Pulse electron-electron double resonance (PELDOR) analysis on a quantum computer

S. I. Protasov
Innopolis University
R. B. Zaripov
Zavoisky Physical-Technical Institute, Russian Academy of Sciences
I. T. Khairutdinov (semak-olic@mail.ru)
Zavoisky Physical-Technical Institute, Russian Academy of Sciences
K. M. Salikhov
Zavoisky Physical-Technical Institute, Russian Academy of Sciences

Research Article

Keywords: quantum computing, quantum Fourier transform, discrete Fourier transform, PELDOR, spin labels, nanometrology

Posted Date: June 29th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1735180/v1

Pulse electron-electron double resonance (PELDOR) analysis on a quantum computer

S.I. Protasov[1], R. B. Zaripov[2], I. T. Khairutdinov[2], K. M. Salikhov[2]

Abstract Quantum processing units (QPU) in theory propose a computational supremacy in a significant number of tasks. Quantum programs are well suited for vector and matrix data processing. The greatest concern is whether physical implementations will step over the noise and decoherence limitations: today in the NISQ (noisy intermediate-scale quantum) era the bigger the problem, the less reliable the results are. In this work we show that even NISQ computers can be used to obtain reliable results in processing experimental data. We perform quantum Fourier transform (QFT) of PELDOR oscillation at IonQ trapped ion QPU followed by spin labels distance measurement by analysis of oscillation data. For 4 and 5-qubit programs we show the results comparable to Fourier analysis on a classical computer.

Key words: quantum computing, quantum Fourier transform, discrete Fourier transform, PELDOR, spin labels, nanometrology

I. T. Khairutdinov
e-mail: semak-olic@mail.ru
[1]Innopolis University, Innopolis, Russian Federation

[2]Zavoisky Physical-Technical Institute, FRC Kazan Scientific Center, Russian Academy of Sciences, Kazan, Russian Federation

1 Introduction

In pulse magnetic resonance spectroscopy, the detected signals are recorded in the time domain. In most cases those signals demonstrate modulation phenomena [1,2]. The frequencies of modulation contain information about spin-spin interactions. For example, pulse electron double resonance (PELDOR) reveals the spin echo envelope modulation induced by the dipole-dipole interaction between two paramagnetic particles [2–9]. The frequency of that modulation is determined by a distance between electron spins in pairs of paramagnetic particles.

To obtain these frequencies of the modulated signals, the Fourier transformation of the original time domain signal is used. In the experiment, we get a discrete data set. Therefore, we use the discrete Fourier transform (DFT) method. This method is well developed and always used.

In this work we have performed the DFT of PELDOR data using a common classic computer. We have done that transform using an available quantum computer as well. We have compared results obtained by these two computers. Of course, we are aware that, as it were, we gave a start to run the distance to the current world champion and a child just starting to run. But we believe in the wonderful future of this child and it is interesting for us to observe his first steps. We think this will help us better understand the capabilities of a quantum computer and how to best use it today and in future.

Interest in quantum computing and quantum processing is steadily growing. By now, some companies are already proposing to use physical quantum bits (qubits) to solve a certain range of problems [10,11]. In this paper, we will not go into the architecture of a quantum computing module. However, we would like to draw the attention of scientific groups to try data processing using quantum computers. This will identify problems with certain physical implementations and will give incentives to the designers of quantum computers to improve their systems.

In this paper, we want to show the possibilities of Fourier analysis of PELDOR data using a quantum computer.

Note that there are interesting steps toward an application of quantum computers for solving problems of magnetic resonance (see, e.g., [12,13]).

1.1 More about measuring the distances between paramagnetic centers in solids by PELDOR nanometrology

The measurement of distances between paramagnetic centers in 1-10 nanometers scale using PELDOR method is of great importance (see [14]). For example, knowing the distance between two spin labels embedded into a DNA molecule makes it possible to more accurately determine DNA structure using “EPR crystallography”. PELDOR nanometrology is also very

important from the point of view of obtaining a given architecture of the spatial arrangement of paramagnetic centers when using unpaired electrons of paramagnetic centers as the element base of quantum computers and in quantum informatics.

PELDOR experiments are widely used for determining distances between spin labels in pairs, and also the number of dipolar-coupled spins in groups, e.g., in spin-labeled proteins, and the distribution of distances between spin labels in groups [1,2,4,9,14–20]. There are two commonly used experimental protocols: three-pulse [1] and four-pulse ELDOR [21]. The first theoretical description of the three-pulse ELDOR experiment was presented in [1]. In this work, we used the experimental data of the four-pulse ELDOR. This method consists of a refocused primary echo sub-sequence of pulses on A spins. The pump pulse on B spins implemented between the second and the third pulses on A spins [5,21].

The dipole–dipole interaction leads to the modulation of the echo signals amplitude. The modulation frequency is determined by the value of the dipole–dipole interaction between spin labels (see, e.g., [9]).

In solids paramagnetic centers have typically broad distribution of the resonance frequencies so that EPR spectra have so called inhomogeneous broadening. Under this condition it is rather safe to consider contribution of the dipole-dipole Hamiltonian in the secular approximation

$$H_{d-d,sec} = \hbar D_{AB} S_{Az} S_{Bz}, \quad (1)$$

where $D_{AB} = D_0(1 - 3cos^2\theta)$, $D_0 = \frac{g_A g_B \beta^2}{\hbar r^3}$. Here r is the distance between A and B spins, θ is the angle between the vector $\boldsymbol{r}$ and the direction of the external magnetic field, β is the Bohr magneton, ħ is the Planck constant, $g_{A,B}$ are the spectroscopic g-factors of A and B spins.

Each orientation of the pair is characterized by its dipolar frequency $D_0(1 - 3cos^2\theta)$. When the polar angle θ varies from 0 to π, the frequency $D_0(1 - 3cos^2\theta)$ changes in the interval $\{-2D_0, D_0\}$, so that the absolute value varies in the interval $\{0, 2D_0\}$. The frequency D_0 occurs when $\theta = \frac{\pi}{2}$, i.e. on the equator of the spherical coordinate system. The statistical weight of equatorial points is the largest one.

Dipolar frequency depends on a distance r between spins. For example, the value $D_0 = 6.5$ MHz corresponds to the distance $r = 2$ nm in the pair.

Time dependence of the observed PELDOR signal is influenced by a number of factors:

- Spin dynamics induced by the spin-spin interaction.
- Paramagnetic relaxation of spins. The relaxation leads to a decay of the observed PELDOR signal with characteristic relaxation time.

- Spin-lattice relaxation produces a red shift of the frequency of modulation induced by dipole-dipole interaction [22].
- PELDOR signal depends also on an excitation pattern of spins by the microwave pulses forming the signals. Note, that the dipole-dipole interaction leads to the modulation of the PELDOR signal is only under the condition that MW pulses excite both spins in a pair [9].

Suppose that there is an ensemble of spin pairs with fixed distance *r* between spins. In glasses due to random orientation of pairs of paramagnetic centers in a space, there is distribution of the modulation frequencies depending on an angle between the orientation of radius-vector ***r*** and the orientation of the external magnetic field $\mathbf{B_0}$. Time dependence of four-pulse ELDOR for pairs with a fixed distance *r* can be written as follows [18]:

$$V(T) = 1 - p + p\ cos(D_{AB}(T - \tau)),$$

where *p* is the probability of B spin being flipped by the pump pulse, *T* is the time of application of pulses on B spins, τ is the time interval between first and second pulses.

As it was mentioned above the modulation frequency D_{AB} has a broad distribution. And this distribution at the frequency $\Omega = D_0$ has a singularity. Near the singularity, the distribution function behaves like $f(\Omega) = \frac{1}{2\sqrt{3D_0}\sqrt{D_0 - \Omega}}$ (see, e.g., [9]). Note that the contribution of majority of pairs of paramagnetic particles to the PELDOR signal decreases to zero over a time of the order of $1/D_0$ due to the destructive interference of terms with different frequencies distributed in the interval $\{0, D_0\}$. At the same time, the contributions to the PELDOR signal from those pairs of spins for which the modulation frequency of the observed signal lies near the singular point of the frequency distribution exhibit constructive interference and give the experimentally observed modulation of the signal with the frequency D_0.

In real systems there might be also a distribution of distance *r* values between two electron spins. As a result, distribution of modulation frequencies will change: there might be two or more maxima which correspond to different sub-ensemble of pairs with different *r* and $D_0(r)$ values.

1.2 About discrete Fourier transform with quantum computers

The algorithm for determining modulation frequencies from PELDOR signal dependence on an observation time intervals τ is well developed [1,14,17,18]. It includes the Fourier transformation of the PELDOR signal. In our work, the experimental dependences of the four-pulse ELDOR are presented.

One can compare the quantum computers niche with a massive parallel processor, which can apply the same instructions to multiple data. Such processors (i.e. graphical cards, tensor

processing units) show supremacy in machine learning, computer graphics, computer vision, and matrix and tensor operations in general. Discrete Fourier transform is a very good example of such an operation: it can be treated as multiplication of the data vector by the specific matrix. In quantum implementation the algorithm itself will show exponential speedup, thus we will benefit, first of all, in the very large-scale problems — e.g., 100-qubit computers will be able to process 2^{100} data points.

In this paper we concentrate on the available quantum computers potential in finding frequency distribution manifesting as modulation of PELDOR signal for several systems. The same PELDOR experiments data were Fourier transformed using the well-established fast Fourier transform (FFT) algorithm with a classic computer and with quantum Fourier transform (QFT) on a quantum computer. We compare results obtained by these two approaches.

