A Pulse Simulator backend for Qiskit Experiments

JeongWon Kim, R. K. Rupesh, Daniel J. Egger

To properly test the experiments in Qiskit Experiments we need to simulate transmon dynamics with the schedules specified by the experiments. These tests need to be efficient and fast if they are to be included in the test suite of Qiskit Experiments. Since the last checkpoint, we created the “SingleTransmonTestBackend” a subclass of “IQPulseBackend”. This backend is used to test single-qubit experiments. We initialize it with the qubit frequency, anharmonicity, transition strength between neighboring levels and an optional relaxation rate. From these arguments, the backend constructs the drift and control Hamiltonians of a three-level transmon model and applies the rotating frame approximation. “IQPulseBackend” has default pulse schedules corresponding to the $X, \sqrt{X}$ gates. The time-evolved operators corresponding to theses default pulses are obtained with the Qiskit-Dynamics Solver by changing each pulse schedules into Signals and computing the time-dynamics under the model Hamiltonian. The unitary of the $R_Z(\theta)$ gate is computed directly. When the run method of the backend is executed on circuits which have their own calibrations the backend extracts the pulse schedules from the calibrations and computes the corresponding time-evolution operators with the Qiskit-Dynamics solver. Importantly, for each circuit, the backend caches the calibrations that it has already solved to efficiently execute repeated quantum gates within the circuit. Repeated gate sequences are typical in error amplifying sequences. This is done by maintaining a dictionary of simulated pulse-schedules in which the keys are a tuple of instruction name, qubits, and parameters and the value is a unitary matrix. This efficiency makes the backend usable in a testing suite.

Furthermore, by giving the relaxation rate to the backend, we can include the effect of T_1 noise which is, e.g., needed to properly simulate spectroscopy experiments. When there is no noise, we treat the Hamiltonian model with the statevector. However, when noise is present in the system, we compute the evolution of the density matrix. With the statevector or density matrix, we can construct the measurement data as counts of IQ data depending on the measurement level.

Finally, we are refactoring the existing single-qubit experiment tests to use our pulse-based backend. For example, we have refactored the tests for the Rabi experiments. With the pulse backend these tests run in 27 seconds while with the fake mock backend these tests run in 17 seconds.

Overall, this backend will make the tests in Qiskit Experiments more realistic and allow us to write tutorials that do not need to be run on the hardware. Furthermore, this backend may help guide the design of a general-purpose pulse simulator in Qiskit Dynamics.