Quantum Algorithms on NMR SpinSolve system

Shashank Dharanibalan[a)]

This lab report shows that an NMR system can be used for quantum computation on a two-qubit system comprised of 1H nuclei and 13C nuclei in a sample of Chloroform. The report demonstrates the application of the pulse sequences that rotate the spins of the respective nuclei to apply the two-qubit quantum Controlled X gate or CNOT Gate. Using sequence of operations, the Deutsch-Josza algorithm is implemented to show that constant functions yield the same state and the balanced functions both yield the same state characterized by two distinct spectra for the four cases. Additionally, the Grover's Search algorithm can be implemented to show that the marked state can be obtained from a single application of the algorithm and that successive applications of the algorithm with a period of 3 yields the same marked state, albeit with 50 % lower signal amplitude between cycles.

Keywords: Rotation, Coupling, Spectrum, Temporal Averaging, Thermal State, Composite Pulses

I. INTRODUCTION

Quantum Computation is a method of information processing that relies on the reversible transformations of the states of quantum systems. These transformations are impossible to conduct on classical computation systems. The information gain in quantum computation is exponentially large to be processed on a classical computer[3].

The ability to process such a large quantity of information on these quantum systems spurred the creation of algorithms that can exploit the laws of quantum mechanics to find solutions with a faster runtime and lower cost than classical analogs. The two noteworthy algorithms are the Deutsch-Jozsa algorithm and the Grover's Algorithm. The Deutsch-Jozsa algorithm helps provide a quantum speedup to determine the nature of a Boolean function with just a single query. Similarly, Grover's search algorithm finds the solution of a database, represented by a quantum state, faster but with the trade-off of being probabilistic instead of deterministic in the case of a classical computer.

The implementation of these algorithms requires the use of a quantum system with well-characterized qubits whose states can be manipulated through unitary transformations called "quantum logic gates". This report uses the liquid state NMR system with the sample of Chloroform as a quantum system. The Chloroform sample contains Hydrogen and Carbon nuclei in the molecules which are the two qubits. Their intrinsic spin is the quantum state that can be manipulated and measured to demonstrate the two algorithms.

II. THEORY

The larger-scale implementation of these algorithms requires reduced interference from the environment and remains one of the major challenges for reliable and scalable quantum computing. The NMR system operates at room temperature and a clever averaging process is used to obtain a classical signal from the quantum projective measurement of the spins of the two qubits in the ensemble which are in a thermal Boltzmann distribution at this temperature. The averaging process is designed to obtain an intuitive repsentation of the computational basis of qubits from the NMR spectrum. Similarly, the design of pulse

sequences in the NMR interface act on the qubit to perform one-qubit and two-qubit logic gates that alter the state as needed by the algorithm in the same measurement basis.

II.A. Nuclear Magnetic Resonance Hardware

The Magitrek SpinSolve NMR system is used with a sample of Chloroform which has a heteronuclean spin-1/2 system of 1H and 13C nuclei. Each nucleus type represents two well-characterized qubits as they can inhabit the positive of negative spin state respectively which is analogous to the $|0\rangle$ and $|1\rangle$ state initialization in the computational basis. This sample is placed in a static and homogenous magnetic field which aligns the spins in the $\hat{z}$. The nature of a heteronuclear system implies the presence of spin-spin coupling called Fermi-contact interaction (J) due to the chemical bond between electrons of the Carbon and the Proton[1].

The spins do not remain static and evolve with time as dictated by the Hamiltonian of the system and Schrödinger's equation. The Hamiltonian in our spin system can be defined by the following, where I_z represents the proton spin, S_z represents the carbon spin and J represents the coupling between the two spins. The other coupling terms are ignored as they average out in this liquid state of Chloroform.

$$H_{int} = -\omega_0^{(I)} I_z - \omega_0^{(S)} S_z + 2\pi J I_z S_z \tag{1}$$

The manipulation of the spins is the crux of quantum computation and it can be achieved by constructing universal quantum gates that operate on the two qubits in a linear order. The construction of these gates can be achieved by pulses and internal Hamiltonian evolution as seen in Eqn 1. The pulses are used in sequence as building blocks to achieve gate operations and eventually an algorithm. The pulses are applied using an external Hamiltonian in the rotation frame, such that the internal Hamiltonian disappears. The hardware implementation is achieved by external radiofrequency (RF) pulses in the transversal $\hat{x}$ & $\hat{y}$[6].

$$H_{ext} = \gamma_I B_I \left(\cos\left(\omega_0^{(I)} t + \phi\right) I_x - \sin\left(\omega_0^{(I)} t + \phi\right) I_y \right) \tag{2}$$

Eqn 2 gives the external Hamiltonian for the manipulation of the spin of Hydrogen, where B_I refers to the pulse amplitude. A similar Hamiltonian can be constructed for the spin of Carbon with a different pulse amplitude and RF frequency. However, when both spins are manipulated simultaneously, the same pulse duration is used whilst the pulse amplitude (B_1) must be recalibrated for Hydrogen as a pulse on Carbon takes longer than Hydrogen. The external Hamiltonian can only pulse for the $\hat{x}$ and $\hat{y}$, so the free evolution of the internal Hamiltonian under J-coupling can also evolve the state about the $\hat{z}$, completing full control of the qubit state on the Bloch sphere.

The nuclear spins eventually decohere due to the inhomogeneity of the static magnetic field from the internal Hamiltonian and the thermal bath that is the environment around the sample leading to a Free Induction Decay (FID) of the time domain signal. The RF coils can detect the magnetization in the transverse field and can be used as a readout measurement of the state after implementing a pulse sequence, gate or algorithm. Pauli observables outside the x-y plane can be rotated into the x-y plane first and then measured in the allowed basis by fourier transforming the decaying time domain signal.

II.B. Single and Two-Qubit Gates

The use of the pulses and free evolution in a sequence is used to create universal gates like X gate, which flips the bit in computational basis and H gate which places the qubit state in a superposition. The X gate involves a π pulse along the $\hat{x}$ which trivially prepares the opposite state after application of the gate. The H (Hadamard) gate takes bring any state to the x-y plane of a superposition. The first pulse brings any states into the x-y plane if was in the $\pm\hat{z}$ and the second pulse places places it back in the x-y plane if it was in the x-y plane already.

$$X_1 = I_X(\pi) \tag{3}$$

$$H^{\otimes 2} = H_1 + H_2 = (I_X + S_X)(\pi)(I_Y + S_Y)\left(\frac{\pi}{2}\right) \tag{4}$$

Eqn 3 & 4 shows the pulse sequence for each of the gates. The nomenclature is that the pulses are rotation matrices that exponentiate the product operator with the specified

angle. This allows for simulation of these pulses for verification using linear algebra and tensors because the gates are reversible and are unitary transformations. The Hadamard gate shows the summation of two product operators to denote a simultaneous pulse on both Hydrogen and Carbon channels.

Similarly, a two-qubit gate like CZ can be made using a pulse sequence. The CZ gate is a conditional gate that applies the Z phase gate when the control qubit is in the $1\rangle$ state. The CZ gate requires an alternative composite pulse sequences because a rotation pulse alone $\hat{z}$ for just one of the qubits is not natively possible in this NMR system. A composite pulse refers to a sequence of pulses that are nearly equivalent to another single rotation pulse.

$$CZ = I_Z\left(\frac{\pi}{2}\right) S_Z\left(\frac{\pi}{2}\right) I_Z S_Z\left(-\pi\right) \tag{5}$$

$$CZ = I_X\left(\frac{-\pi}{2}\right) I_Y\left(\frac{\pi}{2}\right) I_X\left(\frac{\pi}{2}\right) S_X\left(-\frac{\pi}{2}\right) S_Y\left(\frac{\pi}{2}\right) S_X\left(\frac{\pi}{2}\right) U\left(\frac{1}{2J}\right) \tag{6}$$

Eqn 6 shows the full pulse sequence native to the NMR system's pulsing capabilities. The final U rotation refers to free evolution of duration 1/2J, which is the time taken for both spins to precess about $\hat{z}$ for about π. These gates become the building blocks for algorithm implementation

II.C. Temporal Averaging

The NMR SpinSolve system operates at room temperature with a liquid solution. Therefore, the spins are not all intialized in the computational basis but are infact in completely random directions in a Maxwell-Boltzmann thermal distribution given by Eqn 7. Without good characterization of the qubit states in a computational basis, no good information can be processed and extracted.

$$\rho_{in} = \frac{1}{4}\mathbb{I} + 10^{-4}\begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & -3 & 0 \\ 0 & 0 & 0 & -5 \end{bmatrix} \tag{7}$$

The process of Temporal Averaging for this two-qubit system helps average out the spectrums across three separate trials with three initial permutations pulses. The resultant spectrum between these three data trials is in computational basis for useful interpretation. Using the linearity of quantum logic, the spectrum obtained for each of these when averaged together, alters the initial thermal density matrix into a "pseudo-pure" state[5].

