Effects of Dynamical Decoupling and Pulse-Level Optimizations on IBM Quantum Computers

SIYUAN NIU1 AND AIDA TODRI-SANIAL2 (Senior Member, IEEE)

1LIRMM, University of Montpellier, 34090 Montpellier, France
2LIRMM, University of Montpellier, CNRS, 34090 Montpellier, France

Corresponding author: Aida Todri-Sanial (e-mail: aida.todri@lirmm.fr).

This work was supported by the QuantUM Initiative of the Region Occitanie, University of Montpellier and IBM Montpellier.

ABSTRACT Currently available quantum computers are prone to errors. Circuit optimization and error mitigation methods are needed to design quantum circuits to achieve better fidelity when executed on NISQ hardware. Dynamical decoupling (DD) is generally used to suppress the decoherence error, and different DD strategies have been proposed. Moreover, the circuit fidelity can be improved by pulse-level optimization, such as creating hardware-native pulse-efficient gates. This article implements all the popular DD sequences and evaluates their performances on IBM quantum chips with different characteristics for various well-known quantum applications. Also, we investigate combining DD with the pulse-level optimization method and apply them to QAOA to solve the max-cut problem. Based on the experimental results, we find that DD can be a benefit for only certain types of quantum algorithms, while the combination of DD and pulse-level optimization methods always has a positive impact. Finally, we provide several guidelines for users to learn how to use these noise mitigation methods to build circuits for quantum applications with high fidelity on IBM quantum computers.

INDEX TERMS Error mitigation, noisy intermediate-scale quantum (NISQ) hardware.

I. INTRODUCTION Quantum computing is rapidly growing in recent years, and various technologies have been developed for different quantum platforms, the leading candidates being superconducting and trapped ion devices. Several companies, such as IBM, Rigetti, and IonQ, provide publicly available cloud-based services that allow users to access their platforms remotely. But, today’s quantum computers are still prone to errors caused by either unavoidable interactions with the environment or imperfect quantum controls. They are qualified as noisy intermediate-scale quantum (NISQ) computers [33]. The largest quantum chip to date has 127 qubits released by IBM [1]. Quantum error correction (QEC) has been proposed to eliminate the noise impact and help achieve a fault-tolerant quantum device [5], [46]. However, the implementation of QEC codes requires a large number of ancilla qubits, which is not feasible on current hardware. Therefore, alternative approaches are needed to tackle the noise issue.

In order to reduce the noise impact and improve the quantum circuit fidelity, several quantum software packages were designed, and they have made a great contribution to different circuit design processes, such as circuit synthesis [9], [51] or qubit mapping [22], [28]. In addition, quantum error mitigation (QEM) was introduced for error suppression on NISQ devices [12]. There are different QEM techniques, such as readout error mitigation [4], dynamical decoupling (DD) [42], crosstalk mitigation [26], [27], zero-noise extrapolation [23], etc. Most methods require supplementary circuit executions to build the error map for mitigation, whereas DD is one of the simplest strategies, which aims at mitigating the decoherence error (also called idle error) without any circuit overhead and is the focus of this article. The thrust of DD is to insert periodically a series of pulses to the idle qubits and return the qubits to their original states. There is a high probability to have idle qubits during execution, due to the variation of gate latencies and limited parallelism caused by the anticommutative gates. It was shown in [8] that the idle qubits can be almost ten times more subject to errors if adjacent two-qubit operations are executing at the same time on IBM superconducting device. DD plays an important role in reducing the idle error and has been used in quantum volume (QV) experiments [18], noise spectrum characterization [44],
decoherence-protected quantum gate implementation [49], etc. Moreover, there are different DD strategies, such as Hahn echo [39], carr-purcell-meiboom-gill (CPMG) [25], XY4 [50], robust knill dynamical decoupling (KDD) [41], etc., and they have diverse impacts on decoherence error suppression for different quantum devices. It was shown in [32] that DD is able to extend the lifetime of one-qubit states, as well as entangled two-qubit states, for IBM and Rigetti devices using XY4 sequence. But, this sequence is shown to be vulnerable to experimental imperfections, while the use of robust DD was demonstrated to be capable of correcting the pulse errors on Rigetti devices [40].

