IOP Publishing Quantum Sci. Technol. 10 (2025) 015026 https://doi.org/10.1088/2058-9565/ad895b

Quantum Science and Technology

OPEN ACCESS

RECEIVED
7 November 2023

REVISED
23 July 2024

ACCEPTED FOR PUBLICATION
21 October 2024

PUBLISHED
13 November 2024

PAPER

Procedure for reducing cross-resonance gate errors using pulse-level control

David Danin[1] **and Felix Tennie**[1,2,*]

[1] Department of Physics, Clarendon Laboratory, University of Oxford, Parks Road, Oxford OX1 3PU, United Kingdom
[2] Department of Aeronautics, Imperial College London, South Kensington Campus, London SW7 2AZ, United Kingdom
[*] Author to whom any correspondence should be addressed.

E-mail: felix.tennie@physics.ox.ac.uk

Keywords: quantum computing, customised 2-qubit gates, superconducting qubits

Abstract

Current implementations of superconducting qubits are often limited by the low fidelities of multi-qubit gates. We present a reproducible and runtime-efficient pulse-level approach for calibrating an improved cross-resonance gate CR(θ) for arbitrary θ. This CR(θ) gate can be used to produce a wide range of other two-qubit gates via the application of standard single-qubit gates. By performing an interleaved randomised benchmarking sequence, we demonstrate that our approach leads to significantly lower incoherent errors than the circuit-level approach currently used by IBM. Hence, our procedure provides a genuine improvement for applications where noise remains a limiting factor.

Quantum computers promise to provide unprecedented computational power in applications such as optimisation or simulation by exploiting the fact that information is not encoded in classical but in quantum systems [1, 2]. In recent years, the field has seen the rapid development of improved quantum hardware [3]. However, the practical benefit of commercially available quantum computers based on superconducting qubits remains limited by the relatively low fidelities of multi-qubit interactions [4].

In addition to the improvement of hardware components, it is possible to enhance gate fidelities using optimised control approaches [5, 6]. With the introduction of Qiskit Pulse [7], it is now possible to precisely control real quantum hardware via the IBM Quantum Lab [8]. As explained in [9], one can specify the amplitude, frequency and phase of the physical microwave pulses that drive the qubits to implement custom single-qubit and multi-qubit gates [10]. Hence, Qiskit Pulse allows for designing and testing control approaches on the level of physical operations instead of logical operations [11].

The cross-resonance gate (CR) is a particularly important two-qubit interaction on superconducting qubits, as it combines various desirable features and enables the construction of the Controlled-NOT gate [12] which is the standard entangling operation of universal gate sets [2]. As demonstrated in [13], one can implement a high-fidelity CR gate using the pulse sequence schematically illustrated in figure 1. When driving the control qubit Q_C at the resonant frequency ω_T of the target qubit Q_T, the system generally evolves under a Hamiltonian of form

$$H_I = \frac{\sigma_z \otimes \sum_{i=X,Y,Z} C_{Zi}\sigma_i}{2} + \frac{I \otimes \sum_{i=X,Y,Z} C_{Ii}\sigma_i}{2}. \quad (1)$$

When specifically calibrated, this can be simplified to implementing the interaction $H_I \approx g(Z \otimes X)$ with g some coupling constant that depends on the hardware components and the drive amplitude [13, 14]. The time evolution operator generated by this Hamiltonian reads $U(t) = \cos(gt)\mathbf{1} \otimes \mathbf{1} - i\sin(gt)Z \otimes X$ [10]. Setting $gt = \pi/4$ by changing the amplitude or duration of the pulses, a CR($\pi/2$) gate is implemented [13]. In our work, we utilise the above model for the Hamiltonian to predict and investigate a generalised evolution for arbitrary rotation angles by varying the pulse duration. We demonstrate this new physical regime of two-qubit interactions on real IBM Quantum backends. It should be stressed that our approach is not specifically limited to the IBM Quantum machines but rather offers a new tool for improving the control of two-qubit interactions on superconducting quantum computers in general.

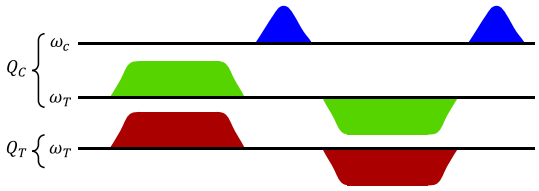


Figure 1. The schematic adapted from [13] illustrates the pulse schedule for a CR($\pi/2$) cross-resonance gate. The control qubit Q_C is driven at the resonant frequency ω_T of the target qubit Q_T. Unwanted terms in the interaction Hamiltonian H_I are suppressed by the echo sequence on Q_C (i.e. the upper two drive lines) and cancellation tones on Q_T (i.e. the lower drive line), such that $H_I \approx g(Z \otimes X)$ follows. Adapted figure with permission from [13], Copyright (2016) by the American Physical Society.

By using Qiskit Pulse on publicly available quantum backends via the IBM Quantum Lab, the method presented in [13] can be extended to significantly reduce the incoherent error of multi-qubit gates[3]. Specifically, we describe a pulse-level approach for calibrating a set of cross-resonance gates and demonstrate that they significantly reduce the incoherent error compared with the circuit-level implementations[4] used by IBM. Crucially, the procedure we present is straightforwardly replicated. Accordingly, we provide a powerful extension to the set of high-fidelity, multi-qubit gates on currently available quantum computers based on superconducting qubits.

