2022 IEEE International Conference on Quantum Computing and Engineering (QCE) | 978-1-6654-9113-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/QCE53715.2022.00051

Time-Efficient Qudit Gates through Incremental Pulse Re-seeding

Lennart Maximilian Seifert[†*], Jason Chadwick[‡*], Andrew Litteken[†], Frederic T. Chong[†] and Jonathan M. Baker[†]
[†]Department of Computer Science, University of Chicago
[‡]Department of Physics, Carnegie Mellon University
[*]These authors contributed equally.

***Abstract*—Current efforts to build quantum computers focus mainly on the two-state qubit, which often involves suppressing readily-available higher states. In this work, we break this abstraction and synthesize short-duration control pulses for gates on generalized *d*-state qu*dits*. We present Incremental Pulse Re-seeding, a practical scheme to guide optimal control software to the lowest-duration pulse by iteratively seeding the optimizer with previous results. We find a near-linear relationship between Hilbert space dimension and gate duration through explicit pulse optimization for one- and two-qudit gates on transmons. Our results suggest that qudit operations are much more efficient than previously expected in the practical regime of interest and have the potential to significantly increase the computational power of current hardware.**

***Index Terms*—quantum computing, qudit, quantum optimal control, pulse synthesis**

I. Introduction

Quantum computing traditionally focuses on the realization of noise-robust two-level systems, known as qubits. However, in many quantum architectures, each qubit is embedded in a much larger Hilbert space, with all other energy levels being ignored or suppressed. Qudits, the extension of qubits to d levels, are a promising topic of study with the potential to increase computational power of a machine without needing to add additional logic units. For instance, a single four-state qudit can store the same amount of information as two qubits, and an eight-state qudit can encode the global state of three qubits. There are many proposed or adapted qudit-based quantum algorithms [1]–[5] and qudit-based improvements in quantum circuit compilation [6], [7]. The latter presents a method of using qudits in intermediate steps of a circuit that asymptotically reduces the required number of ancilla in a quantum algorithm at essentially equal circuit depth.

Qudits have been studied in different experimental settings. Both IBM [8] and Rigetti [9] have demonstrated implementations of three-level qutrits on superconducting hardware, while qudits with $d = 7$ states have been successfully realized using trapped $^{40}Ca^{+}$ ions [10].

This work is funded in part by EPiQC, an NSF Expedition in Computing, under award CCF-1730449; in part by STAQ under award NSF Phy-1818914; in part by NSF award 2110860; in part by the US Department of Energy Office of Advanced Scientific Computing Research, Accelerated Research for Quantum Computing Program; and in part by the NSF Quantum Leap Challenge Institute for Hybrid Quantum Architectures and Networks (NSF Award 2016136) and in part based upon work supported by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers.

FTC is Chief Scientist for Quantum Software at ColdQuanta and an advisor to Quantum Circuits, Inc.

Correspondence: lmseifert@uchicago.edu, jchadwick@uchicago.edu

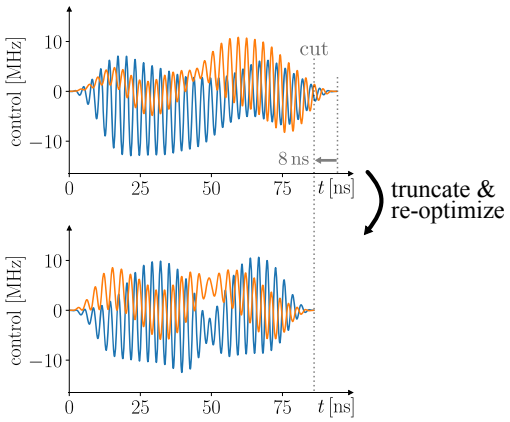

Fig. 1. For quantum optimal control tasks to realize a certain gate or state evolution, truncated control pulses can be used as initial guesses for optimizations of shorter duration. Since they originate from a previous optimization, they already drive the system close to the desired objective. In this figure, the plot on the top shows a high-fidelity pulse being truncated to a shorter duration, and the plot on the bottom is the result of re-optimizing this truncated pulse (achieving the same high fidelity at a shorter duration).

The potential benefits of using qudits are especially important in the near-future NISQ (Noisy Intermediate-Scale Quantum) [11] era of quantum computing, as both total qubit count and maximum circuit depth are severely limited by current hardware. Harnessing qudits could allow current or near-future devices to solve larger, more useful problems sooner than with only qubits. However, qudit gate durations theoretically scale in $\mathcal{O}(h^2)$ time [12], with h being the Hilbert space dimension; this appears to limit the usefulness of higher-dimension qudits, as experimentally achieving the previously-mentioned decreased ancilla or gate count could require much longer individual gate durations.

Quantum logic gates are implemented through external control pulses, meaning that the time it takes to apply a certain gate depends on the duration of the corresponding control pulse. Optimal control software such as GRAPE [13] is designed to find the most accurate control pulse for a certain gate and duration. This means that finding the high-accuracy pulse of *shortest* duration for a given gate requires many independent optimizations at different durations. In this work,

978-1-6654-9113-6/22/$31.00 ©2022 IEEE
DOI 10.1109/QCE53715.2022.00051

we present the Incremental Pulse Re-seeding (IPR) scheme, a method for repeatedly changing the pulse duration and using previous results as new initial guesses for the optimizer, as shown in the example in Figure 1. IPR provides improved shortest-duration results and decreases the dependence on random initial guesses, which is increasingly important for larger Hilbert space dimension.

Using a transmon Hamiltonian, we apply our method to optimizing single-qudit gates of dimension up to eight and two-qudit gates for qudits of dimension up to four to determine the relationship between Hilbert space dimension and shortest gate duration. We recover the theoretical $\mathcal{O}(h^2)$ duration scaling but find it relatively weak for the range of dimensions we consider, giving a promising near-linear relationship.

In Section II we discuss the basics of quantum computation on qubits and qudits. In Section III we explain carrier wave-based optimal control and justify the physical and computational parameters that we choose in our optimizations. In Section IV, we introduce the Incremental Pulse Re-seeding scheme. Finally, we present our optimized pulses and analyze the scaling of gate duration with respect to Hilbert space dimension in Section V.

II. Background

A. Quantum computation with qubits

In this section, we briefly introduce quantum computation on qubits, and then present the more general approach involving qudits. For a more thorough explanation of quantum computing, we refer the reader to [14].

The quantum counterpart to the classical bit is the qubit, the traditional basis of quantum computation. While classical bits can only assume values of either 0 or 1, qubits may exist in a quantum superposition of the corresponding basis states

$$|0\rangle = \begin{bmatrix}1\\0\end{bmatrix} \text{ and } |1\rangle = \begin{bmatrix}0\\1\end{bmatrix}. \tag{1}$$

An arbitrary qubit state is then represented by $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where $\alpha, \beta \in \mathbb{C}$ denote the complex amplitudes of the respective basis states.