Recently, IBM released Qiskit Pulse [3], [24] allowing users to design and customize the gate pulse implementations. Some works have already attempted to optimize the pulse controls and reduce the pulse durations with the help of Qiskit Pulse [11], [20], [37]. However, such method often needs additional calibrations, which is time-consuming and requires a deep familiarity with quantum control. Earnest et al. [10] and Stenger et al. [43] proposed a new technique to create more hardware-native pulse-efficient (PE) gates, which improves the gate fidelity without the overhead of extra calibrations.

So far, the aforementioned techniques were only tested separately for limited benchmarks and quantum hardware. Several questions still remain unclear, which are as follows.

1) What are the impacts of different DD sequences on specific quantum algorithms?
2) For a certain benchmark, does the impact of different DD sequences vary across different quantum chips?
3) Will the combination of DD and pulse-level optimization methods further improve the circuit fidelity?

In our work, we address these questions and our main contributions are listed as follows.

1) We explore the performance of different DD sequences on various quantum applications and evaluate the experiments on several IBM devices with different qubit numbers and QVs. To the best of our knowledge, this is the first attempt to illustrate the behavior of applying robust KDD sequence to IBM quantum devices.
2) We combine the DD technique with PE optimization method to demonstrate their benefits on the quantum approximate optimization algorithm (QAOA) to solve the max-cut problem.
3) Based on the experimental results, we provide guidelines and insights for users to apply application-oriented DD and pulse-level optimization techniques.

II. BACKGROUND
A. DYNAMICAL DECOUPLING

DD is widely used in suppressing the decoherence error by reducing the interaction between the system and the environment. Considering a system-bath Hamiltonian $H$ shown in (1), $H_s$ and $H_B$ being the Hamiltonians of the system and the bath, respectively, and the interacting term being $H_{SB}$ [see (2)], where $\sigma^\alpha_i$ is the Pauli matrix acting on qubit $i$, $B^\alpha_{ij}$ is the operator of the environment, and $\alpha \in \{x,y,z\}$. DD aims at reducing the impact of system–environment interaction, and various DD protocols have been developed to improve the performance of quantum computers. Here, we review the main strategies of DD implementations and a summary of these DD sequences, as given in Table 1.

$$H = H_s + H_B + H_{SB}$$  \hspace{1cm} (1)  

$$H_{SB} = \sum_{i=1}^{N} \sum_{\alpha \in \{x,y,z\}} \sigma^\alpha_i \otimes B^\alpha_{ij}.$$ \hspace{1cm} (2)  

1) HAHN ECHO

The spin-echo sequence [39] is used to reduce the inhomogeneous effects from the environmental magnetic field. It applies a $\pi$ pulse to the spin system to inverse the spins after a period of time $t$ and let the system be refocused during the same duration. At that time, $T_2^*$ effects are removed and we can obtain a $T_2$ echo.

2) CARR-PURCELL (CP) AND CPMG

Carr and Purcell [6] proposed a series of $\pi$ pulses separated by a constant interval, known as the CP sequence, to further reduce the effect of self-diffusion in the inhomogeneous field. But, extra pulses can introduce more errors and destroy the state of the system. Therefore, Meiboom and Gill [25] improved the CP pulse by developing the CPMG sequence, retaining the CP pulse, but introducing an additional phase shift in the first pulse to compensate the pulse errors.

3) UHRIG DYNAMICAL DECOUPLING (UDD)

Hahn echo and CP/CPMG are equidistant pulses, while UDD aims at optimizing the $\pi$ sequence based on CP/CPMG by varying the intervals between each pulse [48]. It is proven to be the optimal pulse to suppress low-frequency noise and is insensitive to thermal fluctuations. It outperforms the equidistant DD sequences, especially for systems whose spectral densities have high frequencies with sharp cutoff. For other more general cases, equidistant DD sequences were shown to perform better [2].

4) XY4

All the aforementioned DD sequences only rotate around one single axis. They are exclusively useful when the system–environment interaction is orthogonal to the rotation axis. XY4 is the simplest DD sequence to generally suppress system–environment interaction along three directions [50]. It inserts alternatively $\pi$ rotations around $X$ and $Y$ axes independently of the initial state.

5) XY8 AND XY16

DD can sometimes detriment the fidelity due to the accumulation of errors caused by pulse imperfections. One approach
to reducing the errors is to combine the basic sequence with its inverse for self-correction so that the unwanted terms can be canceled [16]. XY4 is usually chosen as the basic cycle. XY8 is composed of XY4 sequence and its inverse, while XY16 contains XY8 and its inverse.