We extend previous research on customised cross-resonance gates in at least two ways. First, we improve previous calibration procedures. On the one hand, we describe the calibration of the cross-resonance pulse amplitude. This extends [15, 16] which only consider a rescaling of the pulse duration. Our extension is particularly relevant to applications such as Hamiltonian Simulation where small rotations are required. On the other hand, we describe the calibration of cancellation tones. This extends [17] and is important for reducing unitary errors. Second, our transparent presentation here provides a useful extension by making the calibration procedure readily available to other researchers.

The rest of this paper will be structured as follows. First, we introduce a runtime-efficient procedure for calibrating a $Z \otimes X$ cross-resonance interaction CR(θ) via the IBM Quantum Lab, thereby extending the approach presented in [13] to values of θ other than $\pi/2$. Second, we describe how this CR(θ) gate can be used to straightforwardly implement a range of other two-qubit interactions. And finally, we demonstrate that our pulse-level implementation significantly reduces the incoherent error, compared to the circuit-level implementation which IBM currently uses, by performing a modified interleaved randomised benchmarking sequence [18].

We begin by presenting our procedure for calibrating a CR(θ) gate. The method is adapted from [13] but differs in two important respects. First, we generalise the procedure to values of θ other than $\pi/2$. And second, we streamline the procedure to make it more runtime-efficient. This enables us to perform the full calibration procedure on publicly available quantum backends via the IBM Quantum Lab, even under runtime constraints.

First, we need to determine the correct amplitude for the CR(θ) pulse between our control qubit Q_C and target qubit Q_T. For this, we define a flat-top pulse with Gaussian edges and some real amplitude A. The width and Gaussian rise time of the pulse are inherited from the CR($\pi/2$) pulse that forms part of the standard Controlled-NOT implementation between Q_C and Q_T on IBM Quantum which is available through Qiskit Pulse. While testing these parameters might lead to a more precise calibration, we adopt this assumption to significantly reduce the calibration runtime. We note that this assumption is self-consistent since it leads to a high-fidelity CR(θ) gate as shown in the subsequent demonstrations.

Then, we sweep through different real amplitude values A and measure Q_T in the computational basis to calculate the Pauli expectation value $\langle Z(A) \rangle$. We repeat the sequence with Q_C initialised in $|0\rangle$ and $|1\rangle$. Assuming that the $Z \otimes X$ or $Z \otimes Y$ component in H_I is much larger than the other contributions, we find $\langle Z \rangle \approx \cos(\theta)$, as for an ideal CR($\theta$) gate we have $\langle Z \rangle = \cos(\theta)$. Note that the assumption made here is consistent with the results of the subsequent tomography sequences. Hence, for a given θ, we can find the amplitude A_θ that leads to the correct value of $\langle Z \rangle$ and use this amplitude for our pulse.

Second, we need to determine the correct phase for the CR(θ) pulse. For this, we sweep through different pulse widths with our flat-top Gaussian pulse using the real amplitude that we previously determined. We repeat the demonstration with Q_C initialised in $|0\rangle$ and $|1\rangle$. By measuring the expectation values $\langle X \rangle$, $\langle Y \rangle$,

[3] The authors consider the (imperfect) pulses used to drive qubits as a form of an interaction with the environment and therefore adopt the term 'incoherent' for the resulting errors.

[4] A circuit level implementation of a unitary evolution is given by a decomposition of the unitary operator into elements of a universal gate set.

and $\langle Z \rangle$ on the target qubit, we reconstruct the coefficients of the terms in the cross-resonance interaction Hamiltonian H_I. For details regarding the Hamiltonian tomography sequence, we refer the reader to [13, 19].

Hence, we can determine the coefficients C_{ZX} and C_{ZY} of the cross-resonance $Z \otimes X$ and $Z \otimes Y$ components in H_I, respectively. Recognising that $C_{ZX} \propto \cos(\phi - \phi_0)$ and $C_{ZY} \propto \sin(\phi - \phi_0)$, where ϕ is the phase of the cross-resonance pulse [12], we can set the phase of the pulse to $\phi_0 = -\tan^{-1}(C_{ZY}/C_{ZX})$ such that the $Z \otimes Y$ component in H_I vanishes. Thereby, we can calibrate the phase of the cross-resonance pulse in a single sequence. This provides a far more efficient method than sweeping through phases as described in [9, 13].

Third, we need to determine the correct phase and amplitude for the cancellation pulse, which is a resonant flat-top Gaussian pulse on the target qubit with the same duration and Gaussian rise times as the cross-resonance pulse. The purpose of the cancellation pulse is to neutralise the $\mathbf{1} \otimes X$ and $\mathbf{1} \otimes Y$ components in H_I. The correct phase for the cancellation tone can be inferred from the Hamiltonian tomography sequence we already performed. By reading off the $C_{\mathbf{1}X}$ and $C_{\mathbf{1}Y}$ coefficients of the $\mathbf{1} \otimes X$ and $\mathbf{1} \otimes Y$ components in H_I, we can calculate $\phi_1 = -\tan^{-1}(C_{\mathbf{1}Y}/C_{\mathbf{1}X})$. As the phase of the cross-resonance pulse is set to ϕ_0, the correct phase for the cancellation tone is $\phi_0 - \phi_1$ as presented in [13].