Instructions to manipulate the states of qubits are called quantum gates. According to the laws of quantum theory, these gates are reversible and can be represented by unitary operators. A set of commonly used gates are the Pauli operators:

$$X = \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}, \quad Y = \begin{bmatrix}0 & -i\\i & 0\end{bmatrix}, \quad Z = \begin{bmatrix}1 & 0\\0 & -1\end{bmatrix}. \tag{2}$$

Here the X gate corresponds to the quantum analogue of a NOT operation, mapping the state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ to $X|\psi\rangle = \alpha|1\rangle + \beta|0\rangle$. Useful gates for creating superposition or manipulating the relative phases of qubits are the Hadamard gate H and the T gate, respectively:

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}, \quad T = Z^{1/4} = \begin{bmatrix}1 & 0\\0 & e^{i\pi/4}\end{bmatrix}. \tag{3}$$

Two-qubit gates involve logical operations on several qubits at once, with the most common example being the CNOT (controlled NOT) gate:

$$\mathrm{CNOT}\,|q_1\rangle\,|q_2\rangle = |q_1\rangle\,|q_2 \oplus q_1\rangle\,. \tag{4}$$

Like its classical version, this gate flips the state of the target qubit $|q_2\rangle$ if the control qubit $|q_1\rangle$ is in state $|1\rangle$. The CNOT gate plays a particularly important role in quantum circuits, as it has the power to create entanglement between qubits. Swapping the states of two qubits can be realized with a SWAP gate:

$$\mathrm{SWAP}\,|q_1\rangle\,|q_2\rangle = |q_2\rangle\,|q_1\rangle\,. \tag{5}$$

The H, T, and CNOT gates comprise a universal gate set, meaning any arbitrary qubit circuit can be decomposed into a sequence of these gates.

B. From qubits to qudits

In many proposed quantum hardware systems, such as superconducting or ion trap computers, each quantum logical unit has an infinite spectrum of energy levels. The standard qubit abstraction suppresses the other states. Instead, we consider d-dimensional qu*dits*, which each consist of a superposition of d computational basis states, expressed as

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle + \cdots + \alpha_{d-1}|d-1\rangle = \sum_{k=0}^{d-1}\alpha_k|k\rangle\,. \tag{6}$$

In this work we consider the generalized gates [15], [16]

$$\begin{aligned} X_d|k\rangle &= |k+1 \bmod d\rangle\,,\\ X_d^s &: |0\rangle \leftrightarrow |d-1\rangle\,,\\ H_d|k\rangle &= \frac{1}{\sqrt{d}}\sum_{j=0}^{d-1}\omega_d^{kj}|j\rangle\,,\\ T_d|k\rangle &= \omega_d^{k/4}|k\rangle\,, \end{aligned} \tag{7}$$

where $\omega_d = e^{2\pi i/d}$. X_d maps a computational basis state to the next higher one, and maps $|d-1\rangle$ back to $|0\rangle$. As an alternative generalization, X_d^s swaps the amplitudes of the lowest and the highest state. The generalized Hadamard gate H_d creates an equal-population superposition with different relative phases depending on the input state. In analogy to the qubit case, we define the T_d as the fourth root of the generalized Z_d gate (where $Z_d|k\rangle = \omega_d^k|k\rangle$), such that it preserves populations of the basis states but introduces phase differences. For the above equations, the $d = 2$ case produces the qubit gates as defined in (2) and (3).

Analogously, two-qudit gates can be generalized as well. For example, the CNOT operation from (4) can be generalized to the SUM gate [15] or other useful operations depending on use case [6], [16]. Here we focus on the generalized qudit SWAP gate, the straightforward extension of (5), which fully swaps the amplitudes of two qudits.

C. Quantum optimal control

Quantum systems can be purposefully steered by applying external control fields, which are specific to particular hardware. For instance, in superconducting architectures, the state of qubits are manipulated through analog microwave pulses. Generally, the evolution of a quantum system is determined by its time-dependent Hamiltonian

$$\mathcal{H}(t) = \mathcal{H}_0 + \mathcal{H}_c(t). \tag{8}$$

Here $\mathcal{H}_0$ denotes the system's intrinsic drift Hamiltonian and $\mathcal{H}_c(t) = \sum_k f_k(t)\mathcal{H}_k$ denotes the control Hamiltonian, which is typically described by control operators $\mathcal{H}_k$ and classical tunable control fields $f_k(t)$. Quantum optimal control aims to find the optimal control paths $f_k(t)$ to realize a desired state transition or target unitary. This is achieved by repeatedly solving the Schrödinger equation and adjusting the control fields in every iteration to minimize a certain objective function. Different algorithms and toolboxes have been designed for this purpose [13], [17], [18].

III. Setup

In this section we motivate the physical and computational parameters that we have chosen to use in our optimizations. We also introduce the concept of control pulses parameterized through B-splines and carrier waves, which is a key aspect of the optimizer we use.

A. Model Hamiltonian

We consider a system based on superconducting hardware that consists of two weakly coupled, anharmonic transmons [19] with drift Hamiltonian

$$\begin{aligned}\mathcal{H}_0 = &\sum_{k=1}^{2}\left[\omega_k a_k^\dagger a_k + \frac{\xi_k}{2}a_k^\dagger a_k^\dagger a_k a_k\right] \\ &+ J(a_1^\dagger a_2 + a_2^\dagger a_1).\end{aligned} \tag{9}$$

In order to describe qudits, we typically truncate the ladder operators a_k and $a_k^\dagger$ at level $d+2$. Including two additional guard states allows us to penalize population leakage into higher energy states, while we have found that additional guard states beyond two provide marginal fidelity improvements for higher computational cost. We choose realistic physical parameters inspired by [20]: The 0-1 transition frequencies of the transmons are $\omega_1/2\pi = 4.914\,\mathrm{GHz}$ and $\omega_2/2\pi = 5.114\,\mathrm{GHz}$, and both transmons have the same anharmonicity $\xi_1/2\pi = \xi_2/2\pi = -0.330\,\mathrm{GHz}$. They are effectively coupled with $J/2\pi = 3.8$ MHz. While our specific results depend explicitly on these parameters, our methods apply to systems with very different parameters. Control of our model system is possible through microwave drives that add or remove single excitations, as described by

$$\mathcal{H}_c(t) = \sum_{k=1}^{2} f_k(t)(a_k + a_k^\dagger). \tag{10}$$

To numerically solve the Schrödinger equation, it is often helpful to slow down the time variation by applying the rotating wave approximation. In a rotating frame with equal angular frequency ω_r for both qudits, we thus obtain the full transformed Hamiltonian

$$\begin{aligned}\tilde{\mathcal{H}}(t) &= \tilde{\mathcal{H}}_0 + \tilde{\mathcal{H}}_c(t) \\ &= \sum_{k=1}^{2}\left[(\omega_k - \omega_r)a_k^\dagger a_k + \frac{\xi_k}{2}a_k^\dagger a_k^\dagger a_k a_k\right] \\ &\quad + J(a_1^\dagger a_2 + a_2^\dagger a_1) \\ &\quad + \sum_{k=1}^{2}\Big[p_k(t)(a_k + a_k^\dagger) + iq_k(t)(a_k - a_k^\dagger)\Big].\end{aligned} \tag{11}$$

The rotating frame control functions $p_k(t)$ and $q_k(t)$ are then related to the lab frame controls through $f_k(t) = 2\,\mathrm{Re}\{(p_k(t) + iq_k(t))e^{i\omega_r t}\}$ [17].

For the case that we want to describe only a single transmon, we disregard the transmon at index 2 in the above equations and restrict ourselves to a single anharmonic oscillator with one lab frame control field $f_1(t)$.