6) KDD

The other approach to avoid pulse imperfections is to replace each pulse of a DD sequence by a robust composite pulse, which is designed to generate ideal rotations even if there exist pulse imperfections [15]. KDD constructs a DD block using a five-pulse composite π pulse, and combine two of such DD block with and without a phase shift π/2 as (KDDφ, KDDφ+π/2)2 [41]. It is composed of 20 pulses in total.

DD has been demonstrated to have the capability to mitigate the decoherence errors on IBM and Rigetti platforms using XY4 sequences [32]. The experiment is performed as follow. It prepares different initial states by varying the angles of the rotation gates, inserts XY4 sequences, and compares the difference between input and output states. The impacts of different DD sequences on the Rigetti device were reported in [40]. A large number of DD sequences are evaluated, including XY4, XY8, KDD, etc., and quantum process tomography is used to characterize the evolution. KDD is shown to be the most robust pulse sequence against pulse imperfections. Both papers evaluate DD sequences on simple benchmarks instead of real quantum applications, and there is no experiment to illustrate applying KDD to IBM quantum devices. Moreover, the experiments in [8] showed that the naive implementation of DD (inserting DD sequences to all the idle qubits when it is possible) cannot always improve the circuit fidelity. Therefore, Das et al. [8] proposed an adaptive DD framework to estimate the DD impact for each circuit and adjust DD sequence to ensure it improves the circuit fidelity. This method achieves fidelity improvement, but introduces a large overhead of DD impact characterization for a given application. Also, Ravi et al. [35] proposed a variational approach to quantum error mitigation (VAQEM), an approach that dynamically inserts DD sequences for the variational algorithm with the overhead of tuning DD sequence. In this article, instead of carefully adjusting DD sequences for better error mitigation with extra circuit overhead, we exhaustively evaluate DD sequences on extensively used quantum applications, to give a high-level idea about if general DD techniques can really be beneficial for these applications. It is also the first attempt to implement KDD on IBM devices to check its performance. The experiments are performed on various IBM devices with different qubit numbers and QVs to provide general insights about application-oriented DD noise mitigation on IBM quantum devices.

B. PE TECHNIQUE

CNOT gate is the only two-qubit operation included in the basis gates for IBM quantum devices, and its calibrations are provided through the IBM quantum dashboard. It is implemented by an RZX(\frac{\pi}{2}) gate along with some single-qubit gates [7] on the device. RZX gate is realized by the echoed cross-resonance gate [36], which is specific for the IBM fixed-frequency superconducting transmon qubit device, due to its low overhead and high fidelity. When executing a circuit on IBM quantum hardware, every multiqubit operation needs to be transpiled to the CNOT basis, which is not flexible and less efficient.

Therefore, Earnest et al. [10] and Stenger et al. [43] proposed a PE circuit transpilation framework and showed to achieve higher fidelity than CNOT basis transpilation for certain benchmarks. Instead of only using RZX(θ) with a fixed angle θ = \frac{\pi}{2} for CNOT gate, a flexible echoed RZX gate is implemented with arbitrary angle and RZX(θ) = XRZX(-\frac{\pi}{2})XRZX(\frac{\pi}{2}) to enable transpilations to RZX basis. This method does not require any additional pulse calibration, since the calibration of RZX gate can be easily calculated from the CNOT calibrations.

An example of transpiling RZX gate to two different basis gates is shown in Fig. 1. We first transpile the RZX gate to the IBM CNOT-basis gate sets [see Fig. 1(b)], and then transpile again the original circuit to RZX-basis gate sets [see Fig. 1(c)]. The duration of the second circuit is reduced by 51.1% compared to the first gate sets. This technique exposes the echo of the cross-resonance gates, which enables at most one single-qubit gate between each nonechoed RZX gate so that it can shorten the total circuit duration.

Quantum algorithms requiring a lot of two-qubit control-rotation gates, such as RZZ or RYY, which can be directly compiled to RZX gates along with some single-qubit gates, are particularly benefited from this PE transpilation framework, for example QAOA [13], quantum hamiltonian simulation [31], etc.