To determine the correct amplitude, we perform two Hamiltonian tomography sequences for the full pulse schedule in figure 1. In the first iteration, the cancellation tone amplitude is set to zero while in the second iteration, we set it to some value A_0. The correct order of magnitude for A_0 can be estimated from the cancellation tone of the CR($\pi/2$) pulse that forms part of the Controlled-NOT implementation between Q_C and Q_T. Hence, we can extract the values $C^1_{\mathbf{1}X}$ and $C^1_{\mathbf{1}Y}$ as well as $C^2_{\mathbf{1}X}$ and $C^2_{\mathbf{1}Y}$ from the two iterations. Assuming a linear relationship between the cancellation tone amplitude and the coefficients as seen in [13], we find $A_X = A_0 C^1_{\mathbf{1}X}/(C^1_{\mathbf{1}X} - C^2_{\mathbf{1}X})$ and $A_Y = A_0 C^1_{\mathbf{1}Y}/(C^1_{\mathbf{1}Y} - C^2_{\mathbf{1}Y})$.

If the value of ϕ_1 is calibrated correctly, then we find $A_X \approx A_Y$ as the unique solution for the correct amplitude of the cancellation tone [13]. Hence, to calibrate the full cross-resonance pulse sequence, we only require four Hamiltonian tomography sequences which provides a far more efficient procedure than the calibration methods described in [9, 13]. Furthermore, it is now possible to calibrate the pulse sequence such that it implements a CR(θ) gate for values of θ other than $\pi/2$.

We have implemented this procedure using the seven-qubit IBM Quantum backend `ibm_oslo` with qubit 2 and qubit 1 as the control and target qubit, respectively. The resonance frequency and anharmonicity of the control qubit are $f_2 = 4.962$ GHz and $\delta_2 = -0.344$ GHz, and $f_1 = 5.046$ GHz and $\delta_1 = -0.343$ GHz for the target qubit [20]. For reproducibility, the full configuration and properties of the backend can be found in appendix A. To illustrate our generalised procedure by way of example, we calibrate a CR(θ) gate for $\theta = \pi/5$. In all demonstrations, we use at least 4 000 repetitions per circuit such that statistical errors are negligibly small. We also mitigate readout errors using the method described in [21]. It should be stressed that our results do not rely on this classical post-processing and that the significantly reduced incoherent error of our customised two-qubit gates can be observed in the unmitigated data. A detailed discussion of the technique is given in appendix B.

The results of the amplitude calibration are illustrated in figure 2 and the results of a demonstration that verifies the calibration are displayed in figure 3. For these two demonstrations, we have used 20 000 repetitions per circuit. Setting the pulse width to the inherited width as described above, we receive the CR($\pi/5$) gate. This shows that we can use our runtime-efficient, pulse-level procedure to calibrate a CR(θ) gate for θ other than $\pi/2$.

Having implemented the CR(θ) gate with $H \propto Z \otimes X$, it is straightforward to implement a range of other two-qubit interactions. As illustrated in figure 4, we can use standard single-qubit gates on Q_T and Q_C to convert the $Z \otimes X$ interaction into any $A \otimes B$ interaction with $A, B \in \{X, Y, Z\}$. Note that we have written the gate that corresponds to the Hamiltonian $H = (-\theta/2) A \otimes B$ as $AB(\theta)$ for ease of notation. The relations in figure 4 are easily proven using standard gate identities [2]. Finally, the $XZ(\theta)$, $YZ(\theta)$, and $YX(\theta)$ gates can either be implemented by circuit identities used in figure 4, or alternatively by swapping the control and target qubit in the calibration procedure. Hence, we conclude that having calibrated the $ZX(\theta)$ gate, it is straightforward to implement any of the nine $AB(\theta)$ gates with $A, B \in \{X, Y, Z\}$.

Using pulse-level methods, we can further extend the set of easily implemented two-qubit gates. Note that the S and $S^\dagger$ gates in figure 4 correspond to virtual phase shifts with $\Delta\phi = \pm\pi/2$ on the relevant qubit, respectively [22]. As Qiskit Pulse allows us to directly specify a phase shift, we can also implement values of $\Delta\phi$ other than $\pi/2$. For instance, by shifting the phase of the cross-resonance pulse and cancellation tone by $\Delta\phi_0$, we can convert the $ZX(\theta)$ gate into a $Z(\cos(\Delta\phi_0)X + \sin(\Delta\phi_0)Y)(\theta)$ gate. While this treatment is not exhaustive, it nicely illustrates that using circuit-level and pulse-level methods, a range of two-qubit interactions are straightforwardly implemented once we have calibrated the $ZX(\theta)$ gate.

Any of these gates can also be implemented using circuit-level methods with at most three Controlled-NOT gates [23]. However, the advantage of our pulse-level implementation of the $ZX(\theta)$ gate is

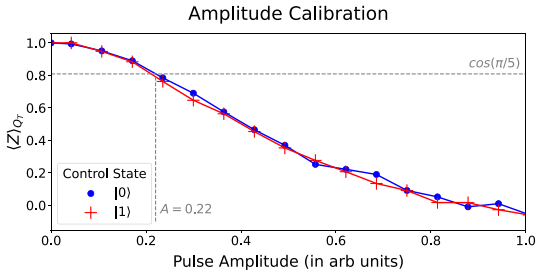


Figure 2. Results for the amplitude calibration sequence with $\theta = \pi/5$. We determine the correct pulse amplitude by reading off A_θ which is defined by $\langle Z(A_\theta)\rangle = \cos(\theta)$. Using this amplitude will lead to a CR($\pi/5$) gate which has the same duration as the CR($\pi/2$) pulse that forms part of the standard Controlled-NOT implementation between Q_C and Q_T. The error bars are smaller than the marker size.