1.2.1 Terms and Related works

From the experimental perspective, let us note the major terms of quantum computing, relevant to the paper results. General purpose quantum computers are physical systems [23], which implement the same mathematical model of computations. *Physical* element of a *quantum processing unit* (QPU) is a *qubit* — quantum object with 2 dedicated states in the chosen measurement basis. This can be a photon polarization, an electron spin, or current in a superconducting circuit. Quantum properties of such objects allow us to use them as a 2-level system.

In physics we manipulate these qubits and change their states with laser and magnetic field pulses. In this way one manipulates the whole system's *quantum state vectors*. Usually these vectors are written in Dirac's ket notation. Let us define dedicated qubit states as 0 and 1. Then state $|010\rangle = |0\rangle \otimes |1\rangle \otimes |0\rangle$ means, that system has 3 qubits, the second is in state 1, e.g. for a spin-based system it has the spin down. As the qubit system state is always a binary number, notation is often shortened to a decimal representation: $|010\rangle = |2\rangle$ and $|111\rangle = |7\rangle$. A system of *n* qubits can be theoretically prepared into arbitrary superpositions of such states; thus the whole system can be described with a single 2^n-dimensional vector. For example, in a 3-qubit register the $\frac{1}{\sqrt{2}}|010\rangle + i|101\rangle$ state in Dirac's notation is equivalent to the $(0, 0, \frac{1}{\sqrt{2}}, 0, 0, \frac{i}{\sqrt{2}}, 0, 0)^T$ column-vector.

In the end of any quantum program the state is measured in the chosen basis, and each measurement result (binary string) is observed with a probability equal to a squared modulus of the state's amplitude.

Qubits and their pairs can be manipulated with laser and magnetic field pulses (as for spins, superconducting circuits) or lenses and filters (in photonics). Mathematical representations of these manipulations are referred to as *quantum gates*. Theoretical model of a quantum gate is a unitary matrix U, which is related to the Hamiltonian operator as $U = e^{-iHt}$, where t is a pulse duration. 2-qubit gates, known as *controlled NOT* (*CNOT*), together with universal single-qubit gates form the basis and can implement any unitary transformation, involving arbitrary number of qubits. CNOT implements the logical operator "if qubit *A=1* then qubit *B*: = *not B*".

Each quantum gate, applied to a single qubit, or a pair of qubits, affects all quantum states of a superposition. State superposition together with the mentioned effect enables *quantum parallelism*: all state amplitude coefficients change, even if we manipulate a single qubit. CNOT gates allow to prepare entangled qubit ensembles.

Existing quantum computers and programming languages share common conventions on gate names. Core operators are **X** (NOT), **Y**, **Z** which implement reflection with respect to the mentioned axis on a Bloch sphere, $\mathbf{R_X(\theta)}$, $\mathbf{R_Y(\theta)}$, $\mathbf{R_Z(\theta)}$ (or Phase gate) which perform corresponding rotations; Hadamard gate **H** which prepares qubit superposition; SWAP gate which exchanges qubit amplitudes; and all their controlled (or conditional) versions, which can be implemented with the help of CNOT. Quantum programming languages like Q#, qiskit, Cirq, provide all these mathematical operators in the form of programming functions. In our work we used IBM qiskit library [24].

Solution of our problem consists of two stages. First, we need to encode the data from a classical system into quantum state amplitudes. For that we use algorithm [25] implemented in qiskit, as it is asymptotically optimal and has implementation in qiskit library. In the methodology section we will show how to improve qiskit implementation to make it work on a real QPU. Second, we use the quantum implementation of Fourier transform [26]. This algorithm mathematically repeats the classic DFT method, thus state vector coefficients store Fourier transform frequency amplitudes. In the end we observe desired frequency component amplitudes at measurement. Probability to observe the frequency is proportional to the squared modulus of a desired component. So, we do multiple experiments to obtain probability distribution of frequencies.

2 Methodology

We perform our calculations on two platforms: classical computer, and 11-qubit fully-connected IonQ computer built with trapped ions [27]. Connectivity here refers to the ability of a QPU to perform CNOT gate to an arbitrary pair of qubits. Partial connectivity requires multiple additional CNOT gates to model desired operation.

Before we stopped our choice on the IonQ platform, we also considered 5-qubit IBM superconducting computers. We run a non-optimized version of our algorithm on 4 and 5 qubits on both QPUs. Our observation showed the IBM QPU generally introduced more noise than IonQ at this scale and could not solve the problem. This paper [28] supports our observation and shows that fully-connected architecture tends to perform better. Thus, for the sake of clarity, in this paper we assess a better platform (IonQ) with 4, 5, and 6-qubit circuits to observe the boundary of computation abilities.

To obtain a reasonable result on either platform, we want the assessed value to be well-interpretable. We propose to solve a task of *dominating frequency detection*, which requires a single numerical answer. We use this frequency value to compute the spin labels distance with equation (1).

Choice of data points. Our input is PELDOR data in a time domain. Using thousands of these points is beyond current QPU capabilities, thus we reduce their number with two techniques. First, as it was mentioned above, there is a broad distribution of non-characteristic dipolar frequencies, which give a fast decay of the experimental curves. This allows us to cut the original fast decay part of the experimental curve. Second, in our experiments we used 4, 5, and 6 qubits, which allow us to deal with 16, 32, and 64 data points. So, we sample the experimental data with the constant time step to keep the required number of PELDOR time measurements.

Performing analysis on a classical computer. We use fast discrete Fourier transform analysis conducted on the classical computer as the reference study. At the initial stage we analyzed dependence of Fourier image on the number of data points which is acceptable with respect to the result precision.

Encoding the data. One of the complicated parts of any quantum program is data transfer from a classical computer into the quantum states. We use the method proposed by authors of [25], which is ideologically similar to divide-and-conquer algorithms. The task of the algorithm is for a given complex vector $\{a_i\} \in C^{2^n}$ (data vector) and n qubits initialized in $|0\rangle$ states, construct such unitary matrix A:

$$\sum_{i=0}^{2^n-1} a_i \, |i\rangle = A|0\rangle^{\oplus n}.$$

A large number of quantum algorithms use amplitude data vector encoding of such form [26,29,30].

In our case vector $\{a_i\}$ stores PELDOR signal amplitudes in the sampled moments of time.

Quantum Fourier transform. From the mathematical point of view, the quantum algorithm of Fourier transform [26] exactly implements discrete Fourier transform matrix. Discrete Fourier transform is a procedure which converts *N* measurements of a signal a_t with constant sampling rate into a discrete frequency space f_k with *N* frequency components:

$$f_k = \sum_{t=0}^{N-1} a_t e^{-\frac{2i\pi kt}{N}}$$

This formula can be expressed in a matrix form for time-based vector *a* and frequency-based vector *f*: $f = FT \cdot a$, where $FT_{kt} = e^{-\frac{2i\pi kt}{N}}$, $N=2^n$.

Thus, for example, 4-dimensional vector will require 4×4 *FT* matrix:

$$FT = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{pmatrix}$$

In practice for simplicity of computation the fast Fourier transform algorithm requires input vector length to be a power of 2, and experimental data is padded with zeros.

All *FT* matrices are unitary matrices. Unitary operators can be efficiently modeled with a sequence of quantum gates, which means we can perform discrete Fourier transform computation in a quantum system. Input data vector $\{a_t\}$ is encoded in state vector amplitudes $|a\rangle$, discrete Fourier transform matrix is implemented with a series of quantum gates (for example, we apply magnetic field pulses for spins serving as qubits). Resulting state amplitudes represent vector ***f***. In our case components of this vector give distribution of frequencies. Note, that encoding of $|a\rangle$ and $|f\rangle$ states require $log_2 N$ qubits.

$|f\rangle = QFT\ |a\rangle$, where QFT - quantum Fourier transform operator.

There are few things we should keep in mind when applying this method. First, vectors $|a\rangle$ and $|f\rangle$ should be valid quantum states, thus vector $|a\rangle$ should be normalized before initialization. We cannot encode arbitrary vectors, but only with values that represent valid quantum states: the sum of their modulus should be equal to 1. To return back to the original vector, the normalization factor can be later re-applied in a classical computer. Second, result vector $|f\rangle$ should be normalized, thus $QFT = \frac{1}{\sqrt{N}} FT$. Third, state vectors in multi-qubit systems can only have power of 2 length. Thus if the number of data points is not a power of 2, we have to follow the same padding strategy as for the fast Fourier transform. In our work we avoid this problem by proper choice of data points number.

Detailed explanation of how to decompose a Fourier transform matrix into primitive quantum operations is given in the supplementary materials.

Overall algorithm scheme is presented in Fig. 1. Each horizontal line represents a qubit, time flows from the left to the right. Boxes and vertical connections represent 1 and 2-qubit operations, respectively. In this scheme one can see 3 types of operators: Hadamard **H** gates, controlled phase (CP, or sometimes **CRz** — controlled rotation about **Z**), and SWAP gates. Hadamard **H** operator is a 1-qubit gate responsible for preparation of a superposed state, 2-qubit CP-gate adds phase components to the amplitudes, 2-qubit SWAP-gate exchange qubits states.

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} \qquad CP(\theta) = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & e^{i\theta}\end{bmatrix} \quad SWAP = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \tag{2}$$

First, a single-qubit **H**-gate is applied to the qubit to prepare it in a superposed state. If a qubit is initially in any pure basis state, after application of **H**-gate it will appear on the equator of the Bloch sphere. We need this transformation to make use of phase gates.

Second, we change the phase of this superposed qubit with controlled phase CP gates. If the controlling qubit is set to $|1\rangle$, then the controlled qubit's phase is increased by a fraction of π. We repeat this for all qubits.

At the last step to obtain the result we reverse the order of the qubits with SWAP gates. Further in our implementation we will drop these gates and perform reordering on a classical computer.