B. Quantum optimal control with quadratic B-splines

Juqbox [17], [21] is an open-source software package designed to solve quantum optimal control problems in closed systems under the rotating wave approximation. In this section we briefly summarize the differences between Juqbox and conventional optimizers such as GRAPE [13], [22]. While algorithms like GRAPE directly adjust the value of a control pulse at each discrete point in time, Juqbox instead parameterizes the pulse with B-splines. The control functions $p_k(t)$ and $q_k(t)$ in the rotating frame are given by a sum of N_f carrier waves with fixed angular frequencies $\Omega_{k,j}$, where each carrier wave has an amplitude envelope $S_b(t)$ that consists of N_b B-splines:

$$\begin{aligned}p_k(t,\vec{\alpha}) &= \sum_{j=1}^{N_f}\sum_{b=1}^{N_b}\mathrm{Re}\{\alpha_{k,j,b}\,e^{i\Omega_{k,j}t}\}S_b(t) \\ q_k(t,\vec{\alpha}) &= \sum_{j=1}^{N_f}\sum_{b=1}^{N_b}\mathrm{Im}\{\alpha_{k,j,b}\,e^{i\Omega_{k,j}t}\}S_b(t).\end{aligned} \tag{12}$$

Juqbox finds optimal pulses by adjusting the $2N_fN_b$ real coefficients $\alpha_{k,j,b} = \alpha_{k,j,b}^{(\mathrm{re})}, +i\alpha_{k,j,b}^{(\mathrm{im})}$ for the quantum devices indexed by k. The collection of these design variables is denoted by $\vec{\alpha}$. The benefit of this approach is that the B-spline parameterization drastically reduces the dimensionality of the optimal control problem, especially for long-duration pulses, and allows carrier wave frequencies to be chosen to specifically address the target state evolutions while avoiding undesired transitions. Figure 2 shows an example pulse defined by a single carrier wave and six B-splines.

The user of this tool must specify a number of physical and numerical parameters for the optimizer, the most significant of which are summarized in Table I and discussed in the following subsection.

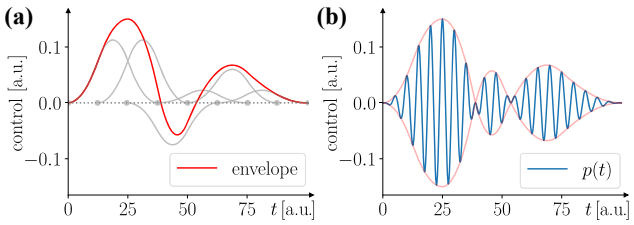

Fig. 2. Real part of an exemplary pulse in the rotating frame, with one carrier wave ($N_f = 1$). (a) Six B-splines shape the envelope of the pulse. Grey dots mark the support of each time-local basis function. (b) The final pulse is constructed from a single carrier wave bounded by the B-spline envelope. With $N_f > 1$, the final pulse is the sum of these individual pulses with one carrier wave.

TABLE I
IMPORTANT PARAMETERS FOR OPTIMIZING IN JUQBOX

Symbol	Description
T	Total pulse duration
$\omega_{r,k}$	Rotating frame frequencies
$\tilde{\mathcal{H}}_0$	Drift Hamiltonian in the rotating frame
$\tilde{\mathcal{H}}_c$	Control Hamiltonian in the rotating frame
$\Omega_{k,j}$	Carrier frequencies in the rotating frame
N_b	B-spline basis functions per carrier wave
V	Target unitary gate
α_{max}	Bound on magnitude of B-splines
`error_threshold`	Goal infidelity
`max_iter`	Maximum optimization iterations

After specifying the required parameters, Juqbox applies gradient-based methods to minimize an objective function consisting mainly of the trace infidelity

$$J = 1 - \frac{1}{h^2}\left|\mathrm{Tr}\left\{U_T^\dagger(\vec{\alpha})V\right\}\right|^2 \tag{13}$$

between the target unitary V and the applied transformation $U_T(\vec{\alpha})$, where h is the Hilbert space dimension (without guard states). The quantum optimal control task is successful if a set of coefficients $\vec{\alpha}$ is found that achieves an infidelity below a desired error threshold. Throughout this work we will also use fidelity $F = 1-J$ to quantify the quality of pulses. For a given target V, certain values of duration T will be too small for the optimizer to find a high-fidelity pulse, in which case a larger T is needed. In our work, the "gate duration" for a gate V refers to the shortest T for which the optimizer converges to our target fidelity.

C. Choosing optimization parameters

We observe that for high-dimensional quantum optimal control problems, it is generally difficult to specify a set of optimization parameters that guarantees finding high-fidelity pulses. Parameters such as N_b, the number of B-splines per pulse, must be large enough to give the optimizer freedom, while not so large as to introduce unnecessary local minima in the highly non-convex search space. In the following subsection, we describe and justify the decisions we made to choose significant optimization parameters.

The carrier frequency and B-spine envelope parameterization of the control pulse has two benefits: It reduces the dimensionality of the pulse, and additionally allows us to easily target the desired state transitions in the qudits by using carrier frequencies directly corresponding to the qudit states. In the lab frame, the N_f carrier frequencies

$$\Omega_{k,j}^{\mathrm{lab}} = \omega_k + j\xi_k, \quad j = 0, \ldots, d-2 \tag{14}$$

correspond to the resonant frequencies of qudit k. For instance, $\Omega_{1,2}^{\mathrm{lab}}$ is the frequency of the $|2\rangle \to |3\rangle$ transition for qudit 1. For a two-qudit gate, each control pulse includes the complete set of $2(d-1)$ carrier frequencies in order to make use of the cross-resonance effect [23], [24], where driving one transmon at a resonant frequency of the other transmon triggers a state transition in the latter. We choose not to correct these frequencies to account for the weak coupling between the transmons, since the small corrections fall inside the widths of the respective Fourier peaks of pulses with finite duration. From experience, we have found that considering all $(d-1)^2$ true resonant frequencies significantly increases the complexity of the problem without providing a noticeable improvement.

Given the fixed set of $2(d-1)$ lab frame carrier frequencies $\Omega_{k,j}^{\mathrm{lab}}$, we choose the average frequency as the rotating frame frequency $\omega_r = (\Omega_{\max}^{\mathrm{lab}} + \Omega_{\min}^{\mathrm{lab}})/2$, which is equal for both qudits (the coupling term is then constant in time, as shown in (11)). This minimizes the magnitude of the carrier frequencies in the rotating frame, allowing the optimizer to choose larger discrete time steps and decreasing overall optimization run time. As an example we consider two qutrits ($d = 3$). The lab frame carrier frequencies are $\Omega_{k,j}^{\mathrm{lab}}/2\pi \in \{4.914, 4.584, 5.114, 4.784\}$ GHz, where the first two correspond to the two resonant frequencies of transmon 1 and the latter two to the resonant frequencies of transmon 2. The rotating frame frequency is $\omega_r/2\pi = (5.114 + 4.584)/2\,\mathrm{GHz} = 4.849\,\mathrm{GHz}$. Thus, the carrier frequencies in the rotating frame are given by $\Omega_{k,j}/2\pi = (\Omega_{k,j}^{\mathrm{lab}} - \omega_r)/2\pi \in \{0.065, -0.265, 0.265, -0.065\}$ GHz.