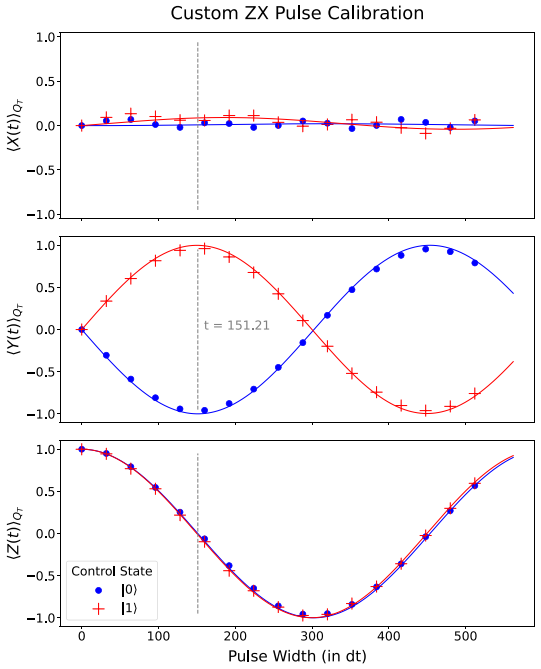


Figure 3. Results of the Hamiltonian tomography sequence using the fully calibrated cross-resonance pulse sequence. We fit the data as described in [19] to extract the coefficients of the contributions in the interaction Hamiltonian H_I and find the values indicated at the bottom of the figure. The coefficient C_{ZX} of the $Z \otimes X$ term is significantly larger than all other coefficients which indicates a successful calibration. The error bars are smaller than the marker size. The unit $dt = 0.222$ ns is pre-set by the IBM Quantum platform.

that we require fewer two-qubit pulses as illustrated in figure 5. As two-qubit interactions on superconducting qubits are susceptible to noise [10], reducing the number of two-qubit pulses should reduce the incoherent error of the $ZX(\theta)$ gate. Further, as single-qubit gates achieve near-perfect fidelities [24] while virtual phase gates have perfect fidelities [22], converting our $ZX(\theta)$ gate into other two-qubit gates as in figure 4 should not lead to a significant additional error.

We test both hypotheses by performing an interleaved randomised benchmarking sequence similar to those in [18, 25]. To measure the incoherent error we proceed as follows. We define a set of gate sequence lengths $\{m_1, \ldots, m_j\}$ with $\Delta = m_{i+1} - m_i$ some fixed positive integer and $m_j = N$. Using the `StandardRB` method in Qiskit Pulse, we sample a set of random gate sequences $\{C_1, \ldots, C_N\}$ and define a new set of gate sequences $\mathbf{R} = \{R_{m_1}, \ldots, R_{m_j}\}$ with $R_{m_i} = C_1 \ldots C_{m_i} \tilde{C}_{m_i}$ where $\tilde{C}_{m_i}$ inverts the previous operations such that the action of each R_{m_i} is just the identity operation. Then, to measure the incoherent error of the standard and custom $ZX(\theta)$ gate, we interleave $ZX(\theta)ZX(-\theta)$ after every C_l in each R_{m_i}, giving two sets of gate sequences $\mathbf{R_S}$ and $\mathbf{R_C}$, respectively. We run each of the gate sequences in $\mathbf{R}$, $\mathbf{R_S}$ and $\mathbf{R_C}$, and measure the fractional ground state population. As the sequences in $\mathbf{R_S}$ and $\mathbf{R_C}$ are identity operations that acquire an additional error due to the interleaved cross-resonance gates, we expect the ground state population to decay

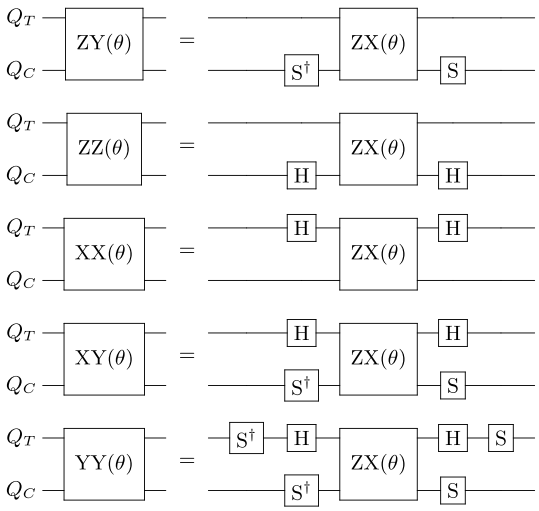


Figure 4. Circuit identities that show how the $ZX(\theta)$ gate can be converted into a range of other two-qubit cross-resonance gates with $H \propto A \otimes B$ where $A, B \in \{X, Y, Z\}$, using single-qubit gates only. The relations are easily shown by applying standard gate identities from [2] to the time evolution operator of the $ZX(\theta)$ gate $U(t) = \cos(gt)\mathbf{1} \otimes \mathbf{1} - i \sin(gt) Z \otimes X$. Here, H indicates the Hadamard gate and S the phase gate.