The fact that the discrete Fourier transform operator is linear enables quantum parallelism. Running the given quantum program for the superposition of input states, results in a superposition of desired frequency components.

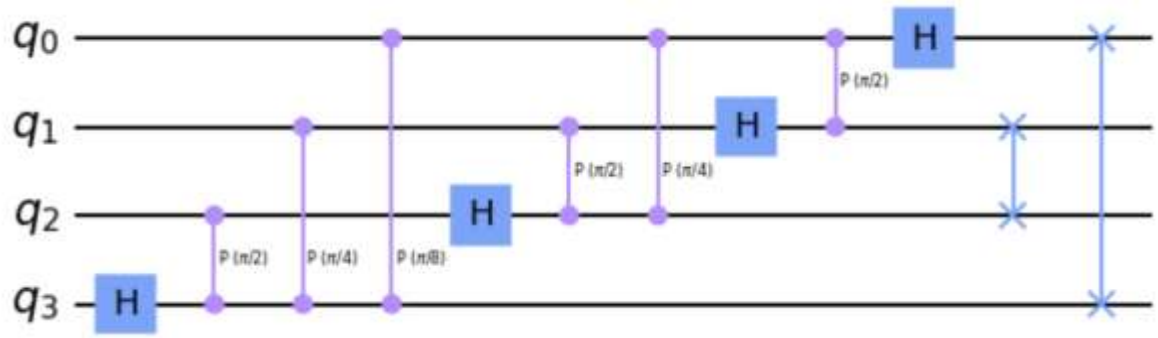


Fig. 1 This quantum circuit represents the sequence of operations (gates) we apply to single qubits and their pairs to implement Quantum Fourier transform. Squares correspond to Hadamard gate, small circle gates with *P*(angle) labels are controlled phase gates (also known as controlled R_Z rotations). Rightmost gates are SWAP gates, which restore qubit order

After application of QFT we measure the qubits. Probability of observing state $|k\rangle$ corresponds to the squared modulus of the k^{th} amplitude. Doing multiple experiments, we approximate a FT decomposition vector with square roots of the empirical probability distribution.

Circuit optimization. Major obstacles to creation of robust quantum programs are execution time and 2-qubit gates count. We want to make both these numbers as small as possible. By addressing the number of gates we implicitly reduce execution time. Here we use four optimization techniques to achieve acceptable quality of the result. It is very important to notice that these optimizations are fully algorithmic, and they are universal for any input data vector.

First, we accept the comment of IonQ best practices [31], that hardware implementations of $\mathbf{R_X}$ and $\mathbf{R_Y}$ rotation gates has a precision limit near $\frac{\pi}{10^3}$ radians. Thus, application of small angle rotation gates extends execution time without doing a useful job. Taking this into account, we remove small-angle single-qubit rotation gates from our circuit. We tried different threshold values starting with $\frac{\pi}{10^3}$, and we observed that even $\frac{5\pi}{10^2}$ rotations can be removed without significant impact on the result for our particular case. Thus, in our experiment all $\mathbf{R_X}$, $\mathbf{R_Y}$ and $\mathbf{R_Z}$ gates with the angle less than $\frac{5\pi}{10^2}$ radians are removed.

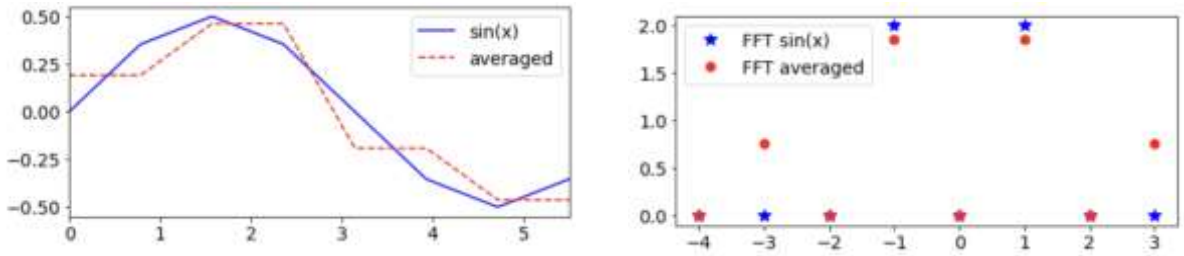
Second, we accept the idea from [31], that SWAP gates can be post-processed algorithmically. Major body of the QFT algorithm is composed in such a way that the resulting frequencies are encoded in the reversed order of qubits (e.g. 3 is $|1100\rangle$ instead of $|0011\rangle$). That is why in qiskit library it is appended with the series of $\frac{n}{2}$ SWAP gates. These gates are very expensive in terms of error: each SWAP gate is implemented with 3 consequent CNOT gates. In total we have $\frac{3n}{2}$ CNOT gates for an operation, which is much faster done on a classical computer. In our implementation we remove SWAP gates and do post-processing in the classical part of our program.

Third, according to [25], a pair of equal *CNOT*(*A,B*) gates cancel each other, even if they are separated with a commuting operator. This situation often appears in the initialization algorithm [25], if two consequent values in the vector are very close or equal. This allows us to remove a significant portion of CNOT gates.

Fourth, to get more profit from the last approach, we force the vector to have equal values in neighboring positions by replacing them with their average. This method adds distortions to the distribution function in a high-frequency region.

Let us provide an example of how data improvement can help with CNOT gates count at the initialization step. We will consider a period of a given definite function, e.g. *sin*(x), and its modified (according to the fourth suggestion described above) version as input data for the

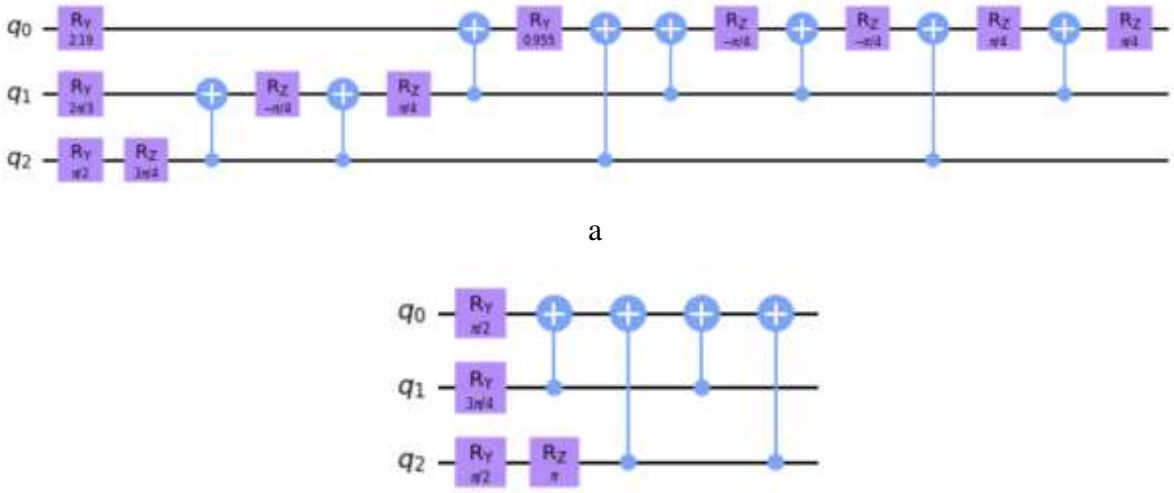
initialization algorithm. One can observe both functions and their Fourier images in Fig. 2. On the graphs one can see that averaging of consequent values lead to appearance of high-frequency aberrations, which is still smaller than the major frequency component.



a b

Fig. 2 Original signal of *sin(x)* (blue), modified data (red) **(a)**, and their Fourier images **(b)**

Now let us consider the circuits produced by the initialization method of qiskit for both vectors. In Fig. 3 one can make two observations. First, the circuit for the modified data is remarkably shorter. Second, the right part of this circuit consists of CNOT operations and has no rotation gates among them. This makes this part of the program suitable for the third optimization described above, and in practice all CNOT gates will be removed.



a b

Fig 3 (a) Quantum circuit to initialize the vector with *sin(x)* values. **(b)** Modified version of the quantum circuit to initialize that vector

Quality test procedure. We require our program to solve a practical task. Here we propose an instrument to check the quality of the proposed quantum program implementation. For quality estimation we propose the following technique. We will assess our system as a black box: vectors are input data, distance predictions are output. We run an optimized circuit 10 times, which is comparably cheap (we pay 0.7 US cent for gates), from which we choose the most probable frequency value (maximum of the frequency distribution function). Distribution is expected to be symmetric, thus we sum amplitudes of positive and negative equal frequencies. If the result is

equal to classic FFT prediction, we count estimation as correct. Obtained results of circuit optimizations and quality estimation are given in Table 1 (see below).

3 Implementation

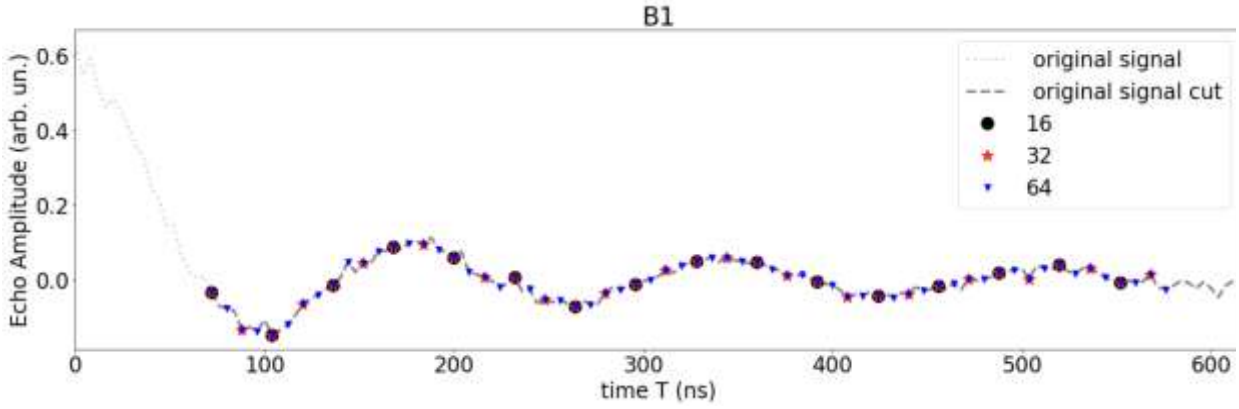
Our experiment design includes data preparation stage, experiment validation on an ideal simulator, and QPU experiments.