The number of B-spline basis functions N_b determines how fast the pulse envelope can vary over time (see Figure 2). While a larger number may enable shorter-duration pulses, this will increase the dimensionality of the problem, as well as introduce non-carrier-frequency control pulse oscillations. We choose $N_b = [T/(10\,\mathrm{ns})] + 2$ (where $[\,.\,]$ denotes the nearest-integer function) for two reasons. First, we want to maintain an approximately constant B-spline density when varying T for consistency between different optimization problems. Second, this choice guarantees a minimum envelope rise time of around 15 ns, which is realistic in experiment [20]. The term $+2$ accounts for B-splines on the time domain boundaries, which are always set to amplitude 0 in Juqbox's implementation to ensure that the final pulse starts and ends at 0.

The parameter $\alpha_{\max}$ limits the amplitudes of the individual B-splines, effectively limiting the maximum amplitude of the final pulses. This is needed to ensure weak driving [25], which increases the accuracy of approximate models like (11). Additionally, limited power reduces the risk of leakage into guard levels due to off-resonant transitions. We aim for lab

frame pulse amplitudes of at most $40\,\mathrm{MHz}$ and achieve this by tuning $\alpha_{\max}/2\pi = 40/(2\sqrt{2}N_f)\,\mathrm{MHz}$.

We set `error_threshold`, the target gate infidelity, to 10^{-3}, corresponding to a fidelity of 99.9%. The optimization terminates if the pulse fidelity reaches this target.

Finally, the parameter `max_iter` limits the number of iterations of the optimizer if it does not terminate early due to reaching `error_threshold`. This variable must be set sufficiently large to ensure that the optimizer has converged very nearly to a local minimum, and is generally set experimentally depending on the dimension and target unitary (typically in the range $200 \ldots 1000$).

IV. Gate duration optimization

In this section we describe our method for finding the shortest-duration, high-fidelity pulse for a given target unitary. Optimal control software generally works with a fixed pulse duration, so finding the shortest high-fidelity pulse involves repeated optimizations at different durations.

By default, each optimization begins with a random guess for the coefficient vector $\vec{\alpha}$ within some bounds for each coefficient (typically $\pm\alpha_{\max}/100$ or $\pm\alpha_{\max}/10$). Due to the non-convex nature of the optimization space, it is generally good practice to try multiple initial guesses if the first does not work and the fidelity seems relatively close to the goal fidelity.

A. Naive approaches to duration optimization

We first introduce two simple approaches to finding optimal gate durations, the exhaustive method and the binary search method, and explain their downsides. The exhaustive process begins at some low $T = T_{\mathrm{start}}$ and increases T at a fixed interval, running an independent random-guess optimization for each duration, until one value of T yields a high-fidelity pulse. While this method will work (at least for low-dimension qudits), it is computationally inefficient, motivating the second approach: a binary search. This method involves specifying $T_{\min}$ and $T_{\max}$ and then optimizing in a binary search pattern within the bounds, decreasing T if optimization succeeds and increasing if it fails (still running an independent, random-guess optimization at each duration) until reaching the specified granularity.

While these approaches are relatively effective for small qubit-based gates, they both have problems when transitioning to high-dimensional qudit gates. Larger qudit dimension significantly increases computational cost of solving the optimization, which means the exhaustive method can potentially be very computationally expensive if the starting T is far below the shortest convergent duration. The binary search method is more efficient, but individual long-duration optimizations near $T_{\max}$ will still be quite expensive if $T_{\max}$ is large.

However, the more significant issue is the increased complexity of the optimization space for higher qudit dimensions. We find that random guesses become less effective at consistently converging to the optimal solution for a given T, as we show in Section IV-D.

B. Incremental Pulse Re-seeding

We have developed an optimization scheme to address both of these problems. The key idea is to reuse the coefficient vector $\vec{\alpha}$ from failed optimizations as a seed for the next. For example, if a pulse of duration T_0 converges to a fidelity of 99.5%, we can reuse this pulse as the starting point for an optimization at duration $T_0 + \delta$, which we find to have a better chance of converging to 99.9% fidelity than a random guess.

Depending on the success of the optimizer to find a high-fidelity pulse for a certain duration, our algorithm changes the duration for the next iteration by a discrete time step and reuses the previous pulse (by extending or truncating it to fit the new duration) as a seed. The step size decreases as the algorithm approaches the final solution. We therefore call this scheme *Incremental Pulse Re-seeding* (IPR). The variables used in IPR are listed in Table II, and a flowchart of the method is shown in Figure 3. We denote an optimized pulse as "successful" if it has fidelity greater than our goal fidelity, and "failed" if not.

TABLE II
Variables used in Incremental Pulse Re-seeding

Symbol	Description
T_{start}	First duration tried by optimizer
`step`	Time step to increase/decrease by
`granularity`	Minimum step size
$\vec{\alpha}_{\mathrm{guess}}$	B-spline coefficients of initial guess pulse
`error_threshold`	Target infidelity

The program starts from an initial duration T_{start} with a random guess $\vec{\alpha}_{\mathrm{guess}}$. A flag variable `found_time`, which indicates if a successful pulse has been found before, is set to `false`. The optimizer then converges to a solution $\vec{\alpha}$ with fidelity F. If the converged pulse is of sufficiently high fidelity, the solution is stored as $\vec{\alpha}_{\mathrm{best}}$, the duration is stored as T_{best}, and `found_time` is set to `true`. We then reduce the gate duration by the step size and re-seed the optimizer with a truncated version of $\vec{\alpha}_{\mathrm{best}}$ that fits the new duration. Figure 1 shows an example of a pulse being truncated and re-optimized.

If the target fidelity is not met, but a successful pulse has been found before, we decrease the step size and repeat the above steps, starting from $T_{\mathrm{best}} - \mathtt{step}$ with $\vec{\alpha}_{\mathrm{guess}} = \mathtt{truncate}(\vec{\alpha}_{\mathrm{best}})$. This only happens if the step size is larger than the predefined granularity limit (in our case $1\,\mathrm{ns}$); otherwise, the algorithm terminates and returns the converged result with the shortest duration.

If the optimized pulse does not meet the target fidelity, and no successful pulse has been found before, we increase the duration and re-seed the optimizer with an extended version of the previous pulse, for as long as the fidelity increases with each optimization. If the fidelity decreases (compared to a shorter duration) before any solution is found, we start the entire procedure with a new random $\vec{\alpha}_{\mathrm{guess}}$. In this case we set $T_{\mathrm{start}} = T_{\mathrm{best}}$, the duration for which fidelity was highest.

