October 22, 2015

Notes on the 12 qubit PPS

Dawei Lu

Notes about the problems in the 12 qubit PPS preparation, including Matlab codes and Experiments.

Dec 12, 2014:

I tried state to state GRAPE pulses. The one to evolve ZZZZZZZIIIII to ZZZZZZZZZZZZ cannot be found, but the C-H SWAP gate can.

Calculating the state to state GRAPE on Ordi2. In pulsefinder folder. paramsfile is 'twqubit_subS2S.m', and the output file is 'twqubit_7zto12z'.

The GRAPE is to evolve ZZZZZZZIIIII to ZZZZZZZZZZZZ. As the couplings between nearest-neighbored C and H are about 150Hz. I set the GRAPE as 4ms and 400 steps, with Zfreedom allowed.

The fidelity keeps 0 all the time. Guess the reason is 'Zfreedom'. Set 'params.Zfreedomflag = 0;'. However, still 0.

Annie said maybe due to the length. Her SWAP gate requires 8ms, so I changed 'params.plength = 800;'. But for with or without Zfreedom, fidelity is still 0.

Check if some of my GRAPE settings are wrong. try to repeat Annie's SWAP gate calculation.

```
% Number of timesteps
params.plength = 800;

% Length of each time step
params.timestep = 10e-6;

params.subsystem{1} = [1 2 3 9 10 11];
params.subsystem{2} = [4 5 6 7 8 12];
params.subsys_weight = [6 6];

% Input and goal states for state to state
params.rhoin = mkstate('+1IIIIIIIZIIII+1IIIIIIIIZIII+1IIIIIIIIIZII+1IIIIIIIIIIZI
    +1IIIIIIIIIIZ',1);
params.rhogoal = mkstate('+1IIIIIIIZIIII+1IIIIIIIIZIII+1IIIIIIIIIZII+1
    IIIIIIIIIIZI+1IIIIIIZIIIIZ',1);

% Allow Zfreedom or not
params.Zfreedomflag = 0;
```

The outputfile is 'twqubit_SWAPC7H5'. And the fidelity is already over 98%. Then I changed 'params.Zfreedomflag = 1;', and the fidelity is over 95% after 30 iterations. Much slower than the no Zfreedom case. Maybe due to different initial guesses.

Dec 15, 2014 to Mar 03, 2015:

All single qubit $\pi/2$ and π have been generated by the sub-system method. The fidelity and unitary operator of each pulse in the whole system is also calculated.

Generate all $\pi/2$ and π pulses for the 7 Carbons, with the Calibration = 25KHz. $\pi/2$ pulses are 1ms length and 100 steps, and π pulses are 2ms length and 200 steps. Generating Code in 'twqubit_shape.m'

```
for ii = 1:7
loadfile = ['twqubit_C', num2str(ii), '180', '.mat'];
eval(['load ', loadfile]);
filename1 = ['twqubit_C', num2str(ii), '180_C_25000.txt'];
filename2 = ['twqubit_C', num2str(ii), '180_H_25000.txt'];
make_bruker_shape(pulses{1}, 25000, filename1,1);
make_bruker_shape(pulses{1}, 25000, filename2,2);
end
```

To generate π pulses, In the beginning, I wanted to combine two $\pi/2$ pulses. A potential problem is when calculating the GRAPE, we have considered the 4us free evolutions in the beginning and in the end. If we combine, we will have an unwanted 8us free evolution in the middle of the new π pulse. So I used 'combine90to180' to check the π pulse fidelity. They are very bad actually. All of them are just 0.75 0.76 in fidelity. However, we can still use them as the initial guess to search the optimal π pulse. By using them, the convergence speed is very fast, which means the initial guess is indeed very important in 12 qubits.

The pulses are saved in Ordi2 '\pulsefinder\12 Qubit\' with the names such as 'twqubit_C590_C_25000.txt'. After that, I checked all the fidelities of the pulses in the folder '\pulseexam_12qubit\C_rotations\check_grape.m'. The code is in Appendix I. To my surprise, all fidelities in the 12-qubit system are still over 0.99, which means the separation of the two sub-systems are pretty well.

Unitaries and Fidelities of the pulses will both be saved in 'twqubit_C590_Ufid.mat', so they can be called for further calculations in the PPS simulation. The folder is '\pulseexam_12qubit\C_rotations\'. Use 'check_power.m' to check the maximal powers for C and H channel to guarantee all powers are within the power limit. For details, see the Table below**??**.