3.1 PELDOR experimental data

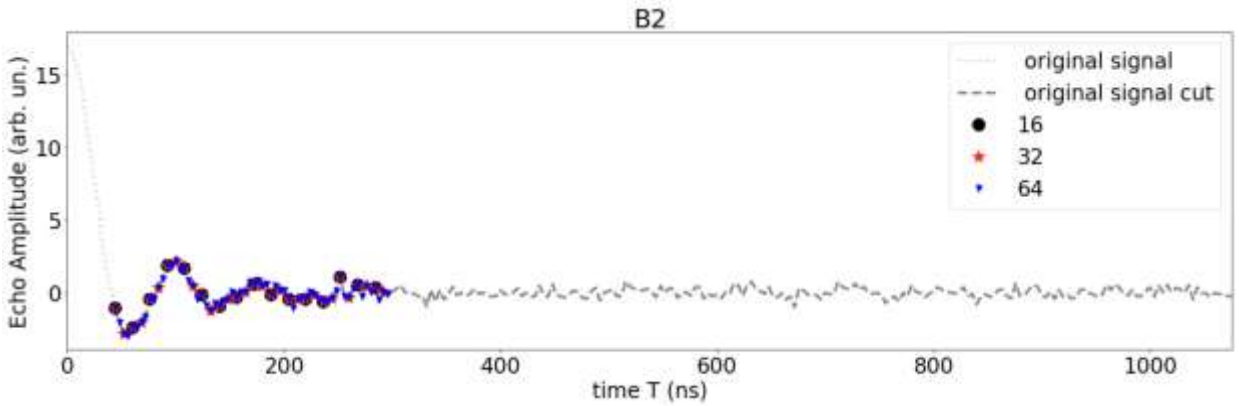
The EPR measurements were carried out on an Elexsys E580 FT- spectrometer (Bruker). Spectrometer operates in both X (9.8 GHz) and Q (34.2 GHz) frequency bands . The spectrometer is equipped with commercially available cavities ER4118MD5-W1 (X-band) and EN5107D2 (Q-band), which was placed in a cryostat CF935. The temperature was controlled using an ITC503 controller (Oxford). All data were obtained using 4-pulse ELDOR. The measurements were carried out at ambient temperature and 80K.

Three known nitroxide biradicals were used as samples: test biradical **B1** supplied by Bruker (r =2.2nm) [32], model biradical (4,4'-di(2,2,6,6-tetramethyl-piperidine-1-oxyl)diimino-2 ,2'-bipyridine) **B2** (r=1.9nm) provided by Prof. Yuri Galyametdinov [33] and a biradical **B3** provided by Prof. Gunnar Jeschke (r =3.6nm) [34,35]. Experimental data for biradicals **B1** and **B3** were obtained at 80K at Q band. Biradical **B2** was measured at X band at ambient temperature.

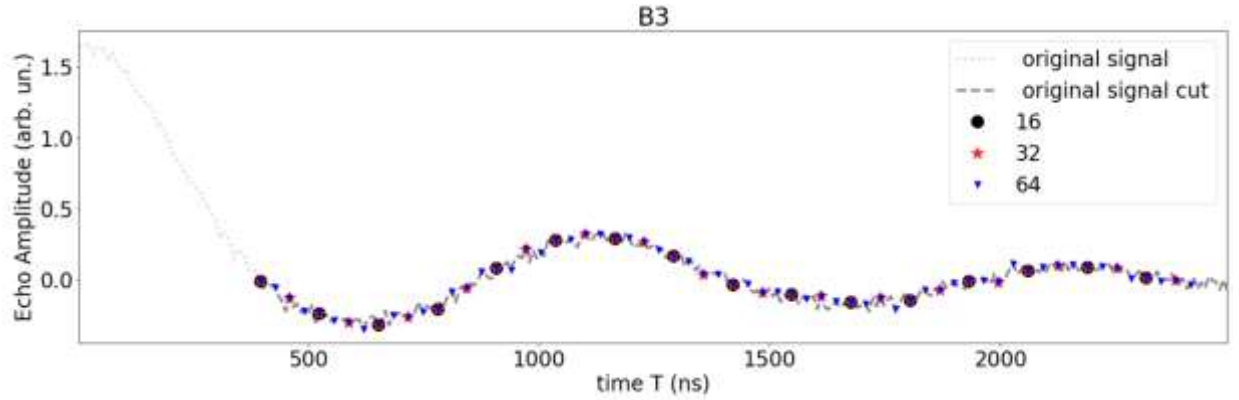
We run our experiments with 4, 5, and 6 qubits. Our intention is to show, at which scale applications of NISQ systems can resist physical errors. On the **data preparation** stage, we reduce the original experimental signal to 2^4, 2^5, and 2^6 points respectively. For data reduction we cut the first decay of the graph until intersection with 0. This helps to highlight the major oscillation frequency. Then we sample points with constant steps to catch a few oscillation periods. We choose every 2^{nd}, 4^{th}, or 8^{th} point depending on the experiment data. Fig. 4 presents experimental PELDOR dataset of biradicals **B1, B2** and **B3**. It compares the sampled data (markers) with the full EPR measurements (lines).



a



b



c

Fig. 4 Experimental PELDOR dataset of **B1 (a)**, **B2 (b)**, and **B3 (c)**. Dot lines present raw experimental data. Circles, stars, and triangles show variants of discretization of the experimental curve into 16, 32, and 64 equidistant time intervals, respectively

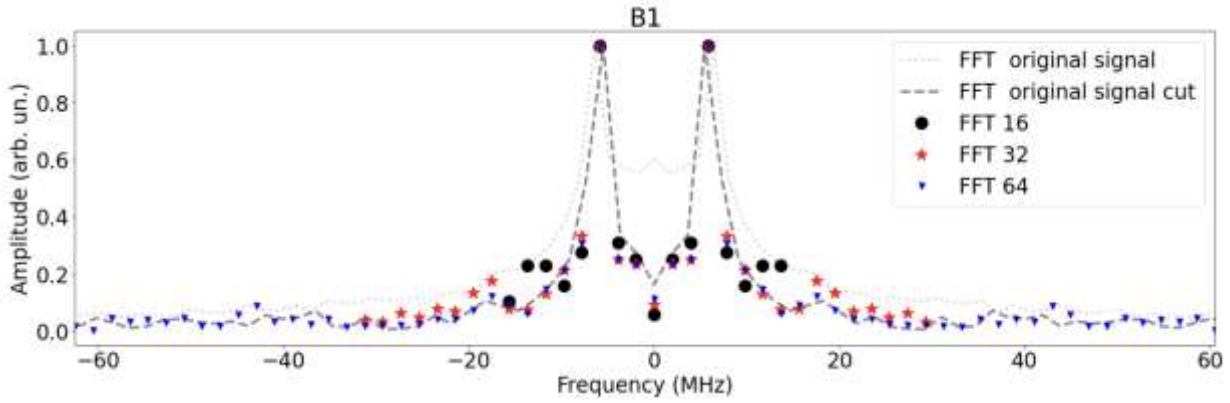
3.2 Validation of quantum Fourier transformation circuit

Second stage of our experiments is to validate that our quantum circuit is equivalent to the DFT method. We run our experiments on a noiseless IBM qiskit probabilistic simulator with 2000 repetition to obtain approximate distribution. This distribution is proportional to the square of quantum states amplitude modulus. Even with approximations mentioned in [31] we got a good agreement of results obtained with FFT and simulated QFT.

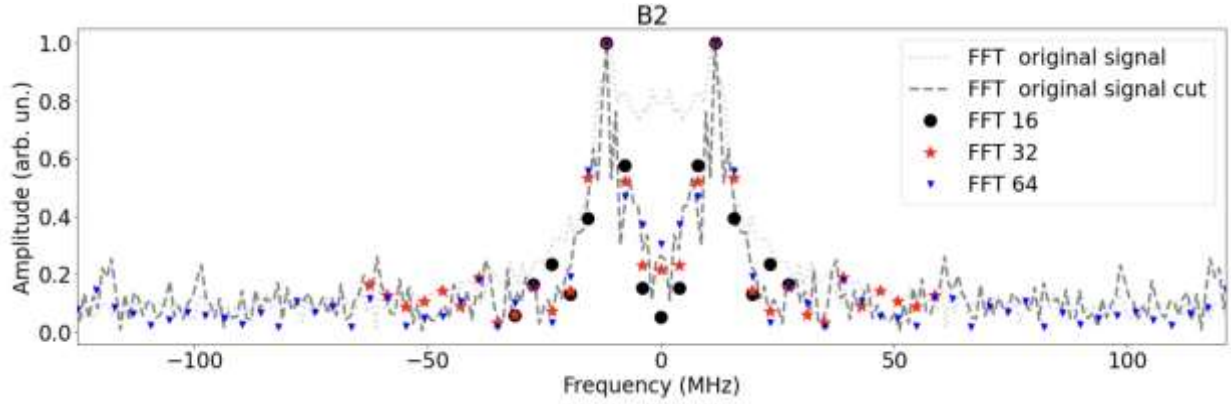
3.2.1 Sampling

In Fig. 5, FFT results for different biradicals and data ranges are presented. The dotted lines correspond to the Fourier transform of the original experimental curve. The dashed lines correspond to the Fourier transform of the experimental curve without the fast decay at the beginning. The black circles, red stars, and blue triangles show the Fourier transform for time splitting the original data into 16, 32, and 64 points, respectively. One can see that FFT of the original PELDOR datasets reveals a broad distribution, which is characterized by the fact that the