C. Example: H_4 gate

As an example, we present an application of Incremental Pulse Re-seeding to find the optimal duration of the $d = 4$

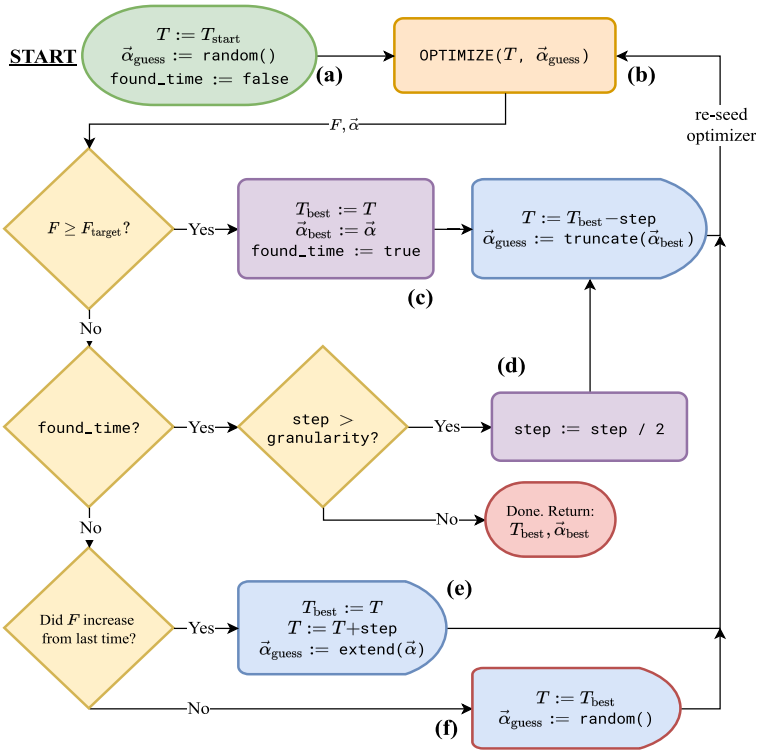

Fig. 3. The Incremental Pulse Re-seeding process. (a) The problem is initialized with a random pulse and duration guess T_{start}. (b) The main optimization function, which searches for the highest-fidelity pulse for a given duration using Juqbox, returning an optimized pulse $\vec{\alpha}$ and a fidelity F. (c) If an optimization succeeds, save the duration and pulse, then decrease T and start from a truncated version of the successful pulse. (d) If an optimization fails but a successful pulse has been found in the past, decrease step size and work backwards from T_{best} again; if step size has reached granularity, return best pulse. (e) If no successful pulse has been found so far, but fidelity improved, mark this duration as the best so far, extend the duration, and re-seed with the extended previous pulse. (f) If no successful pulse has been found and fidelity decreased, restart the whole process with the highest-fidelity duration and a random guess.

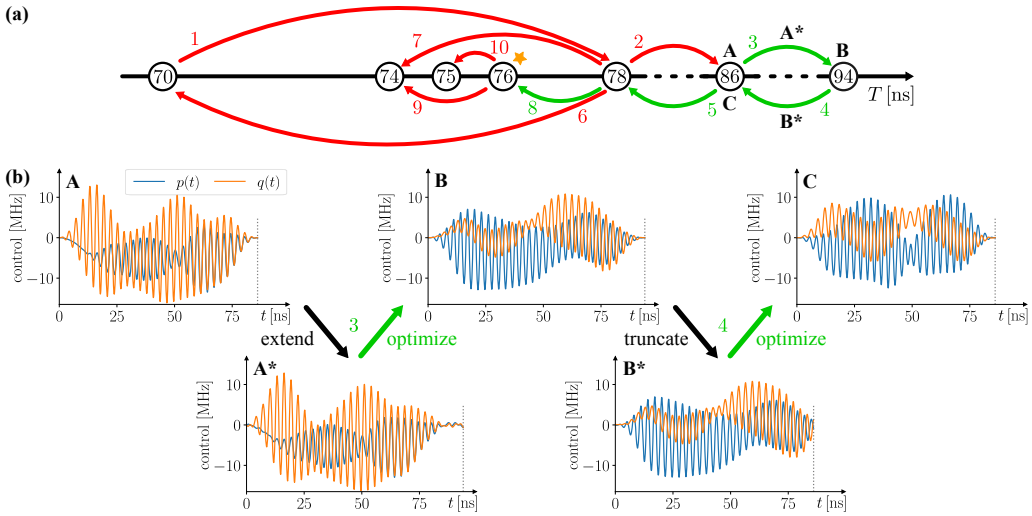

Fig. 4. Applying Incremental Pulse Re-seeding (IPR) to find the shortest duration for the H_4 gate. (a) Visualization of attempted durations during the IPR procedure, with the chronological order of steps indicated by the colored numbers. Green (red) arrows represent steps that did (did not) lead to meeting the pulse target fidelity of 99.9%. Starting from $T_{\text{start}} = 70$ ns and a time step of 8 ns, three pulse extension and re-optimization steps are required before the target fidelity is first reached at 94 ns. The duration is then reduced to 78 ns by successfully re-seeding with truncated pulses. After step 6 the time step is halved because re-optimizing at $T = 70$ ns with the truncated pulse does not meet the target fidelity. Repeating this scheme leads to the final duration of $T_{\text{gate}} = 76$ ns, highlighted by the star, as after step 10 the time step falls below the granularity of 1 ns. (b) Plot A shows the optimized pulse after IPR step 2. A new seed for the optimization at $T = 94$ ns is generated by extending the previous result in duration (A*). At 94 ns the target fidelity is met, so the optimized pulse is truncated (B*) and used as a seed for the second $T = 84$ ns optimization, leading to the successful result shown in C.

Hadamard gate H_4 (see (7)). Figure 4(a) shows the sequence of IPR steps, where the chronological order is given by the numbered arrows. Red arrows represent steps that failed to reach the target fidelity of 99.9%, while green arrows indicate success. We start at $T_{\text{start}} = 70\,\text{ns}$ with an initial time step size of 8 ns and a random guess for the coefficient vector $\vec{\alpha}$. The fidelity of the optimized pulse falls short of the target, so IPR extends the output pulse and uses it to seed the optimizer at a longer duration (step 1). This occurs two more times (steps 2 and 3) before a pulse that reaches the target fidelity is found at $T = 94\,\text{ns}$. In step 4, this solution is then truncated and used to revisit $T = 86\,\text{ns}$. This time the optimization task is successful, so the duration can be further decreased (step 5), resulting in another high-fidelity pulse at 78 ns. $T = 70\,\text{ns}$ fails to converge to a the target fidelity, so the time step is halved and IPR tries 74 ns instead (step 7). After few more steps, the minimum step granularity is reached, and $T_{\text{gate}} = 76\,\text{ns}$ is found to be the shortest duration to realize the H_4 gate with 99.9% fidelity. It is important to note that durations 78 ns and 86 ns both initially resulted in failed optimizations, but then succeeded when re-seeded with truncated successful pulses.

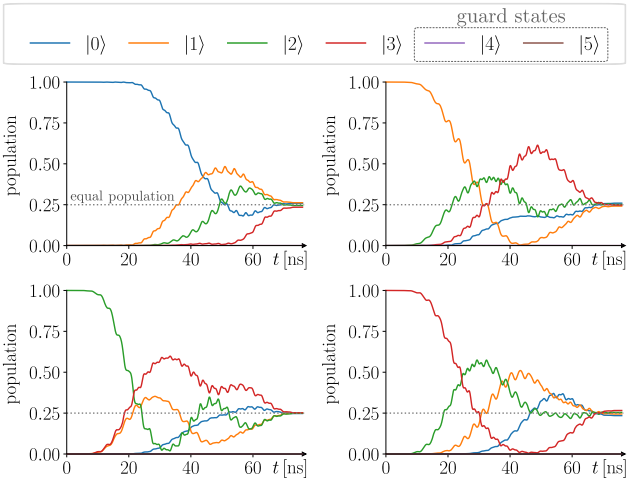

Fig. 5. Evolution of state populations when applying the H_4 pulse of shortest duration (76 ns) to a qudit with $d = 4$. Every computational basis state is transformed into superposition of all four basis states with nearly equal population, deviations only arising from the remaining 0.1% infidelity. At all times the populations of the guard states $|4\rangle$ and $|5\rangle$ are below 2×10^{-3} and 10^{-6}, respectively.