III. METHODS

We construct DD sequences according to Table 1, and n is the number of repetition time for the basic DD pulse. For Hahn
First, we evaluate the DD effects on IBM quantum computers by applying different DD sequences to various well-known quantum applications, including the Bernstein–Vazirani (BV) algorithm, hidden shift (HS) algorithm, quantum Fourier transform (QFT), and graph state (GS). The basis quantum circuit structures of these quantum applications are shown in Fig. 2. The BV algorithm implements an oracle function $f(x)$, which represents the dot product between $x$ and a secret string $s$, with the objective of finding $s$. The HS algorithm constructs an oracle that encodes two functions: 1) $f$; and 2) $g$, and there exists a secret string $s$ such that $g(x) = f(x + s)$. The goal of the algorithm is to find $s$. QFT is the quantum version of discrete Fourier transform and is the essential part for many other quantum algorithms, such as the Shor’s algorithm [38], quantum phase estimation algorithm [21], etc. GS is a quantum state prepared based on a graph. Specifically, we can build a GS according to a given hardware topology, where there is an edge on the graph when the two qubits are coupled, and the connection is represented by a CZ gate. It can entangle all the qubits of the device and is important for error correction.

Second, we evaluate the performance of combining different DD sequences with the PE transpilation technique. We apply them to QAOA to solve the max-cut problem, since QAOA ansatz is composed of $R_{ZZ}$ gates along with some single-qubit gates, which can be profited from the PE method. We generate randomly 3-regular graphs using ReCirq [34] and random graphs using Networkx with different degrees as our benchmarks. For example, we use 3-reg-4 to represent a 3-regular graph with degree of 4 (qubits) and rand-4-0.5 for a random graph with 4 nodes/qubits and a probability for edge creation of 0.5 (see Fig. 3).

We choose the following metrics for different benchmarks to demonstrate the impact of DD and pulse-level optimization technique on application fidelity.

1) **Probability of Successful Trial (PST)** [45]: This metric is defined by the ratio of the number of trials that give
the expected result to the total number of trials, and higher is better. It is dedicated to the benchmarks with one certain correct result, such as the BV algorithm and the HS algorithm.

2) Jensen–Shannon Divergence (JSD): It is used to measure the similarity between two probability distributions, and lower is better. It is suitable for the GS circuit and QFT, whose output is a distribution.

3) Approximation Ratio [17]: It is specifically designed to evaluate the performance of the QAOA circuit and is defined as \( \langle C \rangle / C_{\text{min}} \), where \( \langle C \rangle \) is the expectation value obtained by the quantum computer and \( C_{\text{min}} \) is calculated by the classical NumpyMinimumEigensolver. We aim at maximizing the approximation ratio and 1 means a perfect solution.

### IV. EXPERIMENTAL RESULTS

We perform the experiments on various IBM quantum devices with different qubit numbers and QVs (see Table 2). The size of the benchmarks varies depending on the quantum device. For the first experiment when evaluating the effects of applying DD sequences to different applications, including BV, HS, QFT, and GS, the circuit size varies from 3 to 6 for IBM Q 7 Jakarta, whereas for other devices, the circuit size changes from 3 to 12, since there are too many noises accumulating to obtain meaningful results for benchmarks with more than 12 qubits. When exploring the performance of combining DD and PE optimization technique on QAOA for the second experiment, the degree of the two types of graphs (3-regular graph and random graph) ranges from 4 to 6 for IBM Q 7 Jakarta and 4 to 12 for other devices. The QAOA ansatz has one layer with parameters initialized randomly and optimized using COBYLA optimizer on the simulator. We only execute the final ansatz with optimized parameters on the quantum hardware. All the benchmarks are compiled using Qiskit with the highest optimization level and executed 8192 times.

There are some limitations when inserting certain DD pulses to the idle time. If the DD sequences contain a large number of pulses, such as XY16 and KDD, it might not be possible to insert them to some small benchmarks whose idle time might not be long enough. Moreover, Hahn echo can only be applied if the inverse of the inserted single \( X \) or \( Y \) gate is able to be absorbed into the neighboring gates to ensure the equivalence of the quantum state. Therefore, we check if the DD pulses are actually inserted to the benchmark for each experiment, and we remove the circuit without any DD pulses inserted from the result.

In order to clearly show the impact of various DD sequences and PE optimization method, we use the relative results for the three metrics. The original benchmark with no optimization method applied is marked as the baseline. PST is divided by the baseline, and we use JSD results to divide the baseline, so that we can obtain the relative results when applying DD sequences. If the relative result is larger than 1, it means that there is an improvement, and larger is better. Whereas for QAOA experiments, we use the difference between approximation ratio with DD sequence, or PE technique, and the baseline as the relative result, due to the possible negative value of the approximation ratio. If the difference is larger than zero, it indicates an enhancement, and also larger is better. Each experiment has been repeated three times and results with similar trends were obtained. We show the average of the three experiments for all the results.

