Functional Documentation for Python Quantum Simulation Library

June 18, 2025

Introduction

This document provides the functional documentation for a Python-based quantum simulation library. The library is designed to simulate a qubit under control dynamics and measurements. It supports both mathematical modeling and graphical visualization, and optionally provides a GUI for user interaction.

Installation

To install and run the simulation library, you need the following dependencies:

- Python 3.8 or above
- numpy
- scipy
- matplotlib
- PyQt5 (for GUI)

To install, clone the repository and run:

```
pip install -r requirements.txt
python main.py
```

Main Components

The project is structured around the following core classes:

Qubit

Represents the physical properties of a qubit:

- `omega` – frequency
- `kappa` – control strength
- `gamma1, gamma2` – decoherence parameters

Control

Represents external control field applied to the qubit.

QuantumSystem

Encapsulates the Bloch vector state of a qubit (x, y, z) and handles its evolution via differential equations.

Observer

Handles the measurement process and defines the observable operator.

UtilsModule

Includes utility functions, constants (Pauli matrices), and advanced simulation functions like measurement simulations and elimination algorithms.

Usage Example

Below is an example of how to create and evolve a qubit:

```
from QubitModule import Qubit
from ControlModule import Control
from SystemModule import QuantumSystem
from ObserverModule import Observer
from utils import E1
import numpy as np

q = Qubit(1.0, 0.5, 0.2, 0.1)
c = Control(1.0)
s = System(0.0, 0.0, 1.0)
o = Observer(E1)

Evolve system

s.evolve(q, c, 1.0)

Measure

result = o.measure(s, q, c, 1.0, 100)
```

Graphical Interface (GUI)

The project optionally includes a GUI built with PyQt5. It enables the user to:

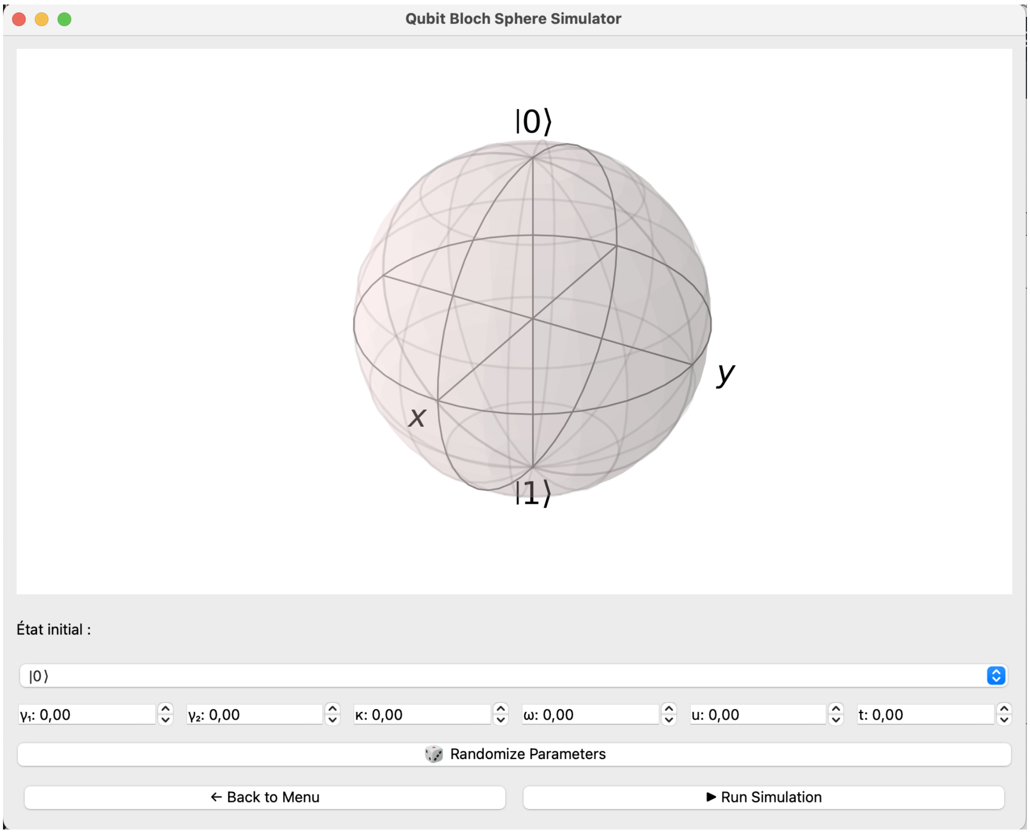
- Configure qubit and control parameters
- Visualize state evolution
- Run simulations and see results interactively

Launching the GUI

Run the GUI using:

```
python graph.py
```

Screenshot of the GUI



Extending the Project

You can add your own:

- Measurement operators
- Qubit configurations
- Elimination algorithms for parameter estimation

Conclusion

This library provides a comprehensive framework for simulating qubit dynamics under noisy environments and controls. It is suitable for research, education, and experimentation.