Supppose, the $|00\rangle\langle 00|$ needs to be prepared, the populations in the other states in the deviation matrix as seen above are noise. This can be averaged out. The zero-th permutation is identity with the same populations as in the deviation matrix, the first permutation shifts the populations along the trace of the matrix, $|01\rangle\langle 01|$ to $|10\rangle\langle 10|$, $|10\rangle\langle 10|$ to $|11\rangle\langle 11|$ and $|11\rangle\langle 11|$ to $|01\rangle\langle 01|$. The second permutation is the inverse of the first permutation and cycles the population accordingly as seen in Eqn 8-10.

$$P_0 = I^{(4)} \tag{8}$$

$$P_1 = S_X\left(\frac{\pi}{2}\right) I_Z S_Z\left(\frac{\pi}{2}\right)(S_Y + I_X)\left(\frac{\pi}{2}\right) I_Z S_Z\left(\frac{\pi}{2}\right) I_Y\left(\frac{\pi}{2}\right) \tag{9}$$

$$P_2 = I_X\left(\frac{\pi}{2}\right) I_Z S_Z\left(\frac{\pi}{2}\right)(S_X + I_Y)\left(\frac{\pi}{2}\right) I_Z S_Z\left(\frac{\pi}{2}\right) S_Y\left(\frac{\pi}{2}\right) \tag{10}$$

$$P_0\rho_{in}P_0^\dagger = \rho_{in}$$

$$P_1\rho_{in}P_1^\dagger = \frac{1}{4}\mathbb{I} + 10^{-4}\begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & -3 & 0 & 0 \\ 0 & 0 & -5 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix}$$

$$P_2\rho_{in}P_2^\dagger = \frac{1}{4}\mathbb{I} + 10^{-4}\begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & -5 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & -3 \end{bmatrix}$$

The sum of these density matrices for each of these permutations will create an effective pure state with effective population only in the $|00\rangle\langle 00|$ state.

$$\Sigma_{i=0}^{2} P_i \rho_{in} P_i^\dagger = \frac{1 - 1.67 \times 10^{-4}}{4} \mathbb{I} + 10^{-4} \begin{bmatrix} 6.67 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

II.D. State Preparation

The temporal averaging process can be conducted on the other states as well using an X gate on the respective qubits as needed for each measurement for the three permutations. The process now provides us with tools to prepare the states in the computational basis and the pulses to apply gates. The expectation of how each state can be identified on the spectrum is crucial, for an arbitrary density matrix, the populations for each state can be calculated for a spectrum by the area underneath the FID signals for both channels.

$$\rho = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & d \end{bmatrix}$$

The temporal averaging process along with correct X gate implementation will prepare the effective pure state with the population of the spins in one of the states. Therefore, using Fig 1, the expected spectrum for each of the states can be seen in Fig 2.

II.E. Deutsch-Jozsa Algorithm

The Deutsch-Jozsa algorithm was one of the first quantum algorithms conceived that helps find the nature of a Boolean function with a single query of the algorithm instead of N/2 queries in a classical algorithm with 'N' bits. The algorithm in this NMR system has four oracle cases which are the gates that represent the Boolean functions. There are two constant functions, that always

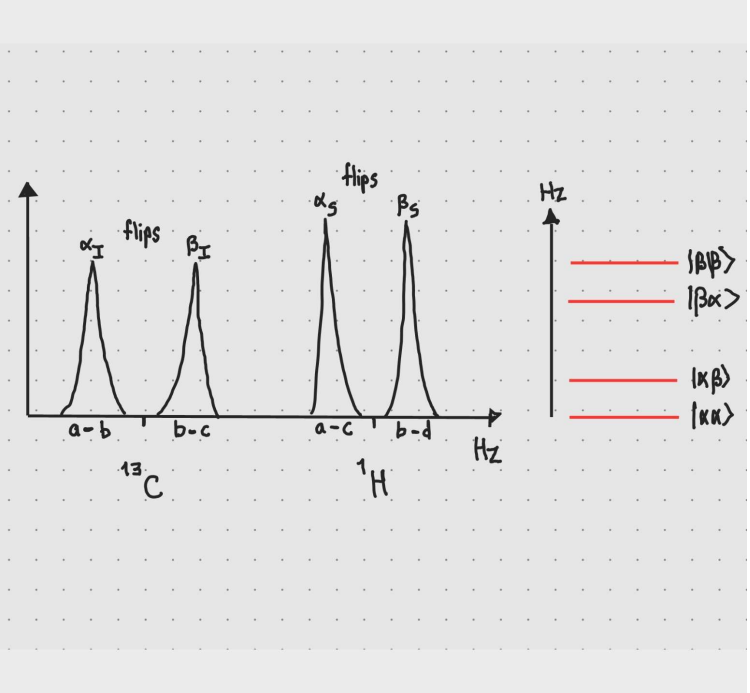


Fig. 1: The populations in the arbitrary density matrix are distributed as such as peak integrals for the four coupled peaks from an FID of the initial thermal state of the NMR sample

return the same bit no matter than input, and there are two balanced functions that return the bit an equal number of times[2].

The complement implementation of the algorithm can be done by initialising the state to $|01\rangle$ and placing each of the four oracles between Hadamard gates on both qubits before reading out the spectrum as a FID. Eqn 11-14 show the full pulse sequence in terms of the gates defined so far.

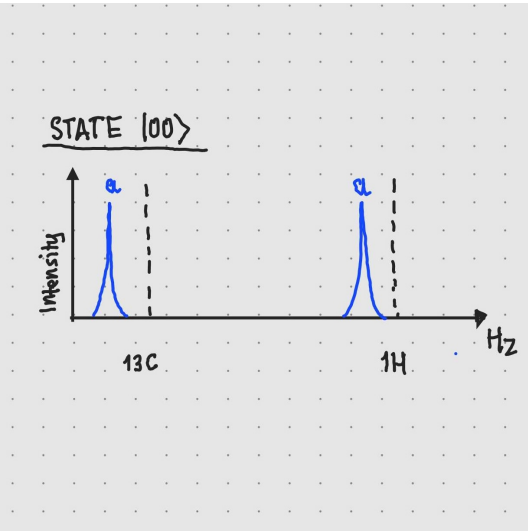
$$DJ_1 = X_2 H^{\otimes 2} \mathbb{I} H^{\otimes 2} Acq \tag{11}$$

$$DJ_2 = X_2 H^{\otimes 2} X_2 H^{\otimes 2} Acq \tag{12}$$

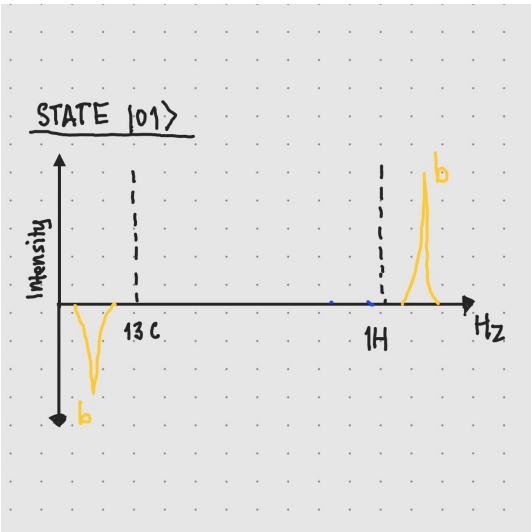
$$DJ_3 = X_2 H^{\otimes 2} H_2 CZ H_2 H^{\otimes 2} Acq \tag{13}$$

$$DJ_4 = X_2 H^{\otimes 2} H_2 CZ H_2 H^{\otimes 2} Acq \tag{14}$$

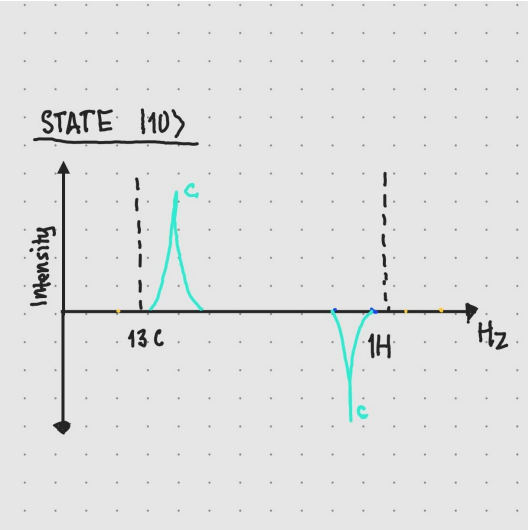
Eqn 11 & 12 are the constant functions and the Eqn 13 & 14 are the balanced functions with CNOT gate in the oracle constructed by CZ gates.