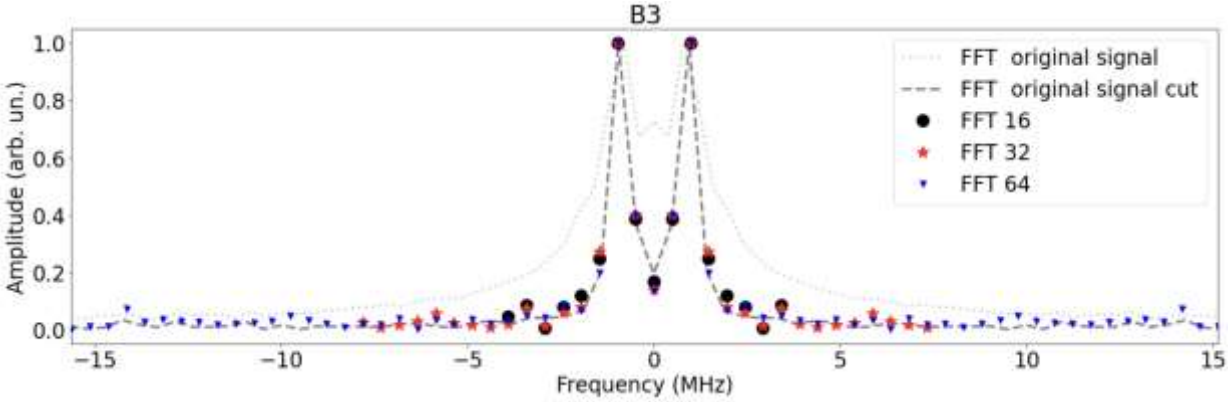
fast decay is determined by the presence of a large number of low frequency contributions. FFT of trimmed original signal and sampled with different intervals result in similar frequency distribution. Therefore, we will not show the data of the original signal further.



a



b

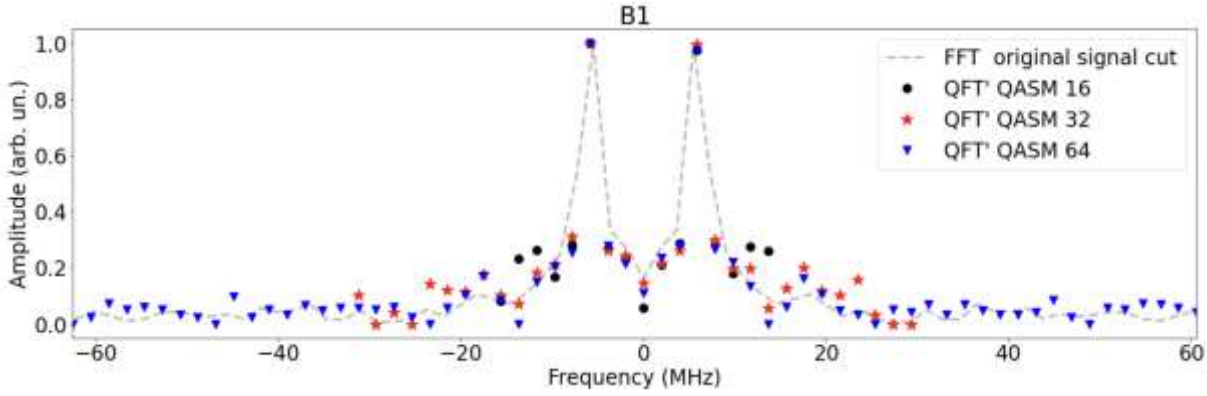


c

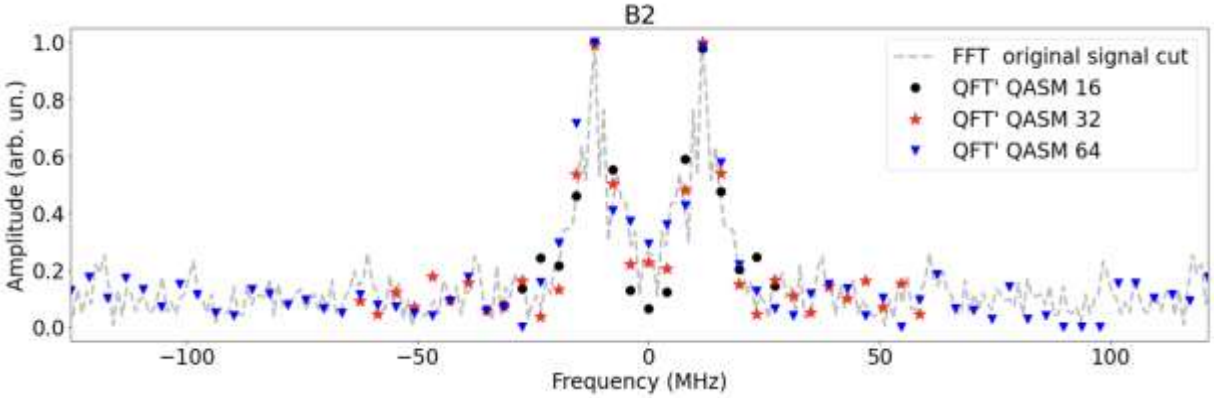
Fig. 5 FFT results for different data ranges of **B1 (a)**, **B2 (b)**, and **B3 (c)**. The dotted lines correspond to the Fourier transform of the original experimental curves. The dashed lines correspond to the Fourier transform of the experimental curve without the fast decay at the beginning. The black circles, red stars, and blue triangles show the Fourier transform for time splitting the original data into 16, 32, and 64 points, respectively

3.2.2 Quantum simulator

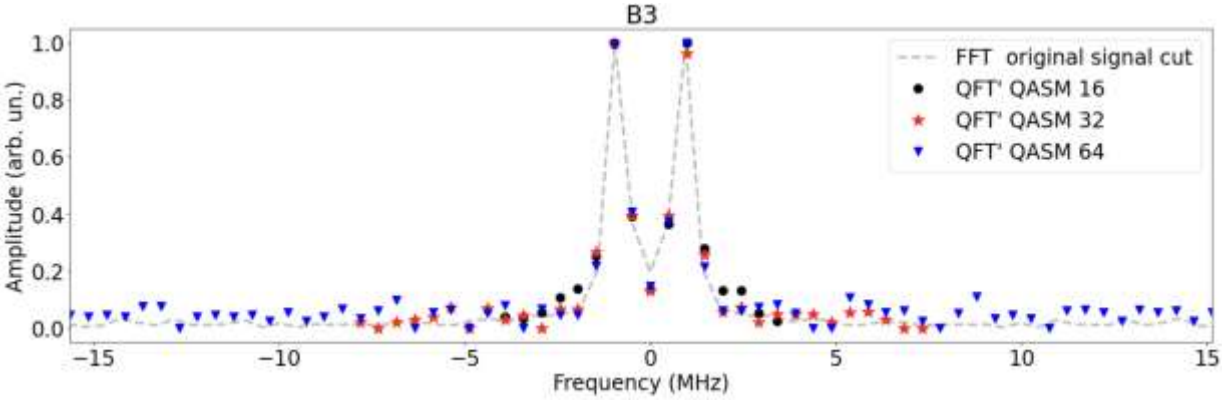
At the next step we decided to check whether the circuit represents Fourier transformation correctly. For this we have performed simulation using the circuit chosen (see Fig. 2).



a



b



c

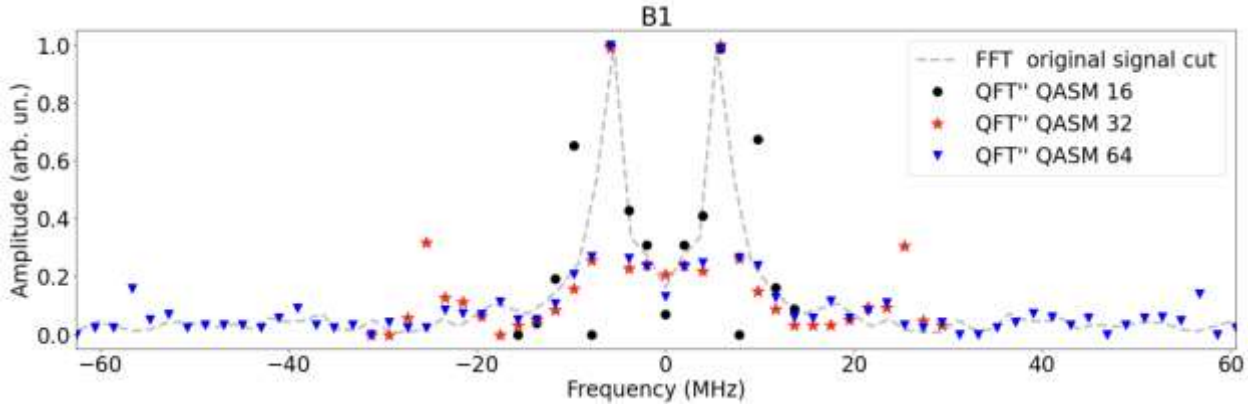
Fig 6 Simulation results of the optimized circuit for the **B1 (a)**, **B2 (b)**, and **B3 (c)**. The dashed lines correspond to the Fourier transform of the experimental curve without the fast decay region at the beginning. The black circles, red stars, and blue triangles show the QFT on the simulator for time splitting the original data into 16, 32, and 64 points, respectively

Results of QFT obtained with the simulator were compared with the data obtained by FFT (see Fig. 6). As we can see there is a good agreement between those FFT and QFT. This agreement confirms that we have composed the correct quantum circuit for QFT.