We first demonstrate applying DD sequences to various quantum applications. Hahn echo is not applicable for all the benchmarks. XY16 and KDD are too long to insert for certain small BV circuits (less than 4 qubits). All the DD sequences cannot be applied to the HS algorithm, since the duration of the idle time is always equal to the duration of one single-qubit gate. Thus, the idle time is too short to fit any DD sequences. The relative PST results for BV circuits and the relative JSD results for QFT and GS circuits are shown in Figs. 4–6, respectively.

The quantitative analysis of the relative results is given in Table 3. For the BV algorithm (see Fig. 4), if the BV circuit involves a large number of qubits, the PST fidelities can be dropped dramatically, such that all the PSTs are below 0.1 even with DD applied for error mitigation. Therefore, we only show the results whose PSTs are below 0.1. The relative PST results demonstrate that all the DD sequences are able to enhance the BV circuit fidelity, especially UDD_X. On average, the fidelity is improved by 1.09, 3.82, 2.79, and 1.68 times compared with the baseline for IBM Q Jakarta, Guadalupe, Toronto, and Montreal, respectively, whereas for QFT, inserting DD sequence is more favorable for circuits with more than 5 qubits. Therefore, the relative JSD results on IBM Q 7 Jakarta are not encouraging. Overall, DD can be beneficial for QFT circuits, but are not as stable as for BV circuits. The fidelity is improved by 1.14, 1.18, and 1.28 times on average for IBM Q Guadalupe, Toronto, and Montreal, respectively. For the GS circuit, the performance of DD is also not always steady and different across IBM quantum devices.

### TABLE 2 Information of Different IBM Q Devices

| IBM Q     | Jakarta | Guadalupe | Toronto | Montreal |
|-----------|---------|-----------|---------|----------|
| n         | 7       | 16        | 27      | 27       |
| QV        | 16      | 16        | 32      | 128      |
DD is helpful for GS circuits on IBM Q Guadalupe, but not for other devices, with an increase of fidelity by 1.17 times on average. The variance of the results for different DD techniques becomes smaller for relatively large-scale QFT and GS circuits, which might be due to the accumulating errors.

To conclude, DD can always improve the fidelity for the BV algorithm to different extents across IBM quantum chips, but for other applications, the performance of DD remains uncertain and can be different on various quantum devices. Moreover, for certain benchmark, such as the HS algorithm, the idle time is always short, so that DD is not applicable regardless of the circuit size.

Second, we show the results of using QAOA to solve the max-cut problem for different graphs by employing diverse DD sequences (see Fig. 7) and DD sequences + PE optimization technique (see Fig. 8). The quantitative analysis of the relative results is given in Table 4. Most of the DD sequences can help enhance the approximate ratio for QAOA, especially CP sequence. The approximate ratio of all the graphs is improved by 6.8%, 5.9%, 3.7%, and 3.7% on average.

![Graphs showing relative PST results for BV circuits on different IBM quantum devices](image)

**FIGURE 4.** Relative PST results for BV circuits on different IBM quantum devices. Higher is better. (a) IBMQ_Jakarta. (b) IBMQ_Guadalupe. (c) IBMQ_Toronto. (d) IBMQ_Montreal.

### Table 3: Relative Results of Applying Various DD Sequences to Different Quantum Applications on Four IBM Q Hardwares

| Benchmarks | IBM Q Hardware | Jakarta | Guadalupe | Toronto | Montreal |
|------------|----------------|---------|-----------|---------|----------|
| BV         | 1.1 (UDD_X) 1.05 (KDD) | 1.09 | 4.53 (UDD_X) 3.16 (CPMG) | 3.82 | 3.18 (UDD_X) 2.44 (XY16) | 2.79 | 1.77 (UDD_X) 1.55 (KDD) | 1.68 |
| QFT        | 1.1 (XY16) 0.49 (UDD_Y) | 1.01 | 1.17 (CPMG) 1.06 (UDD_X) 1.14 | 1.25 (KDD) | 1.14 (UDD_X) 1.18 | 1.33 (XY4) 1.22 (UDD_X) | 1.28 |
| GS         | 1 (CP) 0.9 (UDD_Y) | 0.95 | 1.21 (XY4) 1.06 (UDD_Y) 1.17 | 1.06 (CPMG) | 1 (XY8) | 1.03 | 1.02 (CP) 0.96 (XY16) | 0.99 |

Note: If the relative results > 1, it means that the DD sequence has a positive impact on the fidelity. Otherwise, a negative impact is introduced by DD.