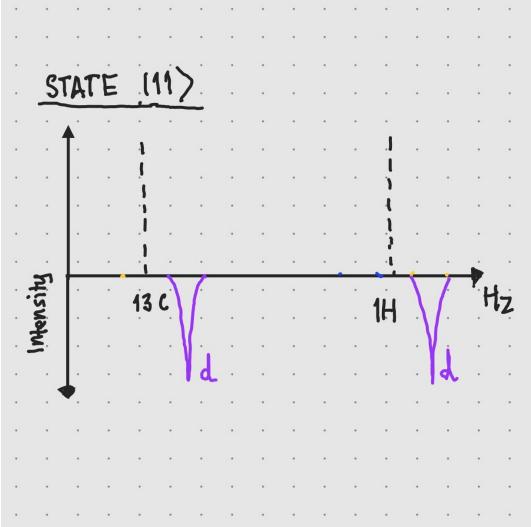
(a) Expected Spectrum for State $|00\rangle$



(b) Expected Spectrum for State $|01\rangle$



(c) Expected Spectrum for State $|10\rangle$



(d) Expected Spectrum for State $|11\rangle$

Fig. 2: The diagrams show the expected peaks and peak integrals for each state after state preparation and temporal averaging.

II.F. Grover's Algorithm

Grover's algorithm is a database searchign algorithm which finds a marked solution in a given database with a speedup but the drawback of arriving at a probabilistic solution instead of a deterministic one in the classical sense. However, the Grover process can be repeated periodically to improve the amplitude and thereby the probability as well.

$$G = H^{\otimes 2} Z_0 H^{\otimes 2} Z_f$$

$$=> Z_0 = I_X \left(\frac{\pi}{2}\right) I_Y \left(\frac{\pi}{2}\right) I_X \left(\frac{-\pi}{2}\right) S_X \left(\frac{\pi}{2}\right) S_Y \left(\frac{-\pi}{2}\right) S_X \left(\frac{-\pi}{2}\right) I_Z S_Z \left(\frac{\pi}{2}\right)$$

$$=> Z_{f=3} = CZ$$

$$|x_k\rangle = GH^{\otimes 2}|00\rangle$$

III. PROCEDURE

1. Using a FID, center frequency is calibrated to get the on-resonance pulsing and so that external Hamiltonian is in the rotating frame. The J-coupling strength can be calculated from both calibrating both Carbon and Hydrogen Spectrum

2. The pulse length calibration is done for both channels to improve signal amplitude during measurement readout and maximize signal-to-noise ratio (SNR)

3. The pulse sequences are applied like building blocks for the algorithm and a small delay is added between pulses for the NMR system to prevent accidental saturation.

4. The state is measured by an FID and the location of the peak is noted from the fourier transformed lineshape.

5. The same algorithm process is repeated with each of the three permutation applied first. The measured lineshapes of the spectrums are averaged together.

6. The area under the peak is numerically computed to obtain the peak integral which characterize the output state of an algorithm

IV. FIGURES

IV.A. Approximate CNOT and Exact CNOT gate

The following equations show the pulse sequence to run the two-qubit approximate and exact CNOT gate. The approximate CNOT gate requires a shorter pulse sequence but is off the exact

CNOT by relative phase. The exact CNOT can be constructed using the CZ gate defined earlier with two Hadamard gates on the target Carbon qubit spin or by the following sequence which builds off of the approximate CNOT pulses using two Z rotations to correct the relative phase.

$$\text{CNOT}_{appx} = e^{\frac{i\pi}{4}} S_Z\left(\frac{-\pi}{2}\right) S_X\left(\frac{\pi}{2}\right) 2I_Z S_Z\left(\frac{\pi}{2}\right) S_Y\left(\frac{\pi}{2}\right) I_Z\left(\frac{\pi}{2}\right)$$

$$\text{CNOT} = S_X\left(\frac{-\pi}{2}\right) S_Y\left(\frac{-\pi}{2}\right) S_X\left(\frac{\pi}{2}\right)\ S_X\left(\frac{\pi}{2}\right) U\left(\frac{1}{2J}\right) S_Y\left(\frac{\pi}{2}\right) I_X\left(\frac{-\pi}{2}\right) I_Y\left(\frac{\pi}{2}\right) I_X\left(\frac{\pi}{2}\right)$$

Before data collection, the pulse sequences above were simulated using rotation matrices to produce the composite unitary matrix. This exact CNOT gate still has relative phases due to our pulse construction, but the gate still works as intended as the phase is on both control and target qubit so they effectively cancel out. For the sake of continuity, this gate sequence is used consistently throughout the rest of the report[4].

$$CNOT_{appx} = \begin{pmatrix} -i & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -i \\ 0 & 0 & -1 & 0 \end{pmatrix} \qquad CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

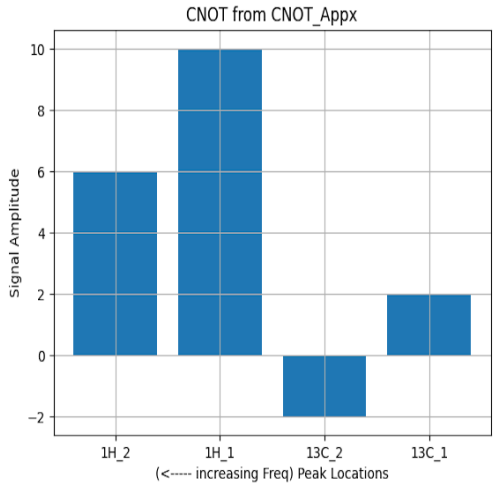


Fig. 3: The axes are flipped along the vertical axis to match the User Interface of the SpinSolve software. The purpose of the simulation was to act a reference of the expected spectrum for each FID measurement on the actual NMR system

Using the simulated unitary matrix for each of the two gates on the initial thermal state of the NMR system in Eqn 7, the spectrum of appropriate peak heights was plotted using the scheme described in Fig 2, to ensure the FID readout matches the expected spectrum.