With the help of Figure 4(b), we take a closer look at steps 3 and 4 to explain how pulses are extended and truncated. Initially (plot A), we have an optimized 86 ns pulse that falls short of the target fidelity. The controls $p(t)$ and $q(t)$ are numerically extended by appending 8 ns of idle time (zero amplitude pulse). The number of B-spines N_b changes due to the duration change, so we obtain the new coefficient vector $\vec{\alpha}_{\text{guess}}$ by applying a least-squares fit according to (12) with the new number of B-splines. The approximation introduces small distortions, as can be seen in plot A*, because the manually modified control functions are not perfectly realizable with the given parameterization. The following optimization (plot B) meets the target fidelity, so the pulse is truncated to 86 ns and re-parameterized (plot B*). Plot C shows the converged pulse after step 4, which successfully reaches 99.9% fidelity. Together, the series of plots (where A and C are both at duration 86 ns) shows how a new minimum was found after visiting 94 ns and then revisiting 86 ns with a better guess.

Figure 5 visualizes the evolution of the basis state populations for the shortest pulse of 76 ns. It is clearly visible how an equal-population superposition emerges in every case, and the small imperfections at the final time come from the remaining 0.1% infidelity. The populations of the guard states cannot be seen because they are suppressed by multiple orders of magnitude. This guard state suppression is achieved by the carrier wave parameterization method, which allows us to selectively induce state transitions.

D. Comparison with random guessing

For the X_8 gate (see (7)), the IPR scheme finds a shortest-duration pulse of 195 ns with target fidelity 99.9%. To compare our approach with a naive approach, we run 20 random-guess, single-duration Juqbox optimizations at every 5 ns between 180 ns and 230 ns, where the components of $\vec{\alpha}_{\text{guess}}$ are sampled from $[-0.1\alpha_{\max}, 0.1\alpha_{\max}]$. Figure 6 shows the distribution of pulse fidelities achieved by this naive random guessing across a range of durations for the same gate, with 20 random attempts at each duration.

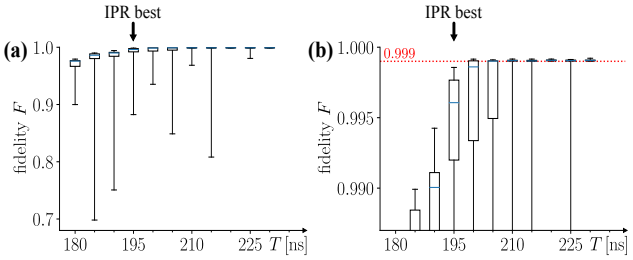

Fig. 6. Distribution of naive random-guess optimizations of the X_8 gate for invididual durations between 180 ns and 230 ns. Each boxplot consists of 20 fidelities from 20 individual Juqbox optimizations, with the blue line indicating the median value, the box indicating the middle 50% of fidelities, and the whiskers indicating the full range of values. (a) The full data, showing a general increase in fidelity as duration increases. (b) Focusing on the area of interest (fidelities close to 99.9%). The IPR scheme outperforms random guessing, finding a minimum duration of $T_{\text{best}} = 195\,\text{ns}$ at 99.9% fidelity, whereas none of the 20 random guesses at that duration reached the same fidelity.

It is evident that there is significant variation in the naive approach; no random-guess optimizations converged to 99.9 fidelity at 195 ns, and only 8 out of 20 random guesses converged to 99.9% fidelity at 200 ns. Even for larger durations such as 215 ns, not all guesses converged to the target fidelity. This is clearly a problem for binary-search-based methods, which implicitly assume that a given duration will either always succeed or never succeed. In contrast, over ten IPR runs (with T_{start} sampled from $[160, 240]$ ns; see Section V-A), we find an average duration of 198.6 ns and a standard deviation of 3.0 ns. This small standard deviation given the large range

of starting times is evidence that our method is more stable than random guessing for large Hilbert space optimizations.

V. Qudit pulse results

Previous work [6], [7] has shown that breaking the qubit abstraction in quantum circuits can lead to substantially reduced circuit complexity involving fewer ancilla, making qudits promising candidates for quantum computation. However, these improvements can only be practical if the gate duration overhead does not scale poorly with the Hilbert space dimension while still maintaining high gate fidelity. In this section we investigate this relation by explicitly constructing short duration pulses of at least 99.9% fidelity for one- and two-qudit gates using Incremental Pulse Re-seeding (IPR).

A. Single-qudit gates

We consider the single-qudit gates X_d, X_d^s, H_d and T_d presented in (7) for dimensions $d = 2, ..., 8$. In each case we include two guard levels to account for and suppress leakage into higher energy states. Expecting short gate durations, we choose $T_{\text{start}} = 50\,\text{ns}$ for the qubit and qutrit ($d = 3$) cases, and use the resulting shortest durations to estimate T_{start} for higher d with a linear extrapolation. We set `step` to the power of two nearest to $0.1\,T_{\text{start}}$ because this results in integer durations. Having found durations T_{best} for every dimension, we then run IPR repeatedly with starting durations randomly chosen from $[0.8\,T_{\text{best}}, 1.2\,T_{\text{best}}]$ to identify other local minima and reduce the gate time further, as well as to measure the uncertainty of the method.

Figure 7 shows the minimum single-qudit gate durations found over ten IPR runs for each dimension. The durations for the generalized gates X_d, X_d^s and the Hadamard gate, H_d, scale similarly with the qudit dimension. We suspect that the generalized T_d gate can be realized in lower time at higher dimension compared to the others because it only causes a change in phase, not in state populations. For this gate, we observe that the best pulse found for $d = 7$ is of shorter duration than the $d = 6$ pulse, when the reverse is intuitively expected. This discrepancy suggests shorter pulses may exist and indicates a highly complicated optimization space. Theoretical studies [12] have shown that optimal gate durations scale asymptotically with $\mathcal{O}(h^2)$, where h is the dimension of the Hilbert space. We therefore fit each sequence of data points with a quadratic polynomial $T_{\text{quad}}(d) = ad^2 + bd + c$ (solid grey lines) in $d = h$ to compare our findings with the theory. Furthermore, the layout of the data also motivates linear regression, so we investigate the performance of a linear model $T_{\text{lin}}(d) = bd + c$ (dotted grey lines) as well. The fit coefficients and their standard deviation estimates are given in Table III, where for each gate the first line corresponds to the linear fit and the second line to the quadratic fit. The quality of each regression is quantified with the coefficient of determination R^2.

The near-ideal R^2 values for quadratic fits to X_d, X_d^s and H_d indicate that the quadratic models describe the empirical scaling behaviors for these gates very well, which agrees with the theoretical predictions. However, we emphasize that linear scaling models capture the essential behavior in this low-dimensional regime (compared to asymptotic considerations) in good approximation too, deviating from the duration points by less than $5\,\text{ns}$ on average for those three gates. Barely any difference between quadratic and linear regression is noticeable for the T_d gate, but the more scattered data points lead to a reduced R^2 value and larger uncertainty. Additionally, the fit parameter $a < 0$ indicates an unrealistic scaling of pulse duration because this leads to negative durations in the limit of large d. T_d is a good case study in the interplay between theoretical expectations and empirical realizations. Specifically, the poor fit obtained here indicates that for many values of d, these gate durations may overestimate the optimal gate times.