### Table 4: Relative Results of Applying Different Methods, Including Only DD, Only PE Method, and DD+PE to QAOA Circuits on Four IBM Q Hardwares

| Methods    | IBM Q Hardware | Jakarta | Guadalupe | Toronto | Montreal |
|------------|----------------|---------|-----------|---------|----------|
| Only DD    | Max. (%) 6.6 (UDD_X) | Min. (%) 2.4 (KDD) | Avg. (%) 6.8 | Max. (%) 7.4 (CP) | Min. (%) 2.5 (KDD) | Avg. (%) 5.9 | Max. (%) 4.4 (CP) | Min. (%) 2.3 (KDD) | Avg. (%) 3.7 | Max. (%) 4.7 (CP) | Min. (%) 1.6 (KDD) | Avg. (%) 4.8 |
| Only PE    | 8.9 | 5.5 | 5.3 | 5.3 | 5.3 | 5.3 |
| DD+PE      | 10.4 (UDD_X) | 6 (KDD) | 8.6 | 9.7 (CP) | 6.8 (KDD) | 8.8 | 7.2 (XY4) | 5.8 (KDD) | 6.7 | 7.4 (CP) | 4.5 (KDD) | 6.5 |

Note: If the relative results > 0, it means an improvement on fidelity. Otherwise, a negative impact is introduced on fidelity.
age for IBM Q Jakarta, Guadalupe, Toronto, and Montreal, respectively. There is no clear relationship between the level of improvement and the size of the benchmark. Comparing to the baseline circuit without applying any error mitigation technique, purely the PE method is already able to raise the approximation value by 8.9%, 5.5%, 5.3%, and 4.8% on average for the four devices. Associating the PE technique with DD sequences can further improve the approximation ratio and the negative impact induced by DD can mostly be canceled. The combination of DD and PE technique can improve the approximation ratio by 26.5%, 49.2%, 81.1%, and 75.7% compared with only applying DD for the four IBM devices.

One interesting thing is that, even though KDD is a robust DD sequence, which is designed to be robust against pulse imperfection and has shown improvements on the Rigetti device, its performance is not as good as expected on IBM quantum devices. For all the quantum applications tested, including the QAOA circuit, there is no growth of the circuit fidelity using KDD compared with other DD strategies, and actually it decreases the circuit fidelity for most of the cases.

V. DISCUSSION
A. DESIGN GUIDELINE
It is important to explore different noise mitigation and pulse-level optimization methods, including DD and PE optimization technique, to better build the quantum circuits. Here, we list several guidelines that can help the community utilize these methods to design circuits with higher fidelity.

1) DD cannot always improve the circuit fidelity and it is highly application-dependent. It is recommended to use DD on specific quantum algorithms, including the BV algorithm and QAOA.
2) When applying DD sequences, it is recommended to check the transpiled circuit to verify if DD is suitable to insert.
3) The robust sequence KDD does not work well on IBM quantum devices for most of the quantum algorithms tested.
4) The PE optimization technique is favorable for QAOA circuits, and combined with DD, it can improve the approximation ratio.

B. FUTURE WORK
As the experiments have demonstrated some discouraging effects when inserting DD sequences for certain quantum applications, exploring the hardware physics behind them becomes important. Application-dependent adaptive DD [8] and variational algorithm-focused VAQEM [35] have been proposed to provide the most beneficial DD sequences, but with large overhead and limited analysis of different DD strategies. A smarter approach for designing appropriate application-oriented DD sequences considering hardware
FIGURE 6. Relative JSD results for GS circuits on different IBM quantum devices. Higher is better. (a) IBMQ_Jakarta. (b) IBMQ_Guadalupe. (c) IBMQ_Toronto. (d) IBMQ_Montreal.

FIGURE 7. Relative approximation ratio of applying DD sequences for the QAOA circuit on different IBM quantum devices. Higher is better. (a) IBMQ_Jakarta. (b) IBMQ_Guadalupe. (c) IBMQ_Toronto. (d) IBMQ_Montreal.
FIGURE 8. Relative approximation ratio of applying DD sequences + PE technique for the QAOA circuit on different IBM quantum devices. Higher is better. (a) IBMQ_Jakarta. (b) IBMQ_Guadalupe. (c) IBMQ_Toronto. (d) IBMQ_Montreal.

physics is left to future works. Recently, DD has been proven to be able to suppress ZZ-crosstalk for a fixed frequency transmon superconducting device [47]. It is interesting to further investigate the performance of DD-based crosstalk suppression at application-level, so that it can contribute to the quantum parallel executions [29], [30]. Finally, the pulse-level optimization experiments can be delved to other quantum applications to provide more guidelines, such as variational quantum eigensolver [19], quantum simulation [14], etc.