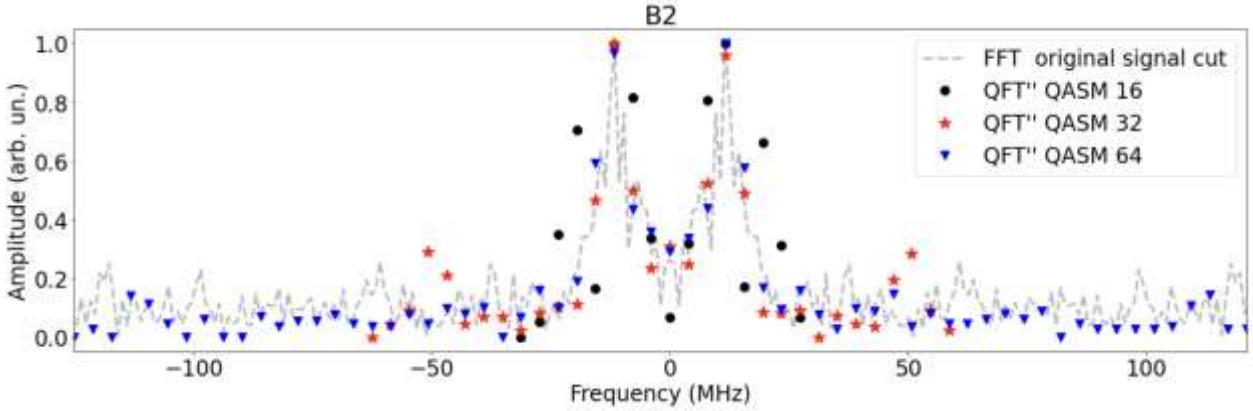
Thus, good coincidence of FFT vs QFT performed by quantum simulator confirms that circuit used is correct. And we can safely use this circuit with the QPU.

3.3 Circuit optimization for execution on the real QPU

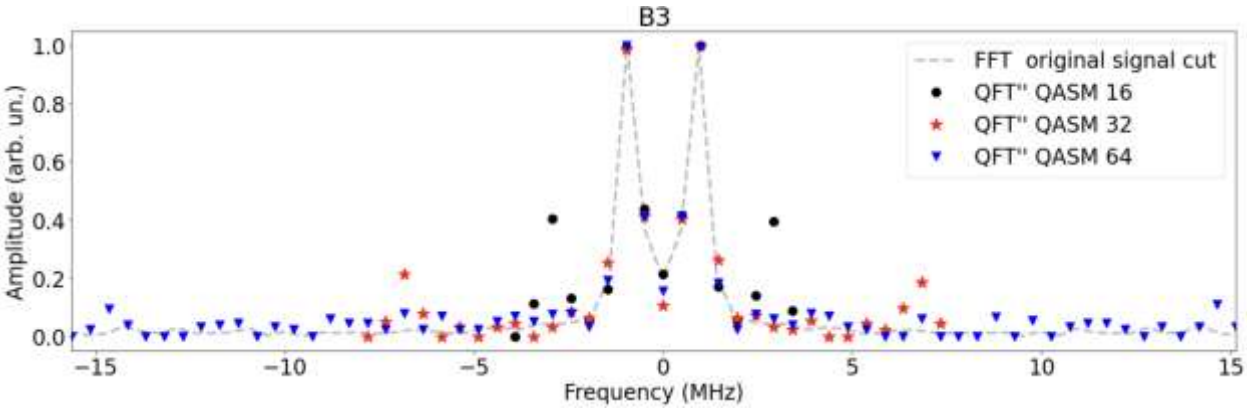
Next step is to prepare the quantum circuit for execution on the real quantum processing unit. We deal with the problem of high error rate, introduced by CNOT gates. In the Methodology section we presented four methods of CNOT gate reduction. We applied them to the composed circuits and obtained significantly smaller circuits (see Table 1). These circuits are not strictly equivalent to the classic Fourier transformation, but their results are acceptable with respect to our problem. See the results of these circuit executions in Fig. 7.



a



b



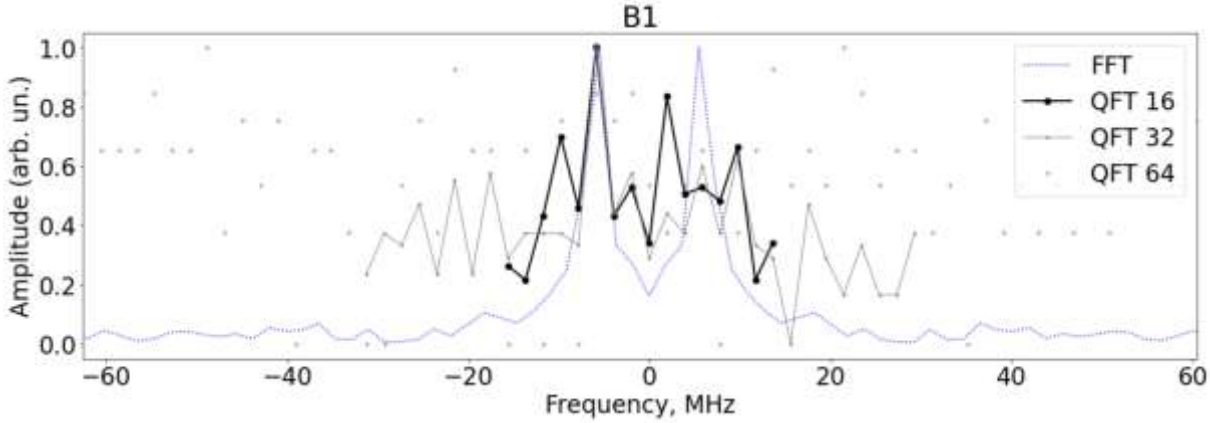
c

Fig. 7 Simulation results for the adjusted data of the **B1 (a)**, **B2 (b)**, and **B3 (c)**. There are aberrations in graphs (see outliers), which still does not affect the peak values. The dashed lines correspond to the Fourier transform of the experimental curve without the fast decay region at the beginning. The black circles, red stars, and blue triangles show the QFT on the simulator for time splitting the original data into 16, 32, and 64 points, respectively

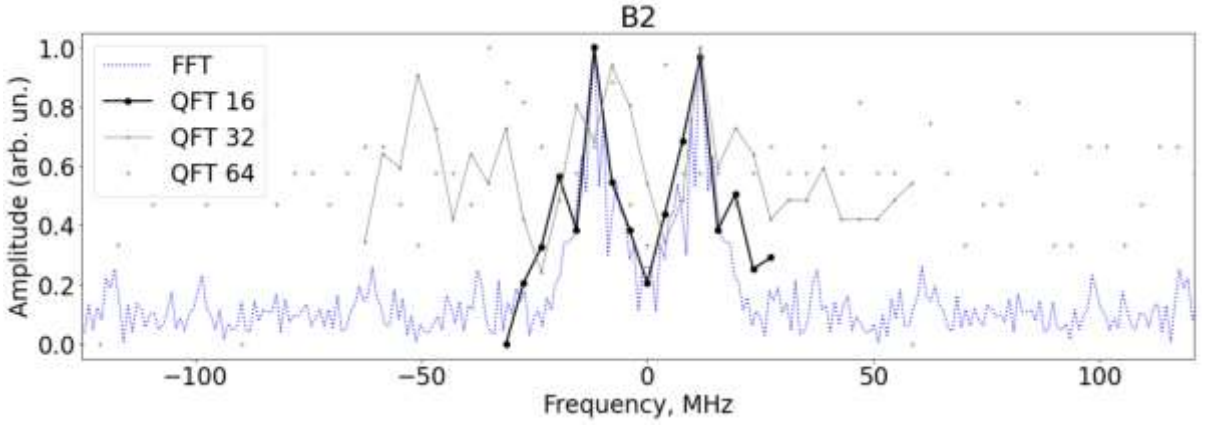
The curves shown in Fig. 7 were calculated on the simulator, considering all four optimization methods described above. Fig. 7 shows "outliers", which are the result of incomplete equivalence of methods. Nevertheless, the results agree quite satisfactorily with the data without optimization (compare Fig. 6 and 7). It should also be noted that when using the "optimized" QFT scheme, the best results were obtained for the **B3** biradical (see Fig. 7). We interpret this as follows. It can be seen from the experimental data for **B1**, **B2**, and **B3** biradicals (see Fig. 4) that the modulation frequency of the PELDOR signal decay for the **B3** biradical is almost an order of magnitude lower than for the **B1** and **B2** biradicals. To "optimize" the QFT circuit, approximations are used that distort the result of the transformation, primarily in the high-frequency region.