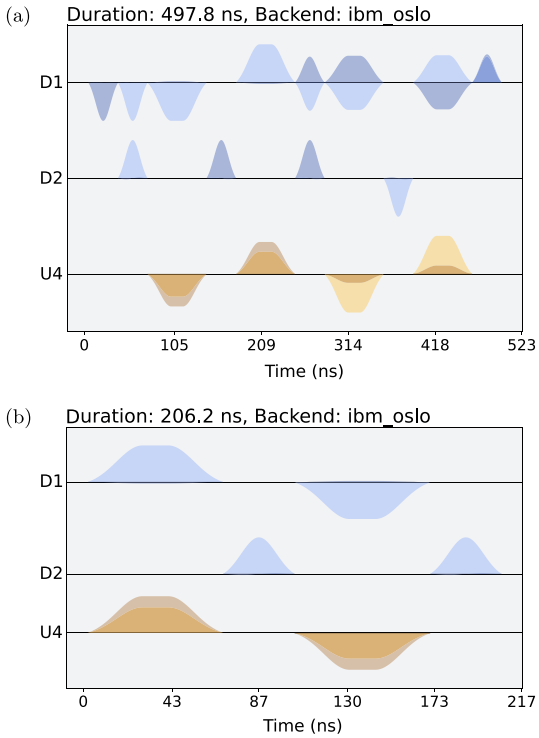


Figure 5. (a) Schedule for the $ZX(\theta)$ gate using IBM's circuit-level implementation. The four yellow pulses are the cross-resonance pulses which form part of the two required Controlled-NOT gates. The gate duration is 497.8 ns. (b) Schedule for the $ZX(\theta)$ gate using our pulse-level approach which only requires two cross-resonance pulses, halving the number of two-qubit pulses in comparison with the circuit-level approach. The gate duration is reduced to 206.2 ns. By comparison, typical qubit lifetimes are of the order of 100 *mus*, cf figure 8, and are therefore not a limiting factor.

faster by a factor of F^{2m_i}, where $F \leqslant 1$ characterises the additional error introduced by the $ZX(\theta)$ gate, in comparison to the case of non-interleaved gate sequences in **R**. By fitting the data to an exponential decay, we find the values of F_S and F_C.

We have implemented the interleaved benchmarking procedure using the same quantum backend and qubits as described above. The gates *ZX, ZY, ZZ, XY, XX* and *YY* are tested using $m_1 = 5$, $\Delta = 7$, and $N = 68$ as benchmarking parameters. For the custom implementation we have used the $ZX(\pi/5)$ gate implemented above while for the standard implementation we have used circuit-level methods in Qiskit [26], employing

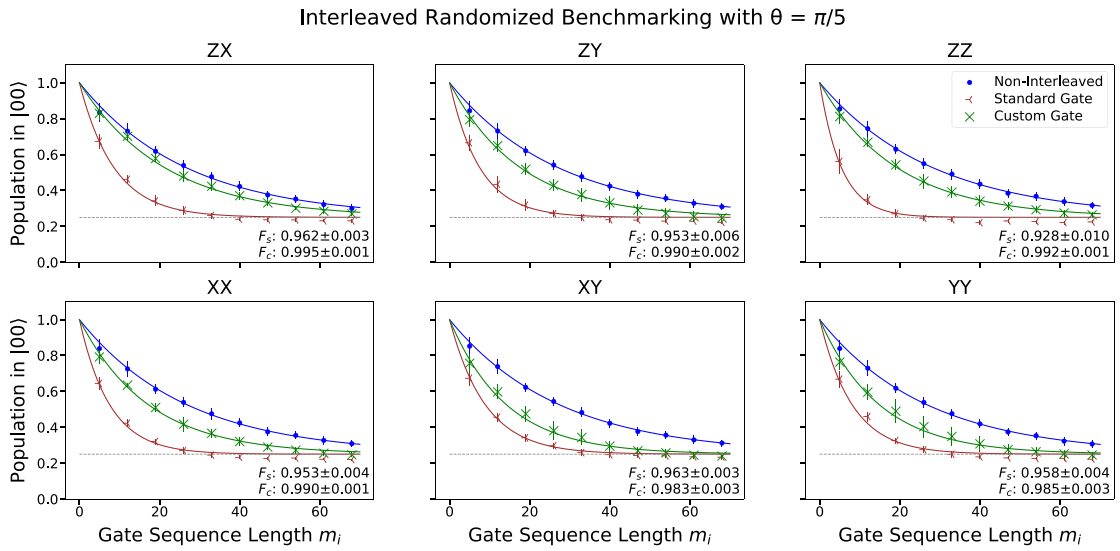


Figure 6. Results of the interleaved randomised benchmarking sequence for six different two-qubit gates in the custom pulse-level and standard circuit-level implementation on a real IBM Quantum backend. Each point indicates the fractional ground state population averaged over ten random gate sequences while error bars show the standard deviations. We observe an exponential decay to the fully mixed state indicated by the dashed line at a fractional ground state population of 0.25. For each gate, F_S and F_C characterise the respective incoherent error of the standard and custom implementation as discussed in the main text. In all cases, we find that $F_C \geqslant F_S$ marks a significant reduction in the incoherent error of the gates.

the circuit identities in figure 4 where necessary. For each gate, we repeat the test with ten random gate sequences using 20 000 repetitions per circuit. The results of our tests are shown in figure 6.

We must be careful in interpreting F_S and F_C as they do not characterise the total gate error but rather the error associated with performing an identity operation by using the $ZX(\theta)$ gate and its inverse. In general, we expect this error to come from coherent errors and noise. Performing additional Hamiltonian tomography sequences for the $ZX(\theta)$ gate and its inverse both in the custom and standard implementation, we see similar coefficients for all terms in the respective Hamiltonians. This rules out the possibility that the large discrepancy between F_S and F_C is due to coherent errors. Also note that the calibration has already been verified in figure 3.