VI. CONCLUSION

Today’s quantum hardware is prone to noise in the NISQ era. Therefore, circuit optimization and error mitigation approaches are required to increase the output fidelity. In this article, we focused on two pulse-level circuit optimization methods: 1) DD; and 2) PE optimization technique. First, we implemented various DD strategies on IBM quantum devices, including nonuniversal, universal, and robust ones. Second, we applied these DD sequences to several well-known quantum applications, such as QFT and BV algorithm, to evaluate comprehensively the impact of diverse DD techniques on IBM quantum devices with different characteristics. We also merged DD with the PE transpilation method and investigated them on QAOA circuits to solve the max-cut problem. Based on the experimental results, we found that DD techniques always show positive impact for some benchmarks, while for others, DD demonstrated some discouraging effects. As there is no-overhead free application-oriented DD approach, we provided a list of design guidelines for users to better understand these pulse-level optimization methods and figure out how to improve the circuit design for various quantum applications.

ACKNOWLEDGMENT

The authors would like to thank Adrien Suau for the helpful suggestions and feedback on an early version of this manuscript. All the scripts to reproduce the experimental results are available at https://github.com/peachnuts/DD_PE.

REFERENCES

[1] IBM quantum unveiled eagle, a 127-qubit quantum processor, 2021.
[2] A. Ajoy, G. A. Álvarez, and D. Suter, “Optimal pulse spacing for dynamical decoupling in the presence of a purely dephasing spin bath,” Phys. Rev. A, vol. 83, no. 3, 2011, Art. no. 032303, doi: 10.1103/PhysRevA.83.032303.
[3] T. Alexander et al., “Qiskit pulse: Programming quantum computers through the cloud with pulses,” Quantum Sci. Technol., vol. 5, no. 4, 2020, Art. no. 044006, doi: 10.1088/2058-9565/aba404.
[4] S. Bravyi, S. Sheldon, A. Kandala, D. C. McKay, and J. M. Gambetta, “Mitigating measurement errors in multiqubit experiments,” Phys. Rev. A, vol. 103, no. 4, 2021, Art. no. 042605, doi: 10.1103/PhysRevA.103.042605.
[5] A. R. Calderbank, E. M. Rains, P. M. Shor, and N. J. A. Sloane, “Quantum error correction via codes over GF(4),” IEEE Trans. Inf. Theory, vol. 44, no. 4, pp. 1369–1387, Jul. 1998, doi: 10.1109/18.681315.
[6] H. Y. Carr and E. M. Purcell, “Effects of diffusion on free precession in nuclear magnetic resonance experiments,” Phys. Rev., vol. 94, no. 3, 1954, Art. no. 630, doi: 10.1103/PhysRev.94.630.
[7] J. M. Chow et al., “Implementing a strand of a scalable fault-tolerant quantum computing fabric,” Nat. Commun., vol. 5, no. 1, 2014, Art. no. 4015, doi: 10.1038/ncomms3015.

[8] P. Das, S. Tanu, S. Dengwal, and M. Qureshi, “ADAPT: Mitigating idling errors in qubits via adaptive dynamical decoupling,” in Proc. 54th Annu. IEEE Int. Symp. Microarchitecture, 2021, pp. 950–962, doi: 10.1145/3466752.3460899.

[9] O. Di Matteo and M. Mosca, “Parallelizing quantum circuit synthesis,” Quantum Sci. Technol., vol. 1, no. 1, 2016, Art. no. 015003, doi: 10.1088/2058-9565/1/1/015003.

[10] N. Earnest, C. Tornow, and D. J. Egger, “Pulse-efficient circuit translation for quantum applications on cross-resonance-based hardware,” Phys. Rev. Res., vol. 3, no. 4, 2021, Art. no. 043088, doi: 10.1103/PhysRevResearch.3.043088.

[11] D. J. Egger and F. K. Wilhelm, “Adaptive hybrid optimal quantum control for precisely characterized systems,” Phys. Rev. Lett., vol. 112, no. 24, 2014, Art. no. 240503, doi: 10.1103/PhysRevLett.112.240503.