3.4 Experimental realization on a quantum computer

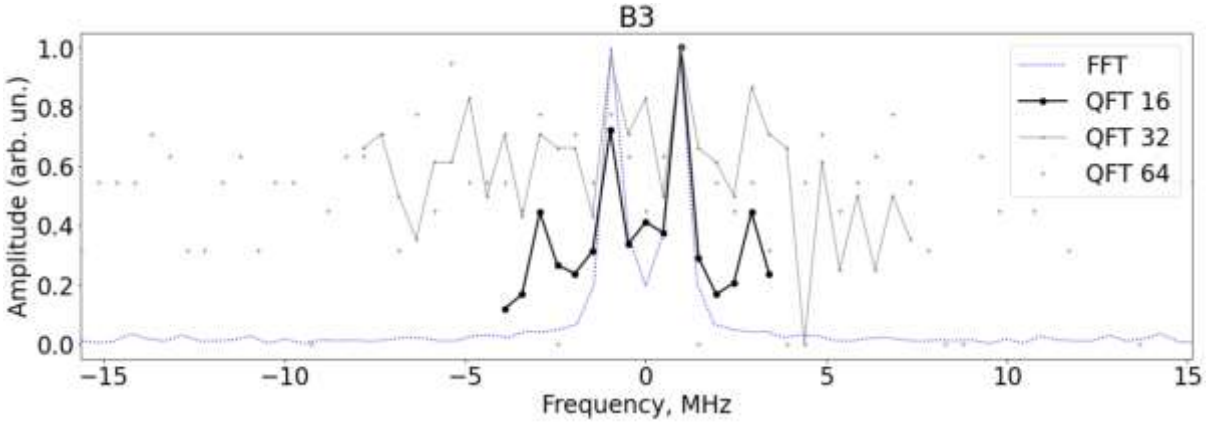
After validation of the algorithm, we proceeded with QPU experiments. We used the same quantum circuit on an 11-qubit IonQ trapped ion quantum computer. We observe that for 4 qubits our quantum results are correlated with classical for every dataset, while 6-qubit results are indistinguishable from the noise. Fig. 8 shows, that the frequency distributions produced by a 4-qubit system (in black) look similar for the desired deterministic distributions (in blue). Surprisingly, with the increase in qubits, distributions align with the classically predicted curve less. We explain this with the corresponding growth of the gates count in the quantum algorithm and, consequently, introduction of more severe errors.



a



b

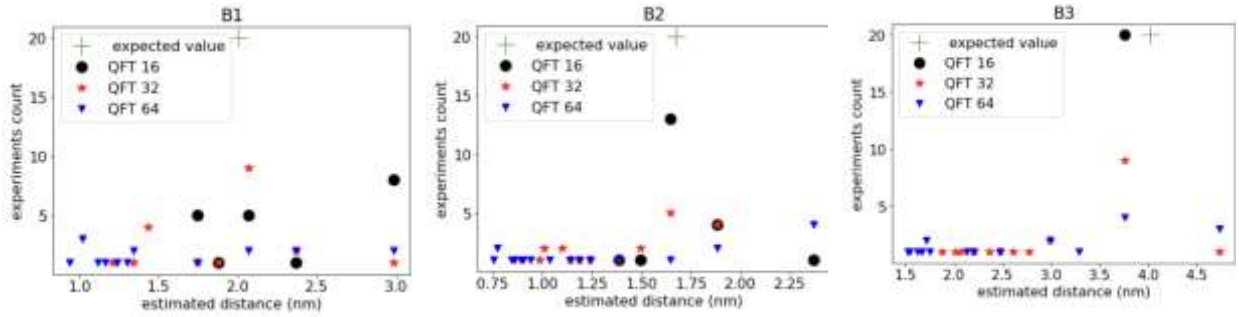


c

Fig. 8 Simulation results for the adjusted data of the **B1 (a)**, **B2 (b)**, and **B3 (c)**. The dotted lines correspond to FFT of the original experimental curves. The black dot lines, grey dot lines, and grey dots show the QFT on QPU for time splitting the original data into 16, 32, and 64 points, respectively

The consequence of the error increase inducted by the qubits increase, is that the initial problem of distance prediction is affected. With the drop of the precision of distribution, distance estimations become less reliable. In Fig. 9 we present the distribution of distance predictions. Fig 9b and 9c show that 4-qubit systems were able to make a correct prediction in most cases. In Fig.

9a, the 4-qubit system failed, but 5-qubit system was able to solve the task correctly 10 times out of 20.



a b c

Fig. 9 Distribution for distance measurements performed with a quantum computer for the **B1 (a)**, **B2 (b)**, and **B3 (c)**. Green cross corresponds to the correct value obtained with classical Fourier transform. Note that for B3 all predictions of QFT with 16 data points were the same, that is why we observe this as a single point

4 Results

For each biradical (**B1, B2, B3**), and for each qubit count (4, 5, or 6) we run and measure an optimized quantum circuit ten times. These ten measurements are the rough estimation of the distribution we obtain with simulation (see Fig. 7). As our method is looking for the peak frequency from the distribution, we select the most common value out of ten observed. If this value is equal to the one we obtain with classic computer using FFT, then we consider it correct.

In Table 1 we present the rate of correct estimations. One can see that with the growth of qubits the percent of correct estimations decreases. This can be explained with the explicit growth of the circuit (or quantum program) length. In classical computation we count primitive operators (memory access, arithmetic operations), which are executed sequentially. In quantum computing, correspondingly, we estimate the complexity of the program using simple gates. The difference is that single-qubit operations can be executed in parallel, if they are applied to different qubits. That is why we consider the shortest path from the initial state to the measurement operation through the quantum circuit. The length of this path computed in gates is called the quantum depth of the circuit. In the table one can see that our optimizations dramatically reduce this number.

Another important fact about the real QPUs is that CNOT operations are the biggest source of errors. Thus, another important metric of circuit quality is the number of CNOT operations. The bigger this number, the higher is the chance that the correct algorithm will have the wrong result due to the increase of probability of gate execution error. We also provide these numbers in Table 1. Please pay attention, that our methods for circuit reduction reduces CNOT count 2-3 times for each circuit as well.

Table 1 The table presents how circuit and data optimization techniques influence depth and CNOT count parameters, and shows, even for a low CNOT count, quantum computers significantly drop computation quality

Number of points, dataset	Original Initialize + QFT circuit		Improved data and optimized circuit		
	Depth	CNOT count	Depth	CNOT count	***Correct estimation, %***
B1 with 4 qubits	55	22	27	10	30
B1 with 5 qubits	117	52	**48**	**22**	**50**
B1 with 6 qubits	226	114	92	46	11.765
B2 with 4 qubits	50	22	**22**	**8**	**65**
B2 with 5 qubits	107	52	46	20	25
B2 with 6 qubits	243	114	94	46	5
B3 with 4 qubits	55	22	**16**	**6**	**100**
B3 with 5 qubits	117	52	48	22	45
B3 with 6 qubits	243	114	94	46	20

5 Discussion

On measurement accuracy. According to formula (1), frequency value D_0 relates to measured distance r as $D_0 = kr^{-3}$. Thus, higher frequency value corresponds to a smaller distance. Therefore, the constant frequency estimation error in higher frequency region leads to smaller error in distance determination.

On error model. The results of the implementation of the QFT, presented in Fig. 8, revealed unexpected feature: the obtained Fourier images are not symmetric with respect to zero frequency. This fact we consider as another manifestation of the errors inherent for specific qubits that are used in the quantum computers used by us.

IonQ documentation, as well as their benchmark paper [27] show only generalized statistics of errors, which include 1- and 2-qubit gate error, state preparation and measurement errors (SPAM).

We tried to guess the error nature by applying different error models. IBM Qiskit framework provides a generic error framework, which allows to model bit flip, readout, depolarization, reset, amplitude and phase dumping errors, and many others. We studied different error types and observed that any error which adds a symmetric effect on the data results with a symmetric distribution graph for our task (e.g. bit flip, depolarization).

We found that asymmetric readout error (i.e. $|1\rangle$ is more frequently read as $|0\rangle$ than vice versa) on just a single the least significant qubit was able to produce an asymmetric QFT image. Until now we did not find which particular asymmetric errors happen to operate to give those asymmetric QFT images presented in Fig. 8. At the moment, we can only say that we have identified the presence of some kind of asymmetric error. To suggest a possible source of such error, we must analyze the physics of the specific qubits used in our calculations.

On optimization techniques. We want to specifically mention that the first three optimization methods (reduction of rotation gates, post-processing instead of SWAP, and CNOT annihilation) are algorithm-agnostic, in other words they can be applied to any quantum circuit without considering its pragmatics; the fourth method (data adjustment) is specific for the initialization algorithm. For this method we account for two important properties, and we want to show that for our setting this optimization is acceptable. First, the data represents the periodic time signal and it is later analyzed in the frequency domain. Thus we totally control the type of aberrations which will be added to the frequency representation by this method. Second, the method is specific for the initialization algorithm [25] and allows at least halving the number of CNOT gates. CNOT gates today are the main source of errors for any physical platform [28], thus in practice the result of initialization will be even closer to the desired state, than the one following the exact procedure.

Our method is application-specific, but we can generalize it to just an approximate state initialization. This idea can be later developed to address precision properties of real QPU hardware. Currently, state decomposition is used for entanglement forging [36] which uses approximate operator decomposition for qubit number reduction. In future work we propose to address CNOT count with state tensor decomposition.

6 Conclusion

In this section we try to justify why learning quantum computations is worth the time investment.

In this work, we determined the distances between two localized unpaired electrons in three biradicals. To solve this problem, we applied the recognized method of recording the modulation of the decay of the envelope of the electron spin echo signal under conditions of

selective spin excitation of these electrons (PELDOR). This method of EPR metrology is based on the fact that the Fourier transform of the decay curve of the spin echo signals gives the energy of the spin-spin interaction. For distances between unpaired electrons greater than 1.5 nm, the main contribution to the spin–spin interaction comes from the dipole–dipole interaction. Therefore, the modulation frequency of the decay of the echo signal makes it possible to calculate the distance between unpaired electrons in the biradical.

But in this work, we wanted a little more. We wanted to see if it is already possible to successfully use the quantum computers available online today for Fourier transform and ultimately obtain the distance between electrons in a pair of paramagnetic particles.

From the very beginning, we had very moderate expectations for today's quantum computers. Therefore, we took it well that we did not get impressive results. But we were not disappointed either. It turned out that even on today's quantum computers it is possible to obtain the most probable frequencies that are close to those that can be obtained by implementing the DFT on classical computers.