Hence, we can interpret F_S and F_C as characterising the error from noise for the standard and custom $ZX(\theta)$ gate implementations, respectively. With this interpretation, we can verify both hypotheses. First, for the $ZX(\theta)$ gate, we observe that F_C is significantly larger than F_S in figure 6, indicating that our custom implementation, requiring fewer cross-resonance pulses, is less susceptible to noise. And second, we see similar reductions for the incoherent error of the other gates that we tested in figure 6, as expected due to high single-qubit gate fidelities. Therefore, our pulse-level implementation of the $ZX(\theta)$ gate provides us with a wide range of two-qubit gates with a significantly reduced incoherent error. Since the overall pulse schedule time is significantly shorter than the coherence time of the qubits [20], we conjecture that this improvement is due to the simplified pulse architecture we developed, rather than the reduced gate time. The relevant improvement stems from reducing the number of cross-resonance pulses from four to only two.

Finally, we comment on the relevance of this result for practical quantum computing. A reduced incoherent error for two-qubit operations is particularly useful in Hamiltonian Simulation. This often requires the repeated application of multi-qubit gates, for instance in Trotterisation approaches [2], and is thus limited by a high incoherent error of multi-qubit gates. By calibrating a gate using our pulse-level approach, the incoherent error can be significantly reduced. This can enable Hamiltonian Simulation, as we will present in a subsequent paper [27]. Thereby, our procedure is not just relevant from an engineering but also from a physics perspective, as we can use the improved gates to simulate interesting physical systems on publicly available quantum backends.

To conclude, we provide a powerful extension to the set of high-fidelity, multi-qubit gates on currently available quantum computers based on superconducting qubits. With our runtime-efficient and reproducible pulse-level approach, one can calibrate a $\mathrm{CR}(\theta)$ cross-resonance gate for a given value of θ which is extended to a wide range of other two-qubit gates by applying single-qubit gates. We have demonstrated that this pulse-level approach, requiring fewer two-qubit pulses than the circuit-level approach currently used by IBM, significantly reduced the incoherent error of the $\mathrm{CR}(\theta)$ gate and related interactions.

While providing a compelling proof of principle, we were limited to performing demonstrations on publicly available IBM Quantum backends. Future work should focus on repeating our demonstrations on IBM Quantum backends which are currently not available to the general public. This could include comparing our procedure with other methods for calibrating a custom cross-resonance gate. Further, our pulse-level approach should be tested in quantum computing applications to demonstrate the practical usefulness of the improvement. This will be explored for Hamiltonian Simulation in a subsequent paper.

Data availability statement

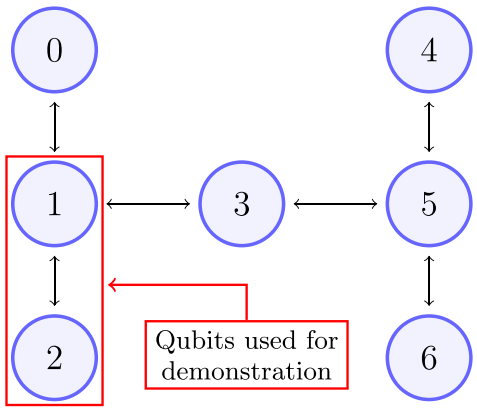
All data that support the findings of this study are included within the article (and any supplementary files).

Acknowledgments

We acknowledge the use of IBM Quantum services for this work. The views expressed are those of the authors, and do not reflect the official policy or position of IBM or the IBM Quantum team. FT acknowledges support from the UKRI New Horizons Grant EP/X017249/1.

Appendix A. Backend configuration and properties

In this appendix, we provide the detailed configuration of the relevant IBM quantum backend and the properties of the specific qubits that we have used for our demonstrations. The calibration data presented here was collected on the same day on which we have performed all our tests. By providing details about the layout and characteristics of the quantum computer which we have used, we aim to enhance the reproducibility of our work.



Backend Configuration	
Backend Name	ibm_oslo
Date	06 March 2023
Processor Type	IBM Falcon r5.11H
Quantum Volume	32
CLOPS	2645

Figure 7. Configuration of the backend used in our demonstrations. All the tests for this paper have been performed on 6 March 2023. The coupling map illustrates the geometrical layout of the computing device. Circles indicate qubits while arrows between circles indicate qubit couplings. Due to their properties and queueing considerations, we have used the directly coupled pair of qubit 1 and qubit 2 in all our tests [20].

In figure 7, we present the configuration of the `ibm_oslo` cloud quantum backend which we have used for our demonstrations. The backend is based on an IBM Falcon r.511H processor with a geometrical configuration and coupling as presented in the figure. Note that the `ibm_oslo` backend has been retired after our tests [8]. However, as other publicly available IBM quantum backends such `ibm_perth` are based on the same processor with an identical configuration [20], it is possible to repeat and extend our tests on these other backends.

In figure 8 we present the qubit properties collected on the day of our tests. We only include the data for qubit 1 and qubit 2 on the `ibm_oslo` backend since we have only used these qubits for our demonstrations.

Qubit Properties of ibm_oslo on 06 March 2023		
Property	Qubit 1	Qubit 2
Frequency (in GHz)	5.046	4.962
Anharmonicity (in GHz)	-0.343	-0.344
Qubit Relaxation Time (in μs)	140.95	119.78
Hahn-Echo Coherence Time (in μs)	35.63	54.61
Readout Duration (in ns)	910.22	910.22
Readout Error	0.021	0.022
Pauli-X Duration (in ns)	35.56	35.56
Pauli-X Error	2.28e-4	2.29e-4
CNOT Duration (in ns)	248.89 (1:2)	213.33 (2:1)
CNOT Error	0.017 (1:2)	0.017 (2:1)

Figure 8. Properties of the qubits used in our tests. The data for qubit 1 and qubit 2 on the ibm_oslo quantum backend was collected on 6 March 2023. Errors are given as fractions and not as percentages. Note that for the CNOT properties, 1:2 and 2:1 indicate that qubit 1 is the control qubit and qubit 2 is the target qubit and vice versa, respectively. Further, note that the Qubit Relaxation Time and Hahn-Echo Coherence Time are often referred to as T1 and T2, respectively [20].