[12] S. Endo, S. C. Benjamin, and Y. Li, “Practical quantum error mitigation for near-future applications,” Phys. Rev. X, vol. 8 no. 3, 2018, Art. no. 031027, doi: 10.1103/PhysRevX.8.031027.

[13] E. Farti, J. Goldstone, and S. Gutmann, “A quantum approximate optimization algorithm,” 2014, arXiv:1411.4028.

[14] I. M. Georgescu, S. Ashhab, and F. Nori, “Quantum simulation,” Rev. Modern Phys., vol. 86, no. 1, 2014, Art. no. 153, doi: 10.1103/RevModPhys.86.153.

[15] D. M. Grant and R. K. Harris, Encyclopedia of Nuclear Magnetic Resonance, vol. 9, Hoboken, NJ, USA: Wiley, 1996.

[16] T. Gullion, D. B. Baker, and M. S. Conradi, “New, compensated Carr–Purcell sequences,” J. Magn. Reson., vol. 89, no. 3, pp. 479–484, 1990, doi: 10.1016/0022-2364(90)90331-3.

[17] M. P. Harrigan et al., “Quantum approximate optimization of non-planar graph problems on a superconducting processor,” Nat. Phys., vol. 17, no. 3, pp. 332–336, 2021, doi: 10.1038/s41567-020-01105-y.

[18] P. Jurcevic et al., “Demonstration of quantum volume 64 on a superconducting quantum computing system,” Quantum Sci. Technol., vol. 6, no. 2, 2021, Art. no. 025020, doi: 10.1088/2058-9565/ab9519.

[19] A. Kandala et al., “Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets,” Nature, vol. 540, no. 7671, pp. 242–246, 2017, doi: 10.1038/nature23879.

[20] N. Khanjea, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, “Optimal control of coupled spin dynamics: Design of NMR pulse sequences by gradient ascent algorithms,” J. Magn. Reson., vol. 172, no. 2, pp. 296–305, 2004, doi: 10.1016/j.jmr.2004.11.004.

[21] A. Y. Kitaev, “Quantum measurements and the abelian stabilizer problem,” 1995, arXiv:quant-ph/9511026, doi: 10.48550/arXiv.quant-ph/9511026.

[22] G. Li, Y. Ding, and Y. Xie, “Tackling the qubit mapping problem for NISQ-era quantum devices,” in Proc. 24th Int. Conf. Architectural Support Program. Lang. Operating Syst., 2019, pp. 1001–1014, doi: 10.1103/quantum.8.031027.

[23] Y. Li and S. C. Benjamin, “Efficient variational quantum simulator incorporating active error minimization,” Phys. Rev. X, vol. 7 no. 2, 2017 Art. no. 021050, doi: 10.1103/PhysRevX.7.021050.

[24] C. David et al., “Qiskit backend specifications for open-QASM and OpenPulse experiments,” 2018, arXiv:1800.03452, doi: 10.48550/arXiv.1809.03452.

[25] S. Meiboom and D. Gill, “Modified SpinEcho method for measuring nuclear relaxation times,” Rev. Sci. Instrum., vol. 29, no. 8, pp. 688–691, 1958, doi: 10.1063/1.1716296.

[26] P. Murali, D. C. McKay, M. Martonosi, and A. Javadi-Abhari, “Software mitigation of crosstalk on noisy intermediate-scale quantum computers,” in Proc. 25th Int. Conf. Architectural Support Program. Lang. Operating Syst., 2020, pp. 1001–1016, doi: 10.1145/3373376.3373877.

[27] S. Niu and A. Todri-Sanial, “Analyzing crosstalk error in the NISQ era,” in Proc. IEEE Comput. Soc. Annu. Symp. VLSI, 2021, pp. 428–430, doi: 10.1109/ISVLSI5119.2021.00084.

[28] S. Niu, A. Suau, G. Staffelbach, and A. Todri-Sanial, “A hardware-aware heuristic for the qubit mapping problem in the NISQ era,” IEEE Trans. Quantum Eng., vol. 1, 2020, Art. no. 310164, doi: 10.1109/TQEX.2020.3026544.

[29] S. Niu and A. Todri-Sanial, “Enabling multi-programming mechanism for quantum computing in the NISQ era,” 2021, arXiv:2102.05321, doi: 10.48550/arXiv.2102.05321.