TABLE III
Linear and quadratic fit coefficients for single-qudit gate durations

Gate	a [ns]	b [ns]	c [ns]	R^2
X_d	1.48 ± 0.12	26.79 ± 1.16	-26.07 ± 6.23	0.991
		12.02 ± 1.18	4.93 ± 2.67	1.000
X_d^s	1.60 ± 0.16	28.64 ± 1.26	-29.79 ± 6.78	0.990
		12.69 ± 1.60	3.71 ± 3.63	1.000
H_d	1.15 ± 0.36	26.04 ± 1.05	-24.61 ± 5.68	0.992
		14.49 ± 3.66	-0.36 ± 8.28	0.998
T_d	-0.38 ± 0.86	14.36 ± 1.37	14.36 ± 7.36	0.957
		18.17 ± 8.74	6.36 ± 19.78	0.959

This result of essentially linear scaling over the practical range of qudit dimensions has promising implications for the current era where qudit experiments have been conducted successfully [8]–[10]. It suggests that the computational advantage single-qudit gates can provide is not outweighed by pulse time overhead.

B. Two-qudit SWAP gate

For the two-qudit case we consider the generalized SWAP gate, which fully swaps the state amplitudes between the qudits. In many architectures, this is an important operation for communication, as limited connectivity between devices requires quantum information to be moved around repeatedly. We apply IPR to find SWAP durations for two qudits of dimensions 2, 3, and 4, with corresponding Hilbert space dimensions 4, 9, and 16, respectively. We limit our analysis to these three cases because simulating the evolution of two qudits with $d \geq 5$ (plus guard states) becomes very resource-expensive. We use two guard states for dimensions 2 and 3 and only one guard state for 4 due to the large Hilbert space. As in the single-qudit case, each data point is the minimum duration from ten IPR runs with different T_{start} values.

Figure 8 shows shortest-duration results for the SWAP gate with respect to the Hilbert space dimension $h = d^2$. Given the small number of points, we do not attempt a fit, but note that the points appear nearly linear, as was the case with the single-qudit gates. It is worth noting that fully swapping the states of two ququarts ($d = 4$) is slightly less expensive ($1128\,\text{ns}$) than performing two sequential SWAP operations on qubits

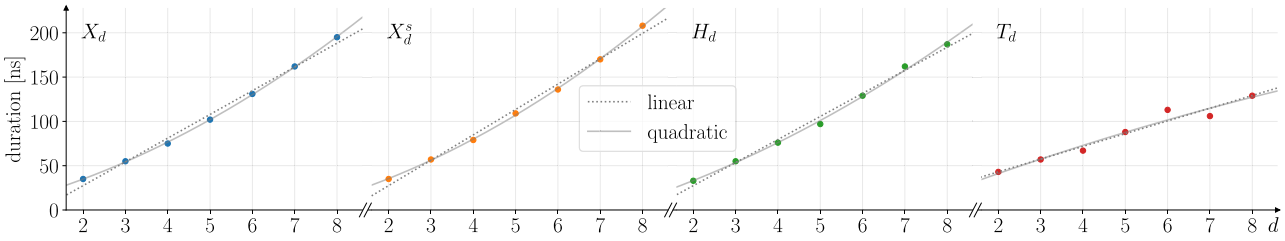

Fig. 7. Dependence of single-qudit gate durations from qudit dimension d. The four gates are defined in (7). Each data point shows the minimum duration of applying Incremental Pulse Re-seeding (IPR) from ten different starting times. The unexpected discrepancy between $d = 6$ and $d = 7$ for the generalized T_d gate suggests a highly complicated optimization space and shows that shorter pulses for some d values may be possible. For each gate the scaling of duration with dimension d is analyzed with quadratic and linear regressions, where the latter is less accurate but captures the essential behavior in this low-dimensional regime. The fit coefficients are presented in Table III.

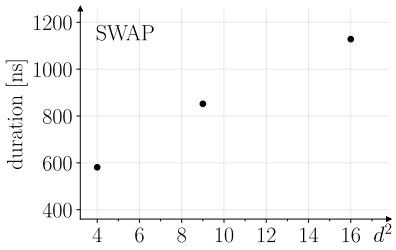

Fig. 8. Gate durations of two-qudit SWAP operations for qudit dimensions $d = 2, 3, 4$. The horizontal axis corresponds to the dimension of the full Hilbert space, $h = d^2$. Due to the low number of data points, regression is not applied, but a near-linear trend is visible.

($581\,\text{ns} + 581\,\text{ns} = 1162\,\text{ns}$). Given that a single ququart has the same information capacity as two qubits, this represents an efficient way of moving quantum information. Additionally, given that the $d = 4$ SWAP optimization space is much more complex than the qubit case, we expect that this duration could be further decreased through more sampling or fine-tuning.

VI. Discussion

We have presented the Incremental Pulse Re-seeding (IPR) scheme, an efficient method for finding short-duration pulses for quantum optimal control tasks, which performs better than random-guess based methods for large Hilbert space gates. We use this method together with the optimal control software Juqbox to demonstrate that single-qudit gate duration scaling is nearly linear for several useful gates, with dimension up to $d = 8$. We also show that the two-qudit SWAP gate appears to follow a similar trend for Hilbert space dimensions up to $d^2 = 16$.

We emphasize that while Incremental Pulse Re-seeding helps to make intelligent initial pulse guesses for the optimizer during intermediate steps, it does not eliminate the possibility of converging to a local infidelity minimum. As quantum optimal control problems are typically underconstrained, different initial guesses can converge to different solutions. In our case, this means starting IPR with different values for T_{start} and $\vec{\alpha}_{\text{guess}}$ may result in slightly different gate durations. However, as an example, our simulations show a relatively small variation in gate duration of $4.69\,\text{ns}$ on average for single-qudit gates, underlining the advantage of our algorithm.

Our results prompt several directions of future research:

- The efficiency of qudit gates motivates further exploration into quantum circuits that explicitly use qudits for computation, such as the asymptotic ancilla reductions achieved in [6] and [7]. The computational advantages of qudit circuits may allow NISQ devices to solve previously intractable problems.
- In this work, we consider the generalized SWAP two-qudit gate, which appears to scale nearly linearly for small qudit dimension. This result encourages analysis of duration scaling for more two-qudit gates, such as the SUM gate (a generalization of CNOT) from [15].
- The presented results are based on a system Hamiltonian modeling superconducting transmons. We emphasize that our approach is not limited to this specific case and could be directly applied to optimal control problems across a wider class of quantum systems.
- Our optimizations use an approximate Hamiltonian in a closed system (governed by the Schrodinger equation). For a more accurate measure of gate duration scaling, similar experiments are needed in an open system, where errors can be modeled more realistically. This can be achieved by describing the dynamics with the GKSL master equation [26], [27]. In addition, benchmarking the performance of qudit pulses on actual noisy quantum architectures would help to investigate the accuracy of models employed. Optimizing control pulses to specifically mitigate the increased sources of errors from operating on higher energy levels may lead to making qudit computation practical.

Overall, our work highlights the potential of qudit-based computation in the future and provides an effective method for finding short-duration qudit pulses. We find high-fidelity pulses of low duration for both one- and two-qudit gates with nearly linear scaling in the hardware-practical regime, suggesting that qudit computation can offer significantly increased efficiency compared to qubit-only circuits.