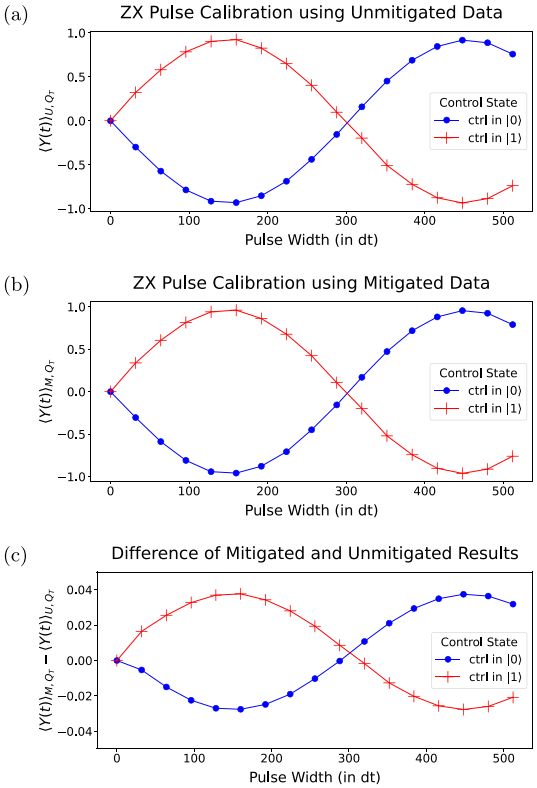


Figure 9. Comparison of the *ZX* pulse verification test results with and without readout error mitigation. (a) Expectation value $\langle Y\rangle_U$ without readout error mitigation using the raw output data. (b) Expectation value $\langle Y\rangle_M$ with error mitigation using the raw output data and classical post-processing. (c) Difference of expectation values $\langle Y\rangle_M$ and $\langle Y\rangle_U$ with and without readout error mitigation. We observe that the error mitigation introduces a minor correction which accentuates the sinusoidal shape. Error bars are smaller than marker size.

For the purposes of reproducibility, we include data that characterises the coherence and control properties of the respective qubits at the time of our demonstrations.

Appendix B. Data analysis techniques

In our demonstrations for this paper, we have used the data analysis technique described in [21, 28] which allows us to correct readout errors by using classical data post-processing methods. Specifically, we have performed a local readout error mitigation sequence using the `LocalReadoutError` class in Qiskit [7] after each calibration and benchmarking test as we expect the readout errors to be largely uncorrelated.

The idea of the local readout error mitigation sequence is to determine the error by preparing the qubits in $|00\rangle$ and $|11\rangle$ and then measure the qubits 4 000 times for each preparation. With this, we can prepare an assignment matrix that models the relationship between the perfect preparation and the actual result, and thus the readout error which is introduced. By inverting the assignment matrix, we can hence create a mitigator that allows us to remove the readout error from the raw output data.

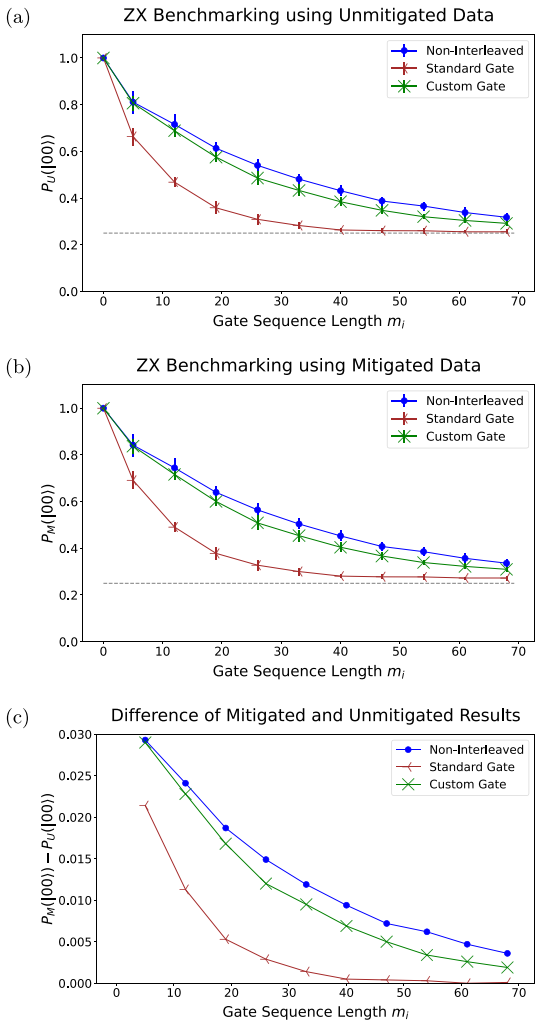


Figure 10. Comparison of the *ZX* benchmarking test results with and without readout error mitigation. (a) Fractional ground state population $P_U(|00\rangle)$ without readout error mitigation using the raw output data. (b) Fractional ground state population $P_M(|00\rangle)$ with error mitigation using the raw output data and classical post-processing. (c) Difference of fractional ground state population $P_M(|00\rangle)$ and $P_M(|00\rangle)$ with and without readout error mitigation. We observe that the error mitigation introduces a minor correction due to the systematic readout error. The correction is significantly smaller than the differences between the decay curves for the standard implementation and our custom implementation. Hence, we conclude that our results are significant as they directly follow from the output data and are not due to the data post-processing which we have performed in all demonstrations.

