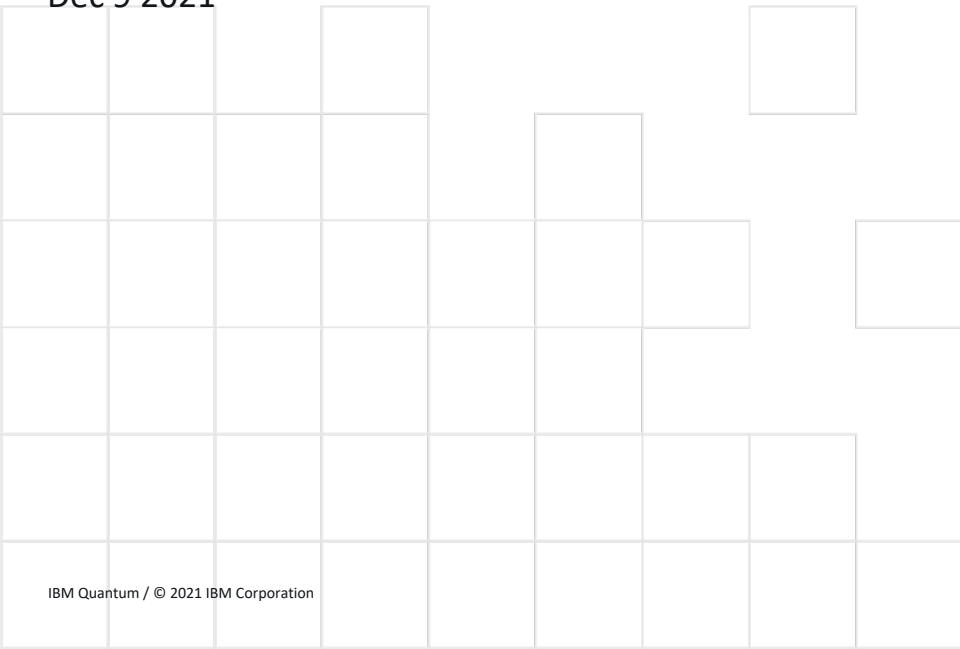


#44 Update qiskit-experiment tutorials

Mentor: Helena Zhang

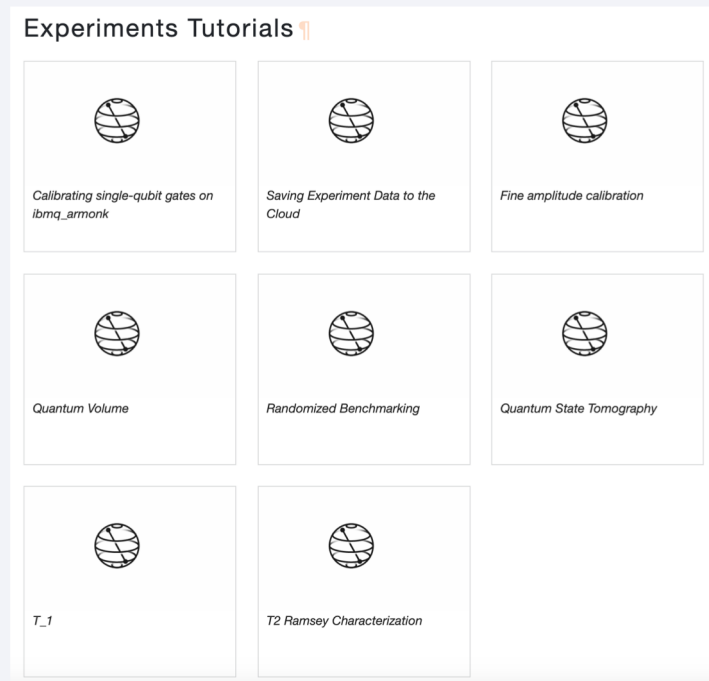
Jale Ipekoglu

Dec 9 2021



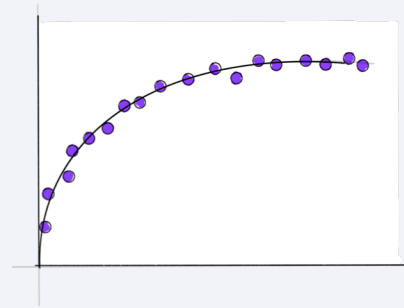
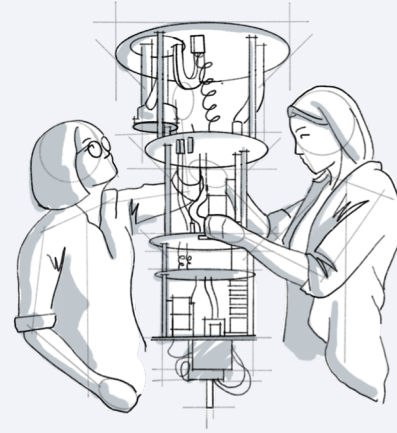
Refactoring Existing and Creating New Tutorials

- Regroup and reformat
- Step-by-step progression
- Goal-oriented how-to guides
- Create a new guide plan
- Define the outline
- Decide on prioritization
- Writing the guides



My Journey - Calibration Experiments

- Not coming from physics background
- No knowledge about qiskit-experiments
- But eager to learn about the underlying hardware



My Resources Along the Way



- Qiskit Textbook
- Calibrations section - <https://qiskit.org/textbook/ch-quantum-hardware/calibrating-qubits-pulse.html>
- qiskit-experiments API documentation
- My mentor and my teammate answered the rest of my questions

The screenshot displays two pages from the Qiskit website. The top page is the 'Calibrating Qubits with Qiskit Pulse' tutorial, which is part of the Qiskit Textbook. It features a navigation bar with 'Overview', 'Learn', 'Community', and 'Documentation' links. The main content area includes a table of contents on the left, a search bar on the right, and the main text of the tutorial. The bottom page is the 'Qiskit Experiments API Reference' page, which lists various package and experiment modules.

Calibrating Qubits with Qiskit Pulse

The new Qiskit Textbook beta is now available. [Try it out now.](#)

Qiskit is an open-source framework for programming quantum computers (Ref. 1). Using Qiskit, quantum circuits can be built, simulated and executed on quantum devices.

Qiskit Pulse provides a language for specifying pulse level control (i.e. control of the continuous time dynamics of input signals) of a general quantum device independent of the specific hardware implementation (Ref. 2).

In this tutorial, we show how to implement typical single-qubit calibration and characterization experiments using Qiskit and Qiskit Pulse. These are typically the first round of experiments that would be done in the lab immediately after a device has been fabricated and installed into a system. The presentation is pedagogical, and allows students to explore two-level-system dynamics experimentally. All units are returned as standard SI (i.e., Hz, sec, etc.).

Each experiment gives us more information about the system, which is typically used in subsequent experiments.