ACKNOWLEDGEMENTS

We would like to thank Stefanie Günther and N. Anders Petersson for valuable advice on using the quantum optimal control software packages Juqbox and Quandary. Additionally, we would like to thank David I. Schuster for helpful discussions regarding quantum optimal control theory.

REFERENCES

[1] Z. Gedik, I. A. Silva, B. Çakmak, G. Karpat, E. L. G. Vidoto, D. O. Soares-Pinto, E. R. deAzevedo, and F. F. Fanchini. Computational speed-up with a single qudit. *Scientific Reports*, 5(1):14671, October 2015. Number: 1 Publisher: Nature Publishing Group.

[2] Yannick Deller, Sebastian Schmitt, Maciej Lewenstein, Steve Lenk, Marika Federer, Fred Jendrzejewski, Philipp Hauke, and Valentin Kasper. Quantum approximate optimization algorithm for qudit systems with long-range interactions. *arXiv:2204.00340 [cond-mat, physics:quant-ph]*, April 2022. arXiv: 2204.00340.

[3] Duc Manh Nguyen and Sunghwan Kim. Quantum Key Distribution Protocol Based on Modified Generalization of Deutsch-Jozsa Algorithm in d-level Quantum System. *International Journal of Theoretical Physics*, 58(1):71–82, January 2019.

[4] Koji Nagata, Han Geurdes, Santanu Kumar Patro, Shahrokh Heidari, Ahmed Farouk, and Tadao Nakamura. Generalization of the Bernstein–Vazirani algorithm beyond qubit systems. *Quantum Studies: Mathematics and Foundations*, 7(1):17–21, March 2020.

[5] Alex Bocharov, Martin Roetteler, and Krysta M. Svore. Factoring with qutrits: Shor's algorithm on ternary and metaplectic quantum architectures. *Physical Review A*, 96(1):012306, July 2017. Publisher: American Physical Society.

[6] Pranav Gokhale, Jonathan M. Baker, Casey Duckering, Natalie C. Brown, Kenneth R. Brown, and Frederic T. Chong. Asymptotic Improvements to Quantum Circuits via Qutrits. *Proceedings of the 46th International Symposium on Computer Architecture*, pages 554–566, June 2019. arXiv: 1905.10481.

[7] Jonathan M. Baker, Casey Duckering, and Frederic T. Chong. Efficient Quantum Circuit Decompositions via Intermediate Qudits. In *2020 IEEE 50th International Symposium on Multiple-Valued Logic (ISMVL)*, pages 303–308, November 2020. ISSN: 2378-2226.

[8] Alexey Galda, Michael Cubeddu, Naoki Kanazawa, Prineha Narang, and Nathan Earnest-Noble. Implementing a Ternary Decomposition of the Toffoli Gate on Fixed-FrequencyTransmon Qutrits. *arXiv:2109.00558 [quant-ph]*, September 2021. arXiv: 2109.00558.

[9] Rigetti Computing. Beyond Qubits: Unlocking the Third State in Quantum Processors. https://medium.com/rigetti/beyond-qubits-unlocking-the-third-state-in-quantum-processors-12d2f84133c4, December 2021. Accessed 29 April 2022.

[10] Martin Ringbauer, Michael Meth, Lukas Postler, Roman Stricker, Rainer Blatt, Philipp Schindler, and Thomas Monz. A universal qudit quantum processor with trapped ions. *arXiv:2109.06903 [quant-ph]*, September 2021. arXiv: 2109.06903.

[11] John Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79, August 2018. arXiv: 1801.00862.

[12] Seth Lloyd and Reevu Maity. Efficient implementation of unitary transformations. *arXiv:1901.03431 [quant-ph]*, January 2019. arXiv: 1901.03431.

[13] Navin Khaneja, Timo Reiss, Cindie Kehlet, Thomas Schulte-Herbrüggen, and Steffen J. Glaser. Optimal control of coupled spin dynamics: design of NMR pulse sequences by gradient ascent algorithms. *Journal of Magnetic Resonance*, 172(2):296–305, February 2005.

[14] Michael A Nielsen and Isaac L Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[15] Daniel Gottesman. Fault-Tolerant Quantum Computation with Higher-Dimensional Systems. In Colin P. Williams, editor, *Quantum Computing and Quantum Communications*, Lecture Notes in Computer Science, pages 302–313, Berlin, Heidelberg, 1999. Springer.

[16] Lia Yeh and John van de Wetering. Constructing All Qutrit Controlled Clifford+T gates in Clifford+T. In Claudio Antares Mezzina and Krzysztof Podlaski, editors, *Reversible Computation*, Lecture Notes in Computer Science, pages 28–50, Cham, 2022. Springer International Publishing.

[17] N. Anders Petersson and Fortino Garcia. Optimal Control of Closed Quantum Systems via B-Splines with Carrier Waves. *arXiv:2106.14310 [quant-ph]*, June 2021. arXiv: 2106.14310.

[18] Stefanie Günther, N. Anders Petersson, and Jonathan L. DuBois. Quantum Optimal Control for Pure-State Preparation Using One Initial State. *arXiv:2106.09148 [quant-ph]*, August 2021. arXiv: 2106.09148.

[19] Jens Koch, Terri M. Yu, Jay Gambetta, A. A. Houck, D. I. Schuster, J. Majer, Alexandre Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf. Charge-insensitive qubit design derived from the Cooper pair box. *Phys. Rev. A*, 76(4):042319, October 2007. Publisher: American Physical Society.

[20] Sarah Sheldon, Easwar Magesan, Jerry M. Chow, and Jay M. Gambetta. Procedure for systematically tuning up cross-talk in the cross-resonance gate. *Physical Review A*, 93(6):060302, June 2016.

[21] N. Anders Petersson, Fortino M. Garcia, Austin E. Copeland, Ylva L. Rydin, and Jonathan L. DuBois. Discrete Adjoints for Accurate Numerical Optimization with Application to Quantum Control. *arXiv:2001.01013 [quant-ph]*, November 2020. arXiv: 2001.01013.

[22] Yunong Shi, Nelson Leung, Pranav Gokhale, Zane Rossi, David I. Schuster, Henry Hoffmann, and Frederic T. Chong. Optimized Compilation of Aggregated Instructions for Realistic Quantum Computers. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 1031–1044, Providence RI USA, April 2019. ACM.

[23] Chad Rigetti and Michel Devoret. Fully microwave-tunable universal gates in superconducting qubits with linear couplings and fixed transition frequencies. *Physical Review B*, 81(13):134507, April 2010.

[24] Matthew Ware, Blake R. Johnson, Jay M. Gambetta, Thomas A. Ohki, Jerry M. Chow, and B. L. T. Plourde. Cross-resonance interactions between superconducting qubits with variable detuning. *arXiv:1905.11480 [quant-ph]*, May 2019. arXiv: 1905.11480.

[25] Easwar Magesan and Jay M. Gambetta. Effective Hamiltonian models of the cross-resonance gate. *Phys. Rev. A*, 101(5):052308, May 2020. Publisher: American Physical Society.

[26] G. Lindblad. On the generators of quantum dynamical semigroups. *Communications in Mathematical Physics*, 48(2):119–130, June 1976.

[27] Vittorio Gorini. Completely positive dynamical semigroups of N-level systems. *Journal of Mathematical Physics*, 17(5):821, 1976.