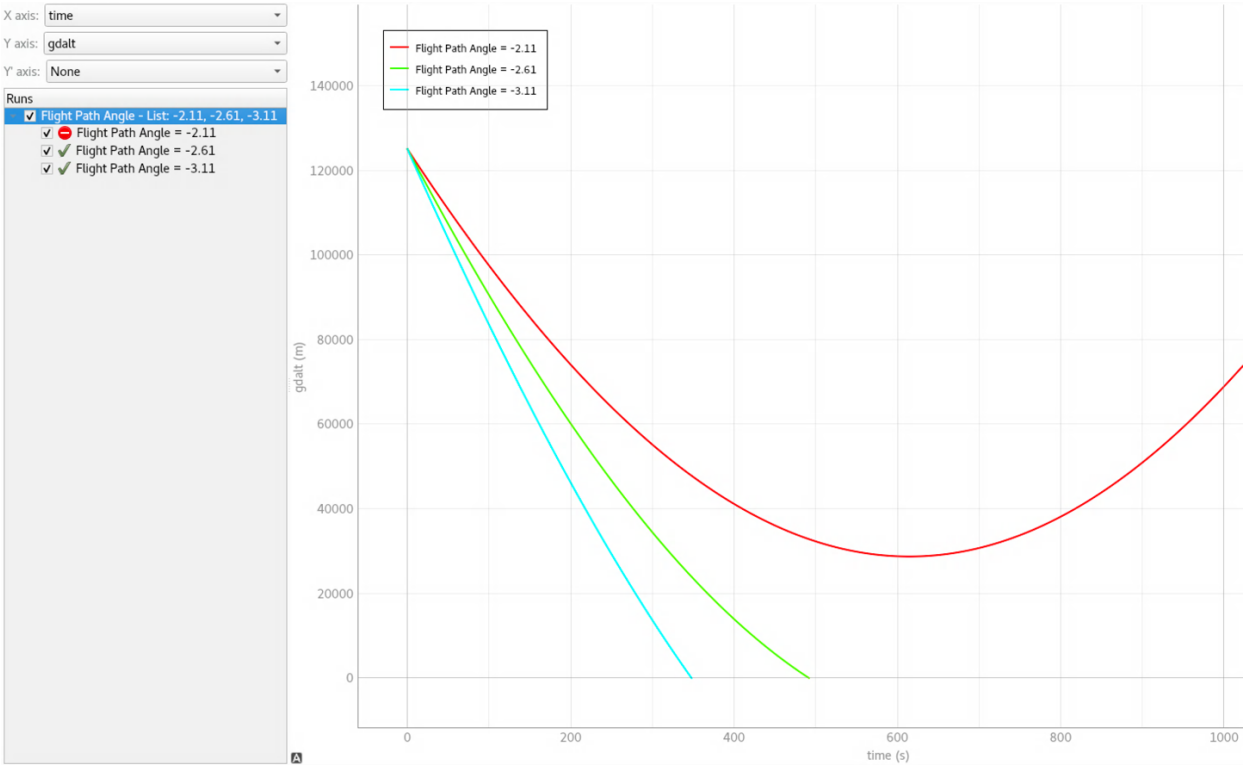

Fig. 2 A populated Explore tab, showing a parametric sweep of flight path angle, and how varying it affects geodetic altitude (gdalt) over time.

C. Explore Tab

The *Explore* tab is meant to be a way for the user to explore the trajectories by interacting graphically with the data. The tab is populated as each trajectory completes its propagation. Each trajectory will populate in the left-hand side table view under a top-level item that describes the run type that was chosen in the *Analyze* tab, and each subsequent entry beneath the top-level item is one of the unique trajectories that was simulated. As part of this list, an icon next to each trajectory to help the user quickly identify which trajectories may be cause for concern: a green checkmark means no warnings or errors, a yellow caution sign signals the existence of only warnings, while the red do-not-enter sign means there were fatal errors that occurred during that trajectory's propagation. These warnings and errors can be viewed by right-clicking on the trajectory and selecting the "View Warnings and Errors" item.

The plot window within the *Explore* tab utilizes the PyQtGraph library, which comes with a rich set of plotting features such as modifying one or more of the axes to be logarithmically scaled. One potentially useful feature is a method to export data from the plot window to multiple formats. Plots in the *Explore* tab are influenced by what output variables and output rate were selected from the *Analyze* tab. The default variable to use for the x-axis in the plot window is time, but this can be modified to be any of the output variables chosen. Units for each variable are queried directly from the POST2 dictionary and populated in the axis labels, as demonstrated in Fig. 2. Additionally, a watermark with the username of the person who generated the data is placed automatically as a traceability measure.