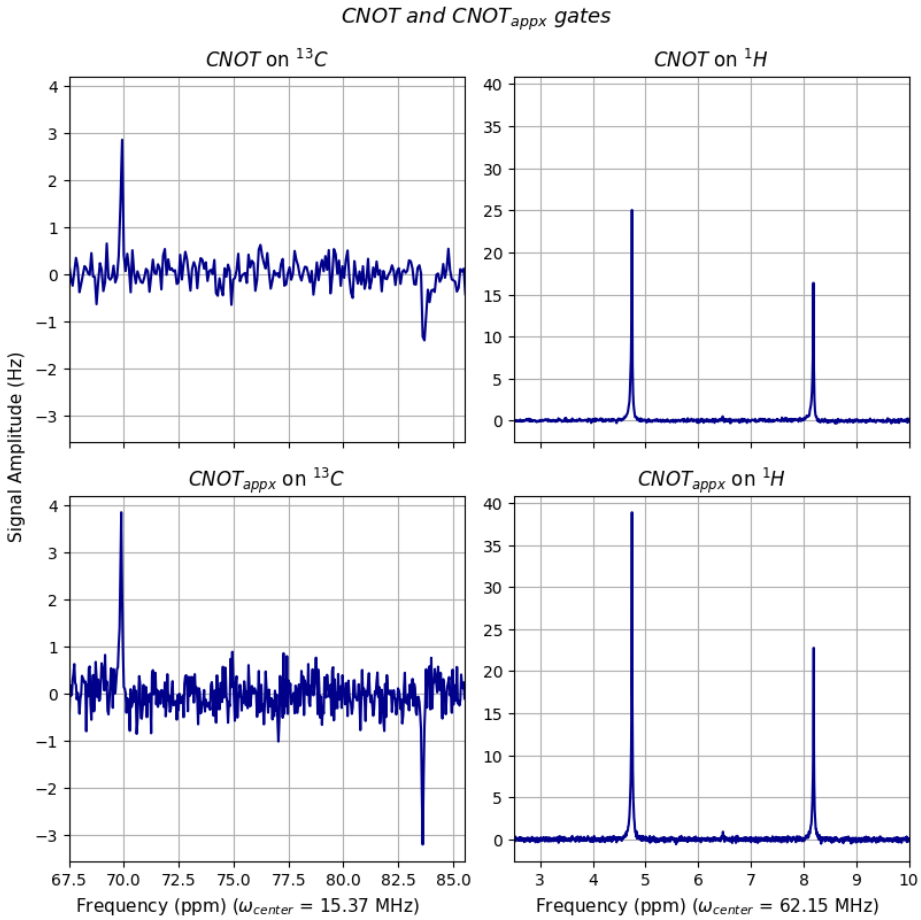


Fig. 4: The resulting spectrum on both Carbon and Hydrogen channels is shown using the pulse sequences above. The approximate CNOT is a shorter pulse to readout but has more noise than the exact CNOT. The spectrums both match the expected simulations

Firstly, the comparison with simulated spectrum in Fig 3 shows that the gates result in a similar spectrum and match the shape and peak ratios by qualitative appearance for each of the channels. This check assures that the pulse sequence worked as intended and matches simulated matrix's operation on the thermal state.

The approximate CNOT gate may appear to have a larger signal amplitude than the Exact CNOT gate suggesting improved performance and fidelity of the spectrum, but the SNR is actually lower for the Approximate gate as seen on the Carbon channel. Additionally, the simulated spectrum plots the peak integrals, not the signal amplitude. The process of temporal averaging and complete spectrum readout are necessary to ascertain of the gates can be successfully implemented and are of high fidelity.

IV.B. Temporal Averaging

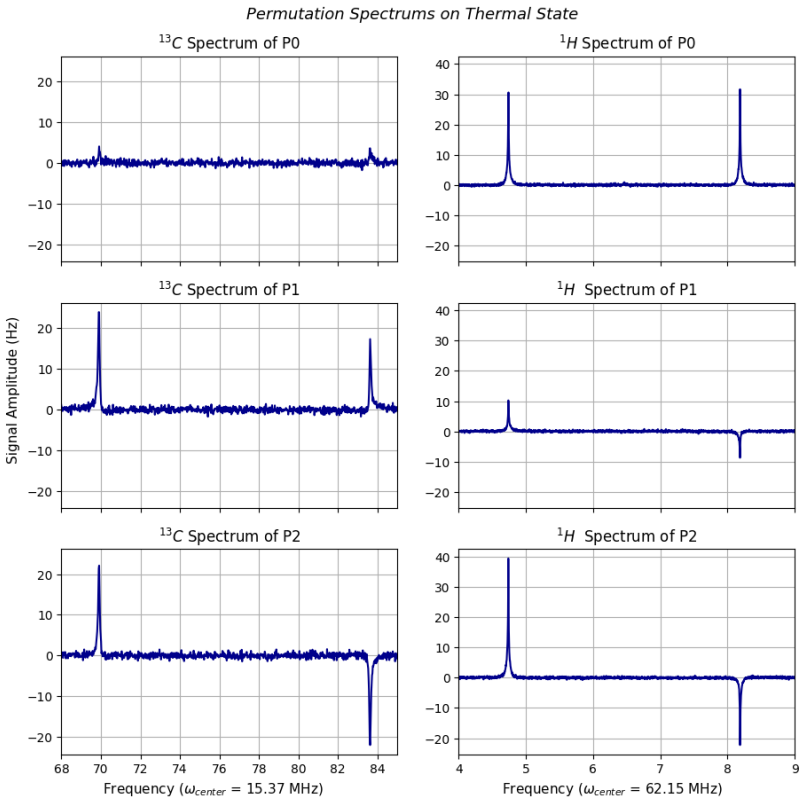


Fig. 5: Each row show the Carbon and Hydrogen spectrum for each Permutation operation pulse sequence on the thermal state. to create ane effective pure state for $|00\rangle\langle 00|$

Fig 5 shows how the populations in the states that constitute noise are cycled effectively for each permutation. It must be noted that the Carbon spectrums is significantly more noisy than the Hydrogen spectrum. The signal amplitude is low and the SNR is not very high, despite running 16 scans for each of the three permutation trials, setting adequate repetition time of 15,000 ms and an appropriately succinct acquisition window to avoid including noise in the FID readout.

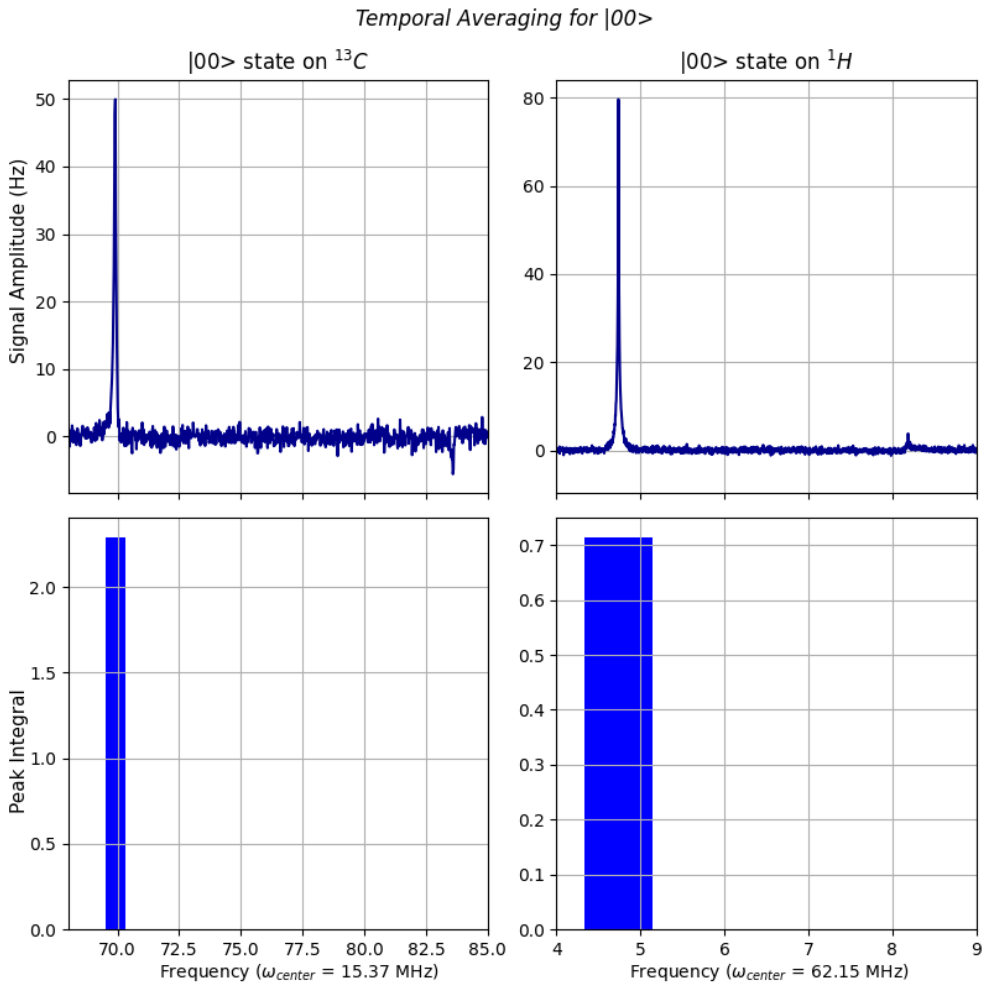


Fig. 6: The aggregation and then averaging produces the desired pseudo-pure state. The noise in the Carbon channel despite multiple scans results in a larger amplitude because the noise is also integrated

The first row of figures shows the averaged spectrums of the permutations, resulting in an spectrum of the $|00\rangle$ effective pure state matching as expected by Fig 2. However, the integrated bar plots in the bottom row do not have same height as the carbon data integrates the noise as well.