Quite unexpectedly, the 4-qubit implementation of the QFT gave better results than the 6-qubit implementation. The opposite situation could be expected, since an increase in the number of involved qubits allows better, more accurate digitization of experimental data in greater detail. However, increasing the number of qubits increases the probability of errors. In our case, it turns out that already 6 qubits have enough probability to make serious errors.

One important statement which we want to make in this conclusion is that quantum computers use very different computation models, and we need to develop a habit of computations with such systems. Considering Fourier transform, we can observe that mathematically both DFT and QFT are similar, but of course there are fundamental differences at all implementation stages.

First, at the data input stage classical computers lose accuracy because of number representation (floating point representation) and these aberrations have a deterministic nature, while quantum computers encode the same numbers with continuous amplitudes. These amplitudes first should be normalized (and then renormalized) on a classic computer, and they are encoded with very complex equipment. Thus, the error in data has probabilistic nature. Second, classic computer programs propose only two options — either the program has terminated, or not. In quantum algorithms one can improve the quality of result with each consequent shot in a QPU. The more experiments are conducted, the closer is observed distribution to the underlying quantum state. One can literally buy accuracy for money. Third, quantum computations have probability in their nature: gate execution and measurement at current level of the hardware cannot be considered exact operation as we used to in classical computers. Being inaccurate in the very essence, quantum programs are still open for improvement. In our work we showed how by bringing a controlled

data precision error (with datapoint averaging) we reduce the error in the computation. This trade allows us in practice to make use of 5 qubits (32 data points), instead of being limited by 4 (16 data points).

What we learned from this experiment is already a well-known fact. Gate implementation and readout errors are the major enemies that prevent us from quantum supremacy in the very near future. A lot of scientists are approaching this abyss from both sides: physicists implement more reliable quantum objects and operations, while mathematicians improve algorithms to better survive in noisy contexts. There is a lot to be done. Just our single experience showed that even characteristics of the IonQ quantum computer are published in open access [27], there is not enough information to reproduce the error model. The feature of this error model is most probably hidden in its asymmetry, which we can only guess. We call on scientific groups who process experimental data with classic computers to give quantum computers a try. This can reveal a lot of interesting challenges for those who develop physics implementation, and also propose a valuable direction of future quantum research. It can result in creation of universal benchmarks, which can be used to assess a quantum computer as a white box, restoring its problematic nodes and gates.

Competing interests

The authors declare that they have no competing interests

Consent for publication

Not applicable

Ethics approval and consent to participate

Not applicable

Funding

This research was funded by Ministry of Science and Higher Education of the Russian Federation, registration number 122011800133-2

Authors' contributions

K.M. Salikhov and S.I. Protasov wrote the main manuscript text and S.I. Protasov performed all calculations on QPU. R.B. Zaripov performed measurements of experimental data. I.T. Khairutdinov wrote text about PELDOR and frequency distribution function. All authors reviewed the manuscript.

Acknowledgements

The authors acknowledge the financial support from the government assignment for FRC Kazan Scientific Center of RAS for supporting of this research activity and the CSF-SAC FRC KSC RAS for providing necessary facilities to carry out this work. The authors thankful to Prof. Gunnar Jeschke (ETH Zürich, Switzerland) and Prof. Yury G. Galyametdinov (Kazan National Research Technological University, Russia) for providing of the biradicals.

References

1. A. D. Milov, K. M. Salikhov, and M. D. Schirov, Fiz. Tverd. Tela (in Russ. **23**, 975 (1981).
2. A. D. Milov, A. G. Maryasov, and Y. D. Tsvetkov, Appl. Magn. Reson **15**, 107 (1998).
3. G. Jeschke, M. Pannier, and H. W. Spiess, Biol. Magn. Reson. Vol. 19 **19**, 493 (2000).
4. I. Krstić, R. Hänsel, O. Romainczyk, J. W. Engels, V. Dötsch, and T. F. Prisner, Angew. Chemie - Int. Ed. **50**, 5070 (2011).
5. M. Pannier, S. Veit, A. Godt, G. Jeschke, and H. . Spiess, J. Magn. Reson. **142**, 331 (2000).
6. P. P. Zänker, G. Jeschke, and D. Goldfarb, J. Chem. Phys. **122**, (2005).
7. O. Schiemann, P. Cekan, D. Margraf, T. F. Prisner, and S. T. Sigurdsson, Angew. Chemie - Int. Ed. **48**, 3292 (2009).
8. A. D. Milov, A. G. Maryasov, Y. D. Tsvetkov, and J. Raap, Chem. Phys. Lett. **303**, 135 (1999).
9. K. M. Salikhov, I. T. Khairuzhdinov, and R. B. Zaripov, Appl. Magn. Reson. **45**, 573 (2014).
10. P. Michael Marthaler, Sebastian Zanker, Iris Schwenk, Jan Reiner, Nicolas Vogt Juha, Peter Schmitteckert, Tomislav Piskor, Cedric Ziegler, S. Pinski, Keith Fratus, Benedikt Schönauer, Nicklas Enenkel, Konstantina Alexopoulou, Florian Eich, Sebastian Lehmann, Kirsten Bark, Reza Shirazi, and G. S. Álvarez Barcia, Damaris Kröger, Vladimir Rybkin, Fabio Covito, Florian Mazur, (2021).
11. 10 Full Stack Quantum Comput. Solut. Co. (2022).
12. S. I. Doronin, E. B. Feldman, and A. I. Zenchuk, Quant. Inf. Proc. **19**, 68 (2020).
13. S. I. Doronin, E. B. Feldman, E. I. Kuznetsova, and A. I. Zenchuk, Appl. Magn. Reson. **53**, (2022).
14. Y. D. Tsvetkov, M. K. Bowman, and Y. A. Grishin, *Pulsed Electron-Electron Double Resonance. Nanoscale Distance Measurement in the Biological, Materials and Chemical Science* (Springer, 2019).
15. G. Jeschke and H. W. Spiess, (1998).
16. Y. Polyhach, E. Bordignon, S. Gandra, A. Godt, and G. Jeschke, Phys. Chem. Chem. Phys. **14**, 10762 (2012).
17. G. Jeschke and Y. Polyhach, Phys. Chem. Chem. Phys. **9**, 1895 (2007).
18. Y. D. Tsvetkov, A. D. Milov, and A. G. Maryasov, Russ. Chem. Rev. **77**, 515 (2008).
19. G. Jeschke, Annu. Rev. Phys. Chem. **63**, 419 (2012).
20. L. D. Kispert, M. K. Bowman, J. R. Norris, and M. S. Brown, J. Chem. Phys. **76**, 26 (1982).
21. R. R. Schmidt, R. E. Martin, M. Pannier, F. Diederich, and V. Gramlich, 2833 (1998).
22. A. A. Sukhanov, M. D. Mamedov, K. Möbius, A. Y. Semenov, and K. M. Salikhov, Appl. Magn. Reson. **49**, 1011 (2018).

23. R. B. S. S. Gill, Adarsh Kumar, Harvinder Singh, Manmeet Singh, K. Kaur, Muhammad Usman, SOFTWARE-PRACTICE Exp. **52**, 66 (2022).
24. S. W. M. Treinish, J. Gambetta, D.M. Rodríguez, M. Marques, L. Bello, C. J. Wood, J. Gomez, P. Nation, R. Chen, E. Winston, J. Gacon, A. Cross, K. Krsulich, I.F. Sertage, S. Wood, T. Alexander, L. Capelluto, S. de la Puente González, J. Rubio, (2021).
25. S. S. B. and I. L. M. V. V. Shende, IEEE Trans. Comput. Des. Integr. Circuits Syst. **25**, 1000 (2006).
26. D. Coppersmith, IBM Res. Rep. RC19642 (1994).
27. K. Wright, K. M. Beck, S. Debnath, J. M. Amini, Y. Nam, N. Grzesiak, J. S. Chen, N. C. Pisenti, M. Chmielewski, C. Collins, K. M. Hudek, J. Mizrahi, J. D. Wong-Campos, S. Allen, J. Apisdorf, P. Solomon, M. Williams, A. M. Ducore, A. Blinov, S. M. Kreikemeier, V. Chaplin, M. Keesan, C. Monroe, and J. Kim, Nat. Commun. **10**, 1 (2019).
28. N. M. Linke, D. Maslov, M. Roetteler, S. Debnath, C. Figgatt, K. A. Landsman, K. Wright, and C. Monroe, Proc. Natl. Acad. Sci. U. S. A. **114**, 3305 (2017).
29. A. W. Harrow, A. Hassidim, and S. Lloyd, Phys. Rev. Lett. **103**, 1 (2009).
30. Alexei Y. Kitaev, Electron. Colloq. Comput. Complex. **3**, n. pag. (1996).
31. P. Chapman, J. Kim, T. Kramer, and C. Monroe, https://ionq.com/best (n.d.).
32. R. T. Weber, *Pulsed Eldor Option User's Manual.* (Bruker BioSpin Corporation, 2006).
33. M. V. Strelkov, R. B. Zaripov, V. I. Dzhabarov, A. A. Knyazev, K. M. Salikhov, V. K. Voronkova, and Y. G. Galyametdinov, Russ. Chem. Bull. **57**, (2008).
34. A. Godt, C. Franzen, S. Veit, V. Enkelmann, M. Pannier, and G. Jeschke, J. Org. Chem. **65**, 7575 (2000).
35. G. Jeschke, ChemPhysChem **3**, 927 (2002).
36. S. S. A. Eddins, M. Motta, Tanvi P. Gujarati, S. Bravyi, A. Mezzacapo, Charles Hadfield, PRX Quantum (2021).