D. Data Tab

The *Data* tab provides the user a means to view data in a raw format. Although the *Explore* tab shows data visually, it can be useful to view data in a tabular numerical format.

E. Explorer File

The *Explorer File* (.exp) was developed to assist in using POST Explorer. This file contains all the information necessary to setup the initial POST2 trajectory, as well as all generated data from any previous analyses that were completed. The *Explorer File* allows the expert user to set up POST Explorer, prepopulate some settings and inputs, and then deliver the file as one package that contains all inputs, the POST2 library, and anything else needed to complete the simulations.

An additional use case that the *Explorer File* enables is for the expert user to run one or more sets of analyses, and then deliver the file to the non-expert user to view the data interactively without having to run the analyses themselves. This may be particularly useful in situations where the expert user is knowledgeable on what the non-expert user is attempting to run.

Since the *Explorer File* contains all input data as well as any generated output data, the file size can become large if a lot of data was created from multiple sets of analyses. To decrease memory usage, the output rate can be modified to output data less often. However, POST Explorer is similar to the POST2 API in that the user is only limited in how much data they can save by the amount of memory on their machine.

III. Potential Use Cases

There are many potential use cases for POST Explorer, or other tools like it that are built to leverage the API for a specific purpose, so the list discussed in this section will not be exhaustive. While POST Explorer was developed so that a non-expert POST2 user would be able to complete meaningful analysis without as much POST2 knowledge, expert POST2 users may find this tool helpful as well.

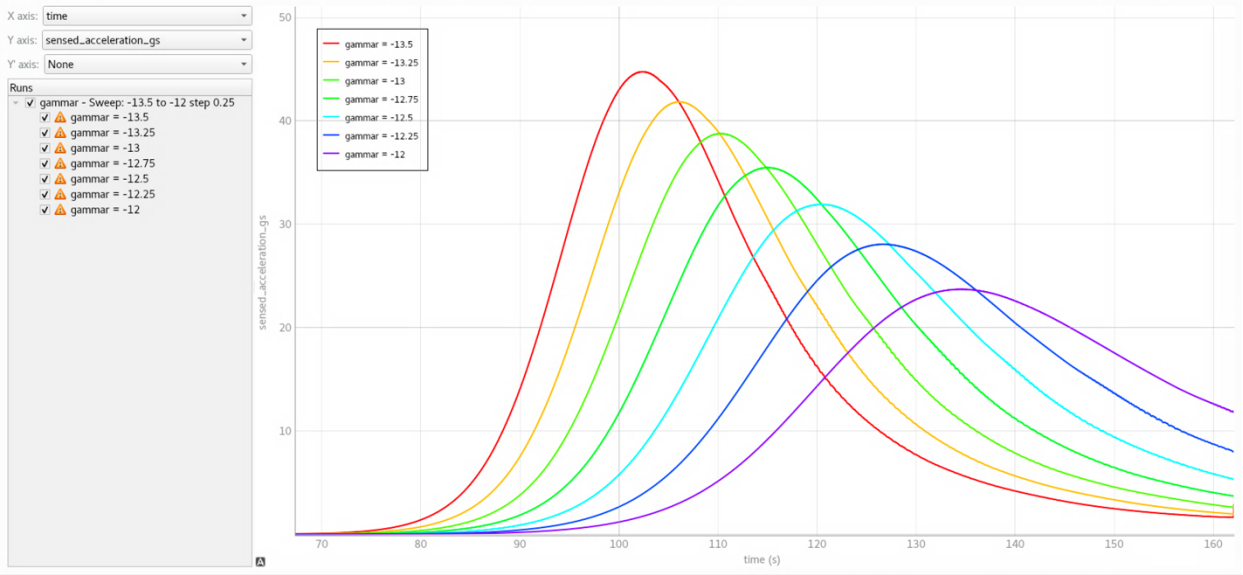

Fig. 3 Use case depicting a sweep on entry flight path angle (gammar) and its effect on sensed acceleration.

A. Single Parameter Sweep

While in the development phase of missions, it is imperative to determine requirements for the mission based on the design limits of the underlying system. An example is demonstrated in Fig. 3 where a parametric sweep was completed on entry flight path angle, and the effect it has on sensed acceleration is shown. If the vehicle design allows for a peak deceleration load of 35 g's, then a flight path angle no steeper than -12.5 should be desired when designing entry maneuvers.

This use case can also be helpful in determining sensitivities of the system by conducting one variable at a time (OVAT) analyses. All inputs to the simulation would remain constant while a single parameter is varied.

Fig. 4 Use case depicting three sets of entry states propagated for different trajectory correction maneuvers (TCMs).