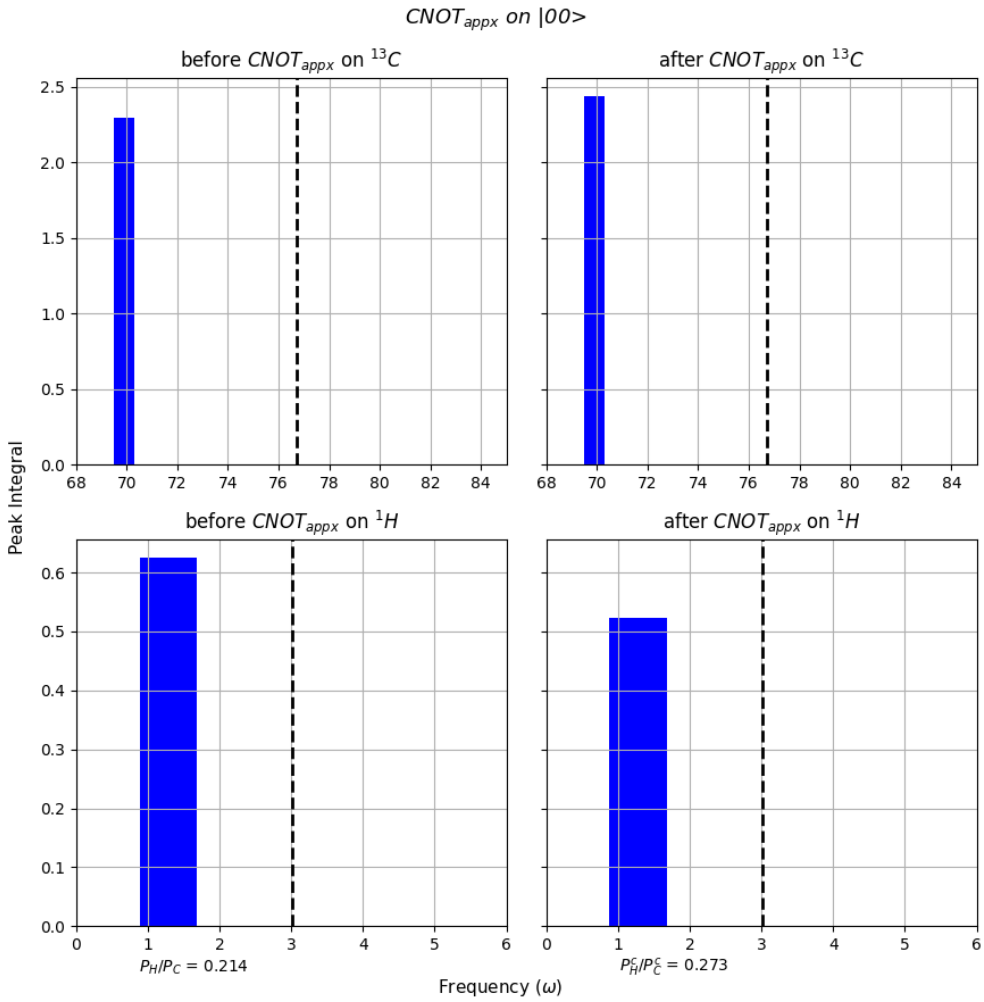


Fig. 7: Similarly, we can see in the case of Approximate CNOT, the ratios may not be as consistent as the exact CNOT. The differing ratios can be used to characterize the noise quantitatively apart from the qualitivatie method of observing stray peak integrals

Using the fact that the peak integrals are not identical between Carbon and Hydrogen spectrums after temporal averaging, the ratio of the peaks before and after the gate can be shown to test of the gate is successfully implemented as the ratio must be similar else there was noise involved and there are other peaks in the spectrum that did not cancel out.

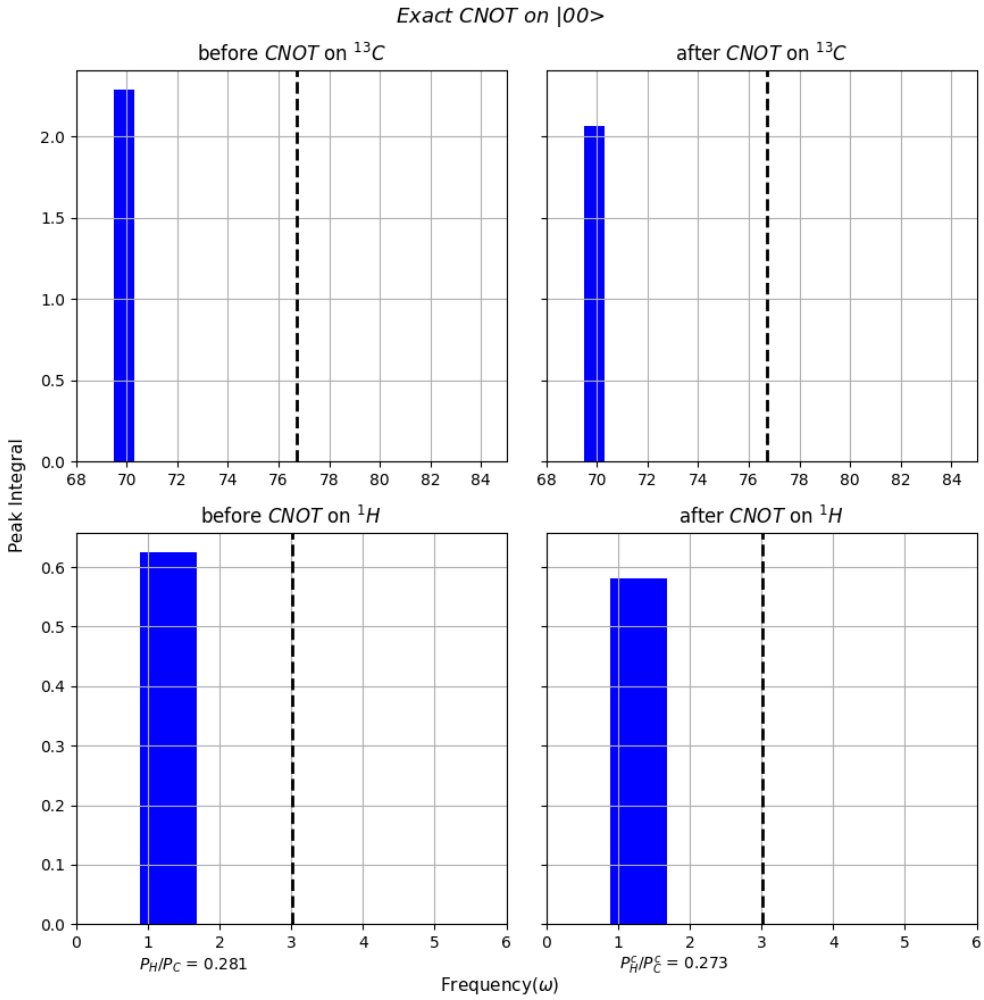


Fig. 8: The Temporal Averaging for both spectrums can be used to show that the gate has been consistently implemented in this system by comparing the ratio of the peak integrals between the ^{13}C and ^{1}H plots, before and after the gate sequence

Therefore, The ratio between the Hydrogen and Carbon peak before the Approximate CNOT gate is $P_{^1H}/P_{^{13}C} = 0.272107$ and the ratio after is $P_{^1H}^{CX}/P_{^{13}C}^{CX} = 0.214115$. The ratio between the Hydrogen and Carbon peak before the CNOT gate is $P_{^1H}/P_{^{13}C} = 0.272107$ and the ratio after is $P_{^1H}^{CX}/P_{^{13}C}^{CX} = 0.282932$

The same data used for pulsing on the Hydrogen and Carbon before application of the CNOT and CNOT Approximate, the pulsing of CNOT and CNOT approximate were done withe same parameter settings and pulse length calibration. Therefore, Fig 6 and 7 show a complete description of both spectrums before and after applying the Approximate and Exact CNOT gate and assuredly

show that there is successful gate implementation.