As we have presented our results based on the mitigated data in the main part of this paper, one might worry whether our results are a consequence of the data post-processing only. In principle, it could be that the readout error mitigation changes the data such that we only find significant confirmation for our hypotheses in the mitigated data but not in the unmitigated data.

To dispel concerns along these lines, we have included an illustrative comparison of our results using the unmitigated data and the mitigated data both for the calibration tests and the interleaved randomised benchmarking tests from the main part. First, we compare the results using the mitigated and the unmitigated data for the *ZX* verification test in figure 9. Second, we compare the results using the mitigated and the unmitigated data for the *ZX* benchmarking test in figure 10. All other calibration and benchmarking tests have similar error mitigation procedures and are hence covered by these two illustrations.

In both cases, we find that the readout error mitigation introduces a shift that is by an order of magnitude smaller than the values of the relevant variables. Hence, we conclude that the readout error mitigation only introduces a minor correction. In particular, for the benchmarking example in figure 10, the difference between the curves that correspond to the standard implementation and our custom implementation of the *ZX* gate is significantly smaller than the differences in shifts introduced by the error mitigation. In other words, the reduced incoherent error is also apparent in the unmitigated data.

Seeing this, one might ask why we should use readout error mitigation at all. There are at least two reasons for this. First, if we have identified an error and know how to mitigate it, then doing so results in more accurate data and hence seems morally appropriate. Second, more practically, error mitigation can lead

to better curve fits and parameter estimates, such as in the calibration experiment in figure 9 where the error mitigation accentuates the sinusoidal shape that we expect theoretically. Hence, using readout error mitigation seems generally advisable.

ORCID iD

Felix Tennie https://orcid.org/0000-0001-9399-710X

References

[1] Montanaro A 2016 *npj Quantum Inf.* **2** 15023
[2] Nielsen M A and Chuang I L 2010 *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press) (https://doi.org/10.1017/CBO9780511976667)
[3] Brooks M 2023 What's next for quantum computing *MIT Technology Review* (available at: https://www.technologyreview.com/2023/01/06/1066317/whats-next-for-quantum-computing/)
[4] Kjaergaard M, Schwartz M E, Braumüller J, Krantz P, Wang J I-J, Gustavsson S and Oliver W D 2020 *Annu. Rev. Condens. Matter Phys.* **11** 369
[5] Glaser S J *et al* 2015 *Eur. Phys. J.* D **69** 279
[6] Koch C *et al* 2022 *EPJ Quantum Technol.* **9** 19
[7] Treinish M *et al* 2023 Qiskit/qiskit: Qiskit 0.42.1 (https://doi.org/10.5281/zenodo.7757946)
[8] IBM Quantum 2023 (available at: https://quantum-computing.ibm.com/)
[9] Alexander T, Kanazawa N, Egger D J, Capelluto L, Wood C J, Javadi-Abhari A and McKay D C 2020 *Quantum Sci. Technol.* **5** 044006
[10] Krantz P, Kjaergaard M, Yan F, Orlando T P, Gustavsson S and Oliver W D 2019 *Appl. Phys. Rev.* **6** 021318
[11] Gambetta J M, Chow J M and Steffen M 2017 *npj Quantum Inf.* **3** 2
[12] Rigetti C and Devoret M 2010 *Phys. Rev.* B **81** 134507
[13] Sheldon S, Magesan E, Chow J M and Gambetta J M 2016 *Phys. Rev.* A **93** 060302
[14] Magesan E and Gambetta J M 2020 *Phys. Rev.* A **101** 052308
[15] Stenger J P T, Bronn N T, Egger D J and Pekker D 2021 *Phys. Rev. Res.* **3** 033171
[16] Satoh T, Oomura S, Sugawara M and Yamamoto N 2022 *IEEE Trans. Quantum Eng.* **3** 1
[17] Garion S, Kanazawa N, Landa H, McKay D C, Sheldon S, Cross A W and Wood C J 2021 *Phys. Rev. Res.* **3** 013204
[18] Magesan E *et al* 2012 *Phys. Rev. Lett.* **109** 080505
[19] The Qiskit Team 2022 Hamiltonian tomography (available at: https://qiskit.org/textbook/ch-quantum-hardware/hamiltonian-tomography.html)
[20] IBM Quantum 2023 IBM quantum resources (available at: https://quantum-computing.ibm.com/services/resources)
[21] The Qiskit Team 2022 Readout mitigation (available at: https://qiskit.org/textbook/ch-quantum-hardware/measurement-error-mitigation.html)
[22] McKay D C, Wood C J, Sheldon S, Chow J M and Gambetta J M 2017 *Phys. Rev.* A **96** 022330
[23] Vatan F and Williams C 2004 *Phys. Rev.* A **69** 032315
[24] Sheldon S, Bishop L S, Magesan E, Filipp S, Chow J M and Gambetta J M 2016 *Phys. Rev.* A **93** 012301
[25] The Qiskit Team 2022 Randomized benchmarking (available at: https://qiskit.org/textbook/ch-quantum-hardware/randomized-benchmarking.html)
[26] The Qiskit Team 2022 Summary of quantum operations (available at: https://qiskit.org/documentation/tutorials/circuits/3_summary_of_quantum_operations.html)
[27] Tennie F, Danin D and Farrow T 2023 in preparation
[28] Bravyi S, Sheldon S, Kandala A, Mckay D C and Gambetta J M 2021 *Phys. Rev.* A **103** 042605