B. Single Set of Parameters Interrogation

One potential use case would be a single set of parameters interrogation where only one type of data is updated, as opposed to only a single parameter being modified. In Fig. 4 there are three different entry states (position and velocity, six total components) corresponding to the three trajectory maneuvers that are being assessed. Within the simulation, the only thing that updates between the TCMs is the entry state. The trajectory is plotted in latitude-longitude space showing how the three trajectories differ. This is a potentially common use case for when a mission gets to flight operations.

C. Model Implementation Validation

POST2 contains algorithms of varying levels of fidelity for many physical models. During earlier stages of development, lower fidelity models are often used. As the missions approaches later stages of development, higher fidelity models are needed to evaluate if the system meets mission requirements. Therefore, it is crucial that POST2 incorporate the higher fidelity models as they are delivered to the flight mechanics analyst. A major step in this integration of outside models is validation that the model behaves as the model developer expects. Typically, the model developer is familiar with POST2, but they are not an expert user. Hence, a use case for POST Explorer is for the expert POST2 user to set up a simulation with the model incorporated, and then deliver a POST Explorer file to the model developer so that they can interrogate the behavior of the model themselves. They would be able to tweak the model inputs from the Analyze tab, and then examine the outputs graphically in the Explore tab. Allowing the model developer to interact with POST2 in this manner may highlight some nuances in model integration that may not be seen otherwise.

D. Training and Onboarding

Training new users on the intricate details of POST2 to become an expert user is a nontrivial task. POST Explorer can be used to alleviate some of the need for POST2 knowledge while allowing a new user to directly see the impacts of modifying inputs to the simulation. The baseline simulation can be set up by an expert POST2 user, and then the new user can experiment with pushing the simulation to its limits.

Additionally, onboarding an expert user to an established project can be difficult. Setting up a current baseline for the given project that the onboarding user can use as a starting point to learn the top-level quirks of the system being simulated can assist in that.

E. Deliverable to Project Stakeholders

Project stakeholders (management, subsystem leads, etc.) may be very interested in how the system operates during entry, descent, and landing (EDL), and they may be curious as to how the behavior of the vehicle changes for a variety of updates (entry states, guidance algorithm gains, etc.). However, the stakeholders are likely not expert POST2 users,

but POST Explorer may assist in that. By delivering a POST Explorer file with the current simulation setup to a stakeholder, the stakeholder is now enabled to determine interesting information about the vehicle system without as much interaction with the POST2 expert user. This could be especially useful for a systems engineer who is interested in where the system starts to break down.

IV. Summary

A design space exploration tool for POST2 called POST Explorer was developed. This tool allows for non-expert POST2 users to interface directly with a POST2 simulation by modifying inputs and interrogating outputs. While the tool does not remove the need to have an expert POST2 user involved, it does decrease the amount of involvement on a day-to-day basis the expert user would have.

Without the recent updates to POST2 to be thread safe and the development of the POST2 API, POST Explorer would not be possible. POST Explorer is the first tool developed with the POST2 API directly, and it is meant to serve two purposes: to be a design space exploration tool, and to demonstrate how powerful the POST2 API can be when leveraged to solve specific problems. A non-exhaustive list of potential use cases for POST Explorer was discussed, largely highlighting the utility for non-expert users; there is potential for expert POST2 users to benefit from POST Explorer as well. The POST Explorer software will be available with a future POST2 release.

Acknowledgments

This project was supported by Entry Systems Modeling, which is a program under NASA's STMD's Game Changing Development Program.

References

[1] R. A. Williams, R. A. Lugo, S. M. Marsh, J. A. Hoffman, J. D. Shidner and J. T. Aguirre, "Enabling Thread Safety and Parallelism in the Program to Optimize Simulated Trajectories II," in *AIAA SciTech Forum*, National Harbor, MD, 2023.

[2] R. A. Williams, R. A. Lugo and J. A. Hoffman, "Design of an Application Programming Interface for the Program to Optimize Simulated Trajectories II," in *AIAA SciTech Forum*, Orlando, FL, 2024.

[3] R. A. Williams, Z. May and C. Brown, "End-to-End Trajectory Optimization Using Copernicus and Program to Optimize Simulated Trajectories II," in *IEEE Aerospace Conference*, Big Sky, MT, 2024.

[4] "Qt for Python Documentation," The Qt Company Ltd., 2024. [Online]. Available: https://doc.qt.io/qtforpython-6/#quick-start. [Accessed 12 November 2024].

[5] "Qt for Python," The Qt Company Ltd., 2024. [Online]. Available: https://doc.qt.io/qtforpython-6/. [Accessed 12 November 2024].

[6] "PyQtGraph Scientific Graphics and GUI Library for Python," 2021. [Online]. Available: https://www.pyqtgraph.org/. [Accessed 12 November 2024].