IV.C. Truth Table for CNOTs

Assured that the CNOT gates is implemented successfully, the truth table can be constructed for the application of the CNOT gates on each of the four states. The expectation is that the CNOT will leave the $|00\rangle$ and $|01\rangle$ as is and swap the spectrums of $|10\rangle$ and $|11\rangle$.

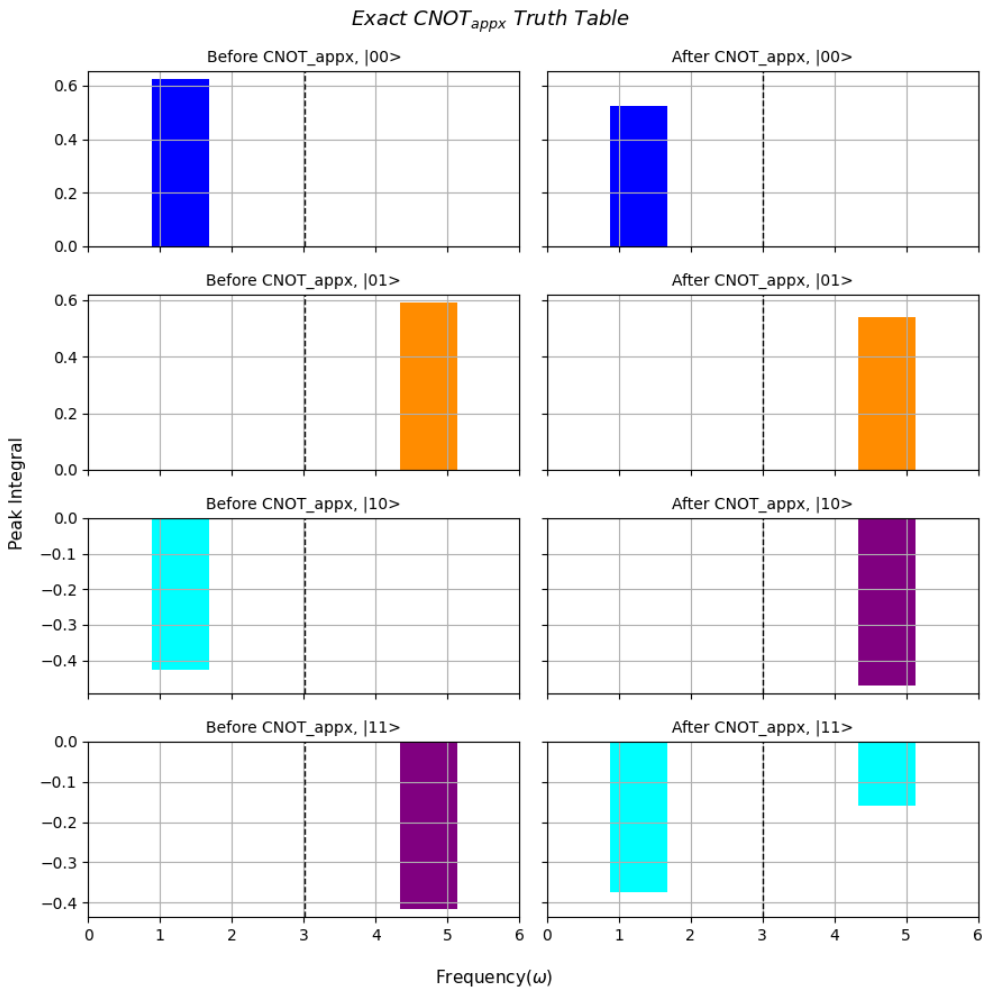


Fig. 9: The bar plot represents the integrated peaks of each permutation spectrum without and with approximate CNOT, averaged to produce the desired pseudo-pure state. There is evidently more noise as the integration also accumulates and then averages the noise as seen in the smaller bars which are not part of the desired state.

There is some noise detected as some stray peak integrals are still observed after averaging,

this would lower the measured peak integral at the expected location, lowering the fidelity of the operation.

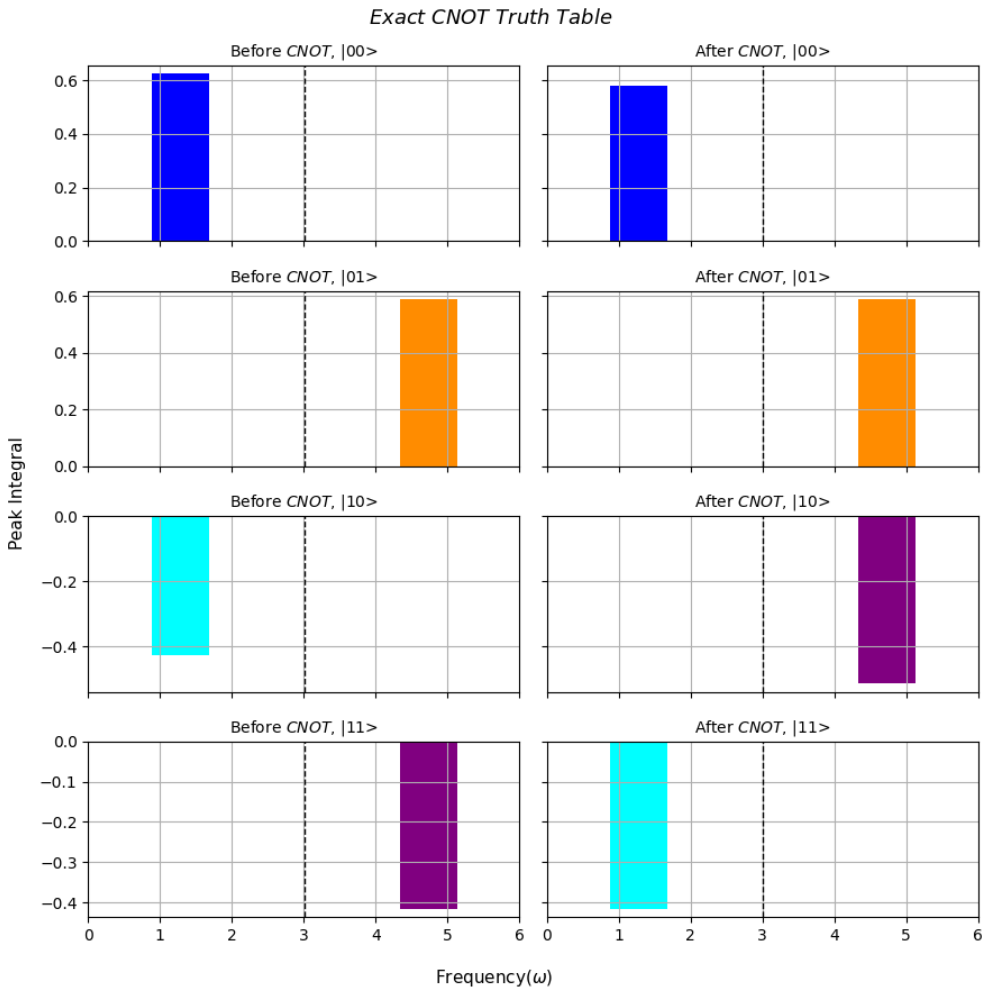


Fig. 10: The bar plot represents the integrated peaks of each permutation spectrum without and with CNOT, averaged to produce the desired pseudo-pure state. The color coding shows how the truth table matches the expectation of how the CNOT gate works on our four two-qubit states.

The fidelity can be calculated by measuring as fraction, how close the resulted spectrum's peak integral was to the expected peak integral. The average fidelity of the gate operation can be calculated using fidelity for each state to compare the fidelity of the approximate and the exact CNOT gate. The Exact CNOT gate is 0.5 % better than the Approximate CNOT Gate in terms of its Fidelity from the Truth Tables.

State	Exact CNOT	Approx CNOT
$\|00\rangle$	0.9295	0.8356
$\|01\rangle$	0.9973	0.9110
$\|10\rangle$	0.7644	0.8711
$\|11\rangle$	0.9734	0.8726
Average Fidelity	0.9162	0.8726

Table I: Table of the Fidelities of the gate operation on each state. The averaged fidelity is better for the Exact CNOT gate than the Approximate CNOT as it rotates the states as expected despite larger pulse sequence length

IV.D. Deutsch-Jozsa Algorithm

This algorithm's implementation should show that for the initialised $01\rangle$ state, the constant functions in the oracle of the algorithm, should remain the same state,returning $|01\rangle$. While the Balanced functions should result in $|11\rangle$. For the four oracles, there will be two distinct spectrums.