Qiskit Experiments API Reference

Package Modules

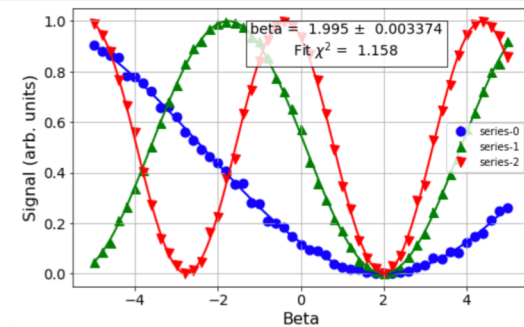
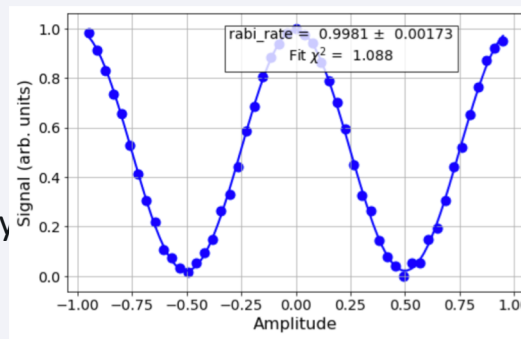
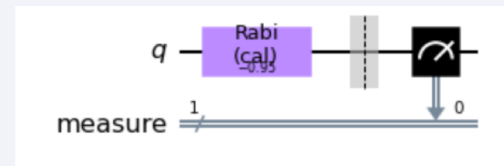
- Qiskit Experiments (`qiskit_experiments`)
- Experiment Framework (`qiskit_experiments.framework`)
- Experiment Library (`qiskit_experiments.library`)
- Data Processing (`qiskit_experiments.data_processing`)
- Curve Analysis (`qiskit_experiments.curve_analysis`)
- Calibration Management (`qiskit_experiments.calibration_management`)
- Database Service (`qiskit_experiments.database_service`)
- Qiskit Experiments Test Utilities (`qiskit_experiments.test`)

Experiment Modules

- Calibration Experiments (`qiskit_experiments.library.calibration`)
- Characterization Experiments (`qiskit_experiments.library.characterization`)
- Randomized Benchmarking Experiments (`qiskit_experiments.library.randomized_benchmarking`)
- Tomography Experiments (`qiskit_experiments.library.tomography`)
- Quantum Volume Experiment (`qiskit_experiments.library.quantum_volume`)

Challenges

- Biggest challenge was to understand the concept and terms
- Having only one Qiskit textbook page and the API documentation as resources
 - *Suggestions;*
 - More documentation about calibrations
 - Adding some terms to IBM Quantum Glossary
- Refactoring of the module



Being a Part of QAMP and What Next?

- Reading,
- Trying out experiment module,
- Learning,
- Working with my mentor weekly
- and having this great opportunity to ask my questions to experts
- *Next*; continue my contribution to the module

Docs > Experiments Tutorials > Run a Single-Qubit Calibration Experiment

Run a Single-Qubit Calibration Experiment ¶

To produce high fidelity quantum operations, we want to be able to run good gates. The calibration module in qiskit-experiments allows users to run experiments to find the pulse shapes and parameter values that maximizes the fidelity of the resulting quantum operations. Calibrations experiments encapsulates the internal processes and allow experimenters do calibration operations in a quicker way. Without the experiments module, we would need to define pulse schedules and plot the resulting measurement data manually (see also [Qiskit textbook](#) for calibrating qubits with Qiskit Terra).

Each experiment usually provides additional information about the system used in subsequent experiments.

```
import numpy as np

import qiskit.pulse as pulse
from qiskit.circuit import Parameter

from qiskit_experiments.calibration_management import BackendCalibrations
```

On our own environment, we may use one of the pulse-enabled real backends for all the experiments like below.

```
# from qiskit import IBMQ
# IBMQ.load_account()
# provider = IBMQ.get_provider(hub='ibm-q', group='open', project='main')
# backend = provider.get_backend('ibmq_armonk')
```

We can verify whether the backend supports Pulse features by checking the backend configuration.

```
# backend_config = backend.configuration()
# assert backend_config.open_pulse, "Backend doesn't support Pulse"
```

Qiskit / qiskit-experiments Public

< Code Issues 80 Pull requests 42 Actions Projects Wiki Security Insights

[WIP] Refactor tutorial for calibration experiments #499

Open jaleipekoglu wants to merge 10 commits into [Qiskit:main](#) from [jaleipekoglu:main](#)

Conversation 1 Commits 10 Checks 11 Files changed 3

jaleipekoglu commented on Nov 5

Summary

I am converting and updating the existing Jupyter notebook for the calibrating armonk tutorial to rst. This is part of a larger refactor as a part of the Qiskit advocate mentorship program.

Details and comments

I will be updating the current document and converting it to rst file.

Thanks to my Mentor, my Teammates, and to
IBM Quantum Community!