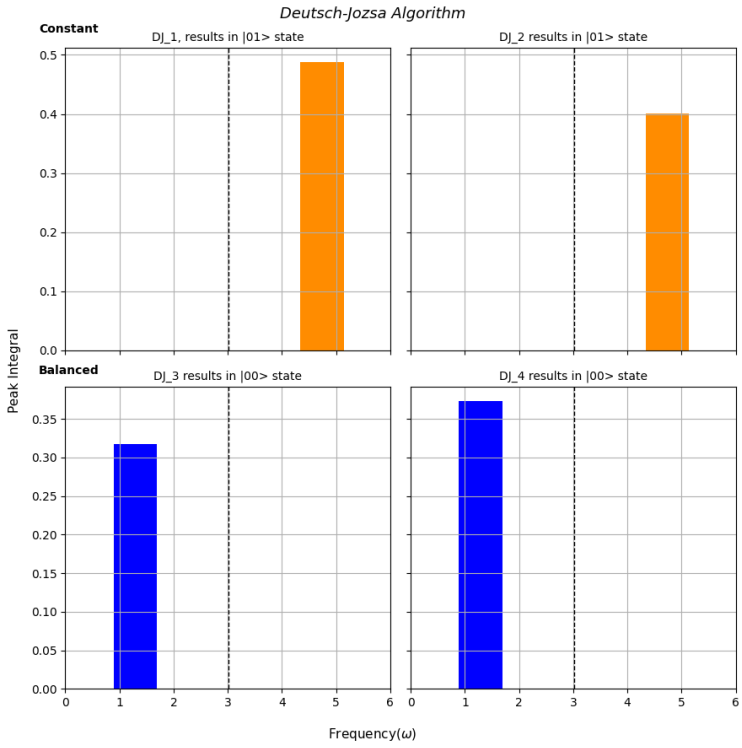


Fig. 11: The constant functions result in the same spectrum as expected in the $|10\rangle$ and the balanced functions result in the same spectrum but in the state $|00\rangle$

As mentioned earlier, the pulse sequence used to construct the CNOT gate contains uncorrected relative phases that flip the second qubit at the end of the balanced oracle instead of the first qubit taking it to the $00\rangle$ state instead. This may be due to the CNOT gates placement between two Hadamard gates where the phase accumulates due to the superposition.

Table II: Deutsch-Jozsa Algorithm Fidelities

Oracle	Fidelities
DJ_1	0.8256
DJ_2	0.6781
DJ_3	0.5083
DJ_4	0.5970

The fidelities once again compare the empirical peak integrals with the expected peak integrals. The fidelity is noticeably poor for the Balanced function because the exact CNOT gate used instead of the approximate CNOT gate. The choice was made when the simulated matrix multiplication could not provide pure state for the algorithm when the approximate CNOT gate was used and was hence ruled out and not tested experimentally.

The algorithm could be improved by using the approximate CNOT gate for the balanced functions reducing circuit depth and reducing pulse sequence execution time to be below the short decoherence time for the Carbon spin. Additionally, extra pulse amplitude calibration experiments can be conducted to find the correct amplitude to allow for accurate simultaneous pulses for the Hadamard gate.

IV.E. Grover's Algorithm

Grover's search algorithm is initialized to the $00\rangle$ and will yield a probabilistic outcome of the marked state depending on the oracle. In the two qubit system, the probability is already very high but the amplitude of algorithm can be improved by more iterations of the total Grover pulse at the period of 3.

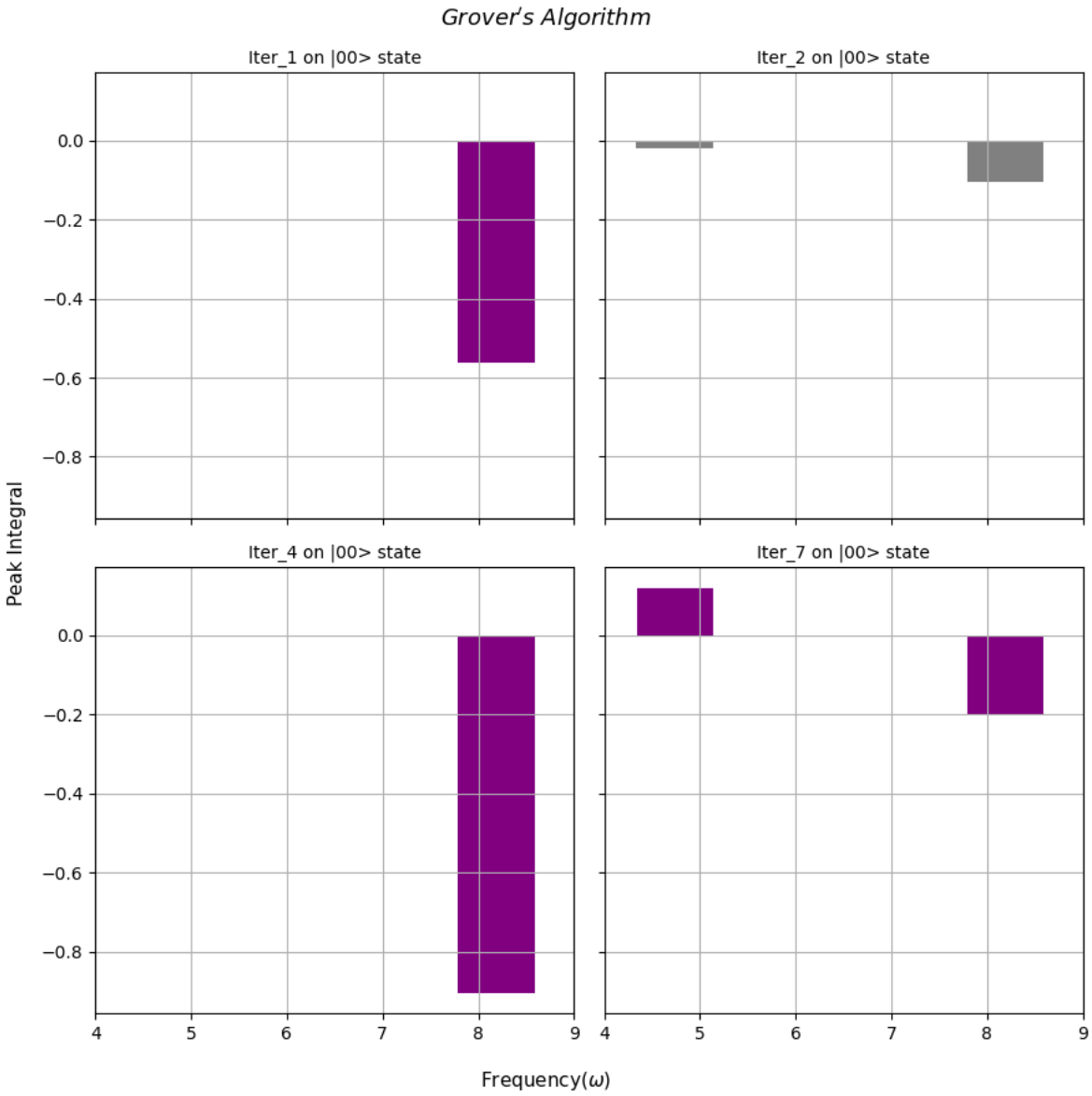


Fig. 12: The cyclical nature of Grover's algorithm can be seen as the peak integral improves from 1 Iteration to 4 Iterations and the plot with 2 Iterations shows that a non-periodic number of iterations does not yield the desired state let alone improve amplitude

Table III: Grover's Algorithm Fidelities

Iterations	Performance
For 1	× 1.3503
For 4	× 2.1782
For 7	× 0.2907

The performance refers to how much larger or smaller peak integrals are from the Grover's algorithm compared to the expeced peak integral for the regular $|11\rangle$ state. 1 iteration improves the peak and 4 iterations improves the peak integral even further, but at 7 Iterations, the total pulse sequence duration is longer than the coherence time of the Carbon spins so the resulting spectrum is poorer.

V. CONCLUSION

The Exact CNOT gate can be successfully implemented on the NMR system with high fidelity compared to the Approximate CNOT gate. The most accurate depiction of the working of the gate would be via overcomplete tomography to show the change in the populations of the density matrix before and after the application of the gate. This necessary as the peak integral data from the Carbon channel is crucial to properly characterize the resultant states. This process in addition to the comparision of peak integral ratios between the two channels can better quantify the fidelity of the gate.

By extension, the Deutsch-Jozsa algorithm also sucessfully displays the correct states expected for the constant and balanced functions but the performance of the algorithm can be improved by using the approximate CNOT gate and using pulse amplitude calibration to obtain the correct pulse amplitude for Hydrogen when pulsing on both channels at the same time.

Similarly, Grover's algorithm shows that for the chosen oracle, the correct state has been obtained at the amplitude increases for periodic iterations and fails when the number of iterations is not periodic. A more comprehensive way method to show the cyclical nature is to obtain the peak integrals of the expected state for a range of iterations between 1 to 7. Beyond seven iterations, the pulse sequence will be too lengthy and will always result in noise.

REFERENCES

[a)]Physics and Astronomy Department, UCLA

[1]Isaac L. Chuang, Neil Gershenfeld, and Mark Kubinec. Experimental implementation of fast quantum searching. *Phys. Rev. Lett.*, 80:3408–3411, Apr 1998.

[2]David Cory, Amr Fahmy, and Timothy Havel. Ensemble quantum computing by nmr spectroscopy. *Proceedings of the National Academy of Sciences of the United States of America*, 94:1634–9, 04 1997.

[3]Kavita Dorai, T S Mahesh, Arvind Arvind, and Anil Kumar. Quantum computation using nmr. *Current Science*, 79, 11 2000.

[4]Jonathan A. Jones. Quantum computing with nmr. *Progress in Nuclear Magnetic Resonance Spectroscopy*, 59(2):91–120, August 2011.

[5]E. Knill, I. Chuang, and R. Laflamme. Effective pure states for bulk quantum computation. *Physical Review A*, 57(5):3348–3363, May 1998.

[6]Dawei Lu, Aharon Brodutch, Jihyun Park, Hemant Katiyar, Tomas Jochym-O'Connor, and Raymond Laflamme. Nmr quantum information processing, 2015.

Appendix A: Pulse Program Sequences

1. Hadamard Gate

```
pulse(1, a90HC, 0, freq1H, 2, a90C, 0, freq13C, d180C)
delay(0.25) #(IX+SX)(pi)}
pulse(1, a90HC, 1, freq1H, 2, a90C, 1, freq13C, d90C)}
delay(0.25) #(IY+SY)(pi/2)}
#Readout Pulse
pulse(1, a90H, 0, d90H)
```

2. Permutations

a. P1

```
pulse(1, a90H, 0, d90H)
delay(dEvolution)
pulse(1, a90HC, 1, freq1H, 2, a90C, 0, freq13C, d90C)
delay(dEvolution)
pulse(2, a90C, 1, d90C)
```

```
delay(0.25)
#Readout Pulse
pulse(1, a90H, 0, d90H)
```

b. P2

```
pulse(2, a90C, 0, d90C)
delay(dEvolution)
pulse(1, a90HC, 0, freq1H, 2, a90C, 1, freq13C, d90C)
delay(dEvolution)
pulse(1, a90H, 1, d90H)
delay(0.25)
#Readout Pulse
pulse(1, a90H, 0, d90H)
```

3. Approximate CNOT

```
pulse(2, a90C, 0, d90C) #X-Rotation
delay(dEvolution) #ZZ-Coupled term Rotation}
pulse(2, a90C, 1, d90C) #Y-Rotation
delay(0.25)}
#Readout Pulse
pulse(1, a90H, 0, d90H)
```

4. CNOT

```
pulse(2, a90C, 2, d90C) #SZ Composite Pulse
delay(0.25)
pulse(2, a90C, 3, d90C)
delay(0.25)
pulse(2, a90C, 0, d90C)
delay(0.25)
pulse(2, a90C, 0, d90C) #X-Rotation
delay(dEvolution)       #ZZ-Coupled term Rotation}
pulse(2, a90C, 1, d90C) #Y-Rotation
delay(0.25)
pulse(1, a90H, 2, d90H) #IZ Composite Pulse}
delay(0.25)
pulse(1, a90H, 1, d90H)
delay(0.25)
pulse(1, a90H, 0, d90H)
delay(0.25)

#Readout Pulse
pulse(1, a90H, 0, d90H)
```

5. Deutsch-Jozsa Oracles

a. Constant Function U_2

```
pulse(2, a90C, 0, d180C)
delay(0.25) #SX (pi)
```

b. Balanced Function U_3

```
pulse(2, a90C, 2, d90C)#SZ Rotation
delay(0.25)
pulse(2, a90C, 3, d90C)
delay(0.25)
pulse(2, a90C, 0, d90C)
delay(0.25)
pulse(2, a90C, 0, d90C)#X-Rotation
delay(dEvolution)      #ZZ-Coupled term Rotation
pulse(2, a90C, 1, d90C)#Y-Rotation
delay(0.25)}
pulse(1, a90H, 2, d90H)#IZ Rotation
delay(0.25)
pulse(1, a90H, 1, d90H)
delay(0.25)
pulse(1, a90H, 0, d90H)
delay(0.25)
```

c. Balanced Function U_4

```
pulse(2, a90C, 2, d90C)#SZ Rotation
delay(0.25)
pulse(2, a90C, 3, d90C)
delay(0.25)
pulse(2, a90C, 0, d90C)
delay(0.25)
pulse(2, a90C, 0, d90C)#X-Rotation
delay(dEvolution)      #ZZ-Coupled term Rotation
pulse(2, a90C, 1, d90C)#Y-Rotation
delay(0.25)}
pulse(1, a90H, 2, d90H)#IZ Rotation
delay(0.25)
pulse(1, a90H, 1, d90H)
delay(0.25)
pulse(1, a90H, 0, d90H)
delay(0.25)
pulse(2, a90C, 0, d180C)#SX (pi)
delay(0.25)
```

6. Grover's Oracles

a. P

```
pulse(1, a90H, 0, d90H) #IZ Composite Pulse
delay(0.25)
pulse(1, a90H, 1, d90H)
delay(0.25)
pulse(1, a90H, 2, d90H)
delay(0.25)
pulse(2, a90C, 0, d90C) #SZ Composite Pulse
delay(0.25)
pulse(2, a90C, 1, d90C)
delay(0.25)
pulse(2, a90C, 2, d90C)
delay(dEvolution)        #ZZ-Coupled term Rotation
```

b. Z_3

```
pulse(1, a90H, 1, d90H) #IZ Composite Pulse
delay(0.25)
pulse(1, a90H, 0, d90H)
delay(0.25)
pulse(1, a90H, 3, d90H)
delay(0.25)
pulse(2, a90C, 1, d90C) #SZ Composite Pulse
delay(0.25)
pulse(2, a90C, 0, d90C)
delay(0.25)
pulse(2, a90C, 3, d90C)
delay(dEvolution)       #ZZ-Coupled term Rotation
```

7. Z_2

```
pulse(1, a90H, 2, d90H) #IZ Composite Pulse
delay(0.25)
pulse(1, a90H, 1, d90H)
delay(0.25)
pulse(1, a90H, 3, d90H)
delay(0.25)
pulse(2, a90C, 2, d90C) #SZ Composite Pulse
delay(0.25)
pulse(2, a90C, 0, d90C)
delay(0.25)
pulse(2, a90C, 3, d90C)
delay(dEvolution)       #ZZ-Coupled term Rotation
```

8. Z_1

```
pulse(1, a90H, 2, d90H) #IZ Composite Pulse
delay(0.25)
pulse(1, a90H, 0, d90H)
delay(0.25)
pulse(1, a90H, 3, d90H)
delay(0.25)
pulse(2, a90C, 2, d90C) #SZ Composite Pulse
delay(0.25)
pulse(2, a90C, 1, d90C)
delay(0.25)
pulse(2, a90C, 3, d90C)
delay(dEvolution)       #ZZ-Coupled term Rotation
```