Rotation	Length	Fidelity	File	MaxPower C	MaxPower H
$R_x^1(\pi/2)$	1ms	0.9981	twqubit_C190_Ufid.mat	56.0%, 14000Hz	22.3%, 5557Hz
$R_x^2(\pi/2)$	1ms	0.9986	twqubit_C290_Ufid.mat	41.7%, 10422Hz	23.5%, 5878Hz
$R_x^3(\pi/2)$	1ms	0.9981	twqubit_C390_Ufid.mat	31.9%, 7979.0Hz	22.3%, 5568Hz
$R_x^4(\pi/2)$	1ms	0.9976	twqubit_C490_Ufid.mat	31.6%, 7892.0Hz	23.8%, 5954Hz
$R_x^5(\pi/2)$	1ms	0.9981	twqubit_C590_Ufid.mat	56.1%, 14033Hz	30.7%, 7678Hz
$R_x^6(\pi/2)$	1ms	0.9979	twqubit_C690_Ufid.mat	57.3%, 14333Hz	34.4%, 8595Hz
$R_x^7(\pi/2)$	1ms	0.9986	twqubit_C790_Ufid.mat	43.7%, 10925Hz	24.8%, 6207Hz
$R_x^1(\pi)$	2ms	0.9976	twqubit_C1180_Ufid.mat	62.6%, 15655Hz	34.9%, 8726Hz
$R_x^2(\pi)$	2ms	0.9980	twqubit_C2180_Ufid.mat	51.1%, 12783Hz	32.4%, 8094Hz
$R_x^3(\pi)$	2ms	0.9975	twqubit_C3180_Ufid.mat	37.4%, 9350.0Hz	24.0%, 5997Hz
$R_x^4(\pi)$	2ms	0.9970	twqubit_C4180_Ufid.mat	45.1%, 11268Hz	20.4%, 5108Hz
$R_x^5(\pi)$	2ms	0.9975	twqubit_C5180_Ufid.mat	67.6%, 16895Hz	31.1%, 7782Hz
$R_x^6(\pi)$	2ms	0.9976	twqubit_C6180_Ufid.mat	71.8%, 17948Hz	33.6%, 8396Hz
$R_x^7(\pi)$	2ms	0.9977	twqubit_C7180_Ufid.mat	51.0%, 12759Hz	32.1%, 8022Hz

Dec 22, 2014 to Feb 06, 2015:

Got all the 5 GRAPE pulses for encoding. The folder is '\pulseexam_12qubit\C_rotations\'. Here is the Table listing all the details of the 5 encoding pulses.

Rotation	Length	Fidelity	File	MaxPower C	MaxPower H
$R_x^{5,7}(\pi)$	2ms	0.9980	twqubit_C57180_Ufid.mat	32.3%, 8072.5Hz	24.2%, 6049Hz
$R_x^{2,3}(\pi)$	2ms	0.9978	twqubit_C23180_Ufid.mat	32.4%, 8101.5Hz	22.8%, 5701Hz
$R_x^{2,3,4,7}(\pi/2)$	1ms	0.9970	twqubit_C234790_Ufid.mat	37.4%, 9358.3Hz	28.9%, 7213Hz
$R_x^{1,5,6}(\pi)$	2ms	0.9974	twqubit_C156180_Ufid.mat	32.2%, 8039.7Hz	20.3%, 5086Hz
$R_x^{2,4,7}(\pi/2)R_{-y}^3(\pi/2)R_{-z}^{i=2,3,4,7}((w_i - O_1) * 3.36\text{ms})$	1ms	0.9964	twqubit_C234790withPC_Ufid.mat	26.1%, 6514.5Hz	20.2%, 5048Hz

The last pulse for the Encoding is (From the PPS.m file in folder 'Twqubit')

```
%Phase Correction
U7 = R(gop(2,X),90)*R(gop(2,-Z),360*(Para(2,2)-20696)*1/2/148.5)*...
R(gop(3,-Y),90)*R(gop(3,-Z),360*(Para(3,3)-20696)*1/2/148.5)*...
R(gop(4,X),90)*R(gop(4,-Z),360*(Para(4,4)-20696)*1/2/148.5)*...
R(gop(7,X),90)*R(gop(7,-Z),360*(Para(7,7)-20696)*1/2/148.5);
```

Feb 18, 2014:

Got two pulses for phase cycling and polarization crush. One is 'twqubit_all90.m' which is used for phase cycling, and the other one is 'twqubit_all90butC7.m' used for the polarization crush in the beginning.

Rotation	Length	Fidelity	File	MaxPower C	MaxPower H
$R_x^{1-12}(\pi/2)$	1ms	0.9977	twqubit_all90_Ufid.mat	27.8%, 6956.6Hz	30.4%, 7594Hz
$R_x^{1-6,8-12}(\pi/2)$	1ms	0.9977	twqubit_all90butC7_Ufid.mat	24.5%, 6134.9Hz	25.0%, 6239Hz

Mar 03, 2014:

All 6 pulses for the decoding part have been checked.

Rotation	Length	Fidelity	File	MaxPower C	MaxPower H
$R_x^{2,3,4,7-12}(\pi)$	2ms	0.9988	twqubit_C2347andH180_Ufid.mat	61.6%, 15400Hz	52.2%, 13039Hz
$R_x^{1,3,4,6}(\pi/2)R_{-y}^{8-12}(\pi/2)$	1ms	0.9974	twqubit_C134690andH90_Ufid.mat	24.8%, 6203.2Hz	22.1%, 5529Hz
$R_x^{2,3,4,5,6}(\pi)$	2ms	0.9984	twqubit_C23456180_Ufid.mat	37.8%, 9438.2Hz	23.0%, 5746Hz
$R_{-y}^{4,6}R_y^{1,3}(\pi/2)R_x^2(\pi/2)R_{-z}^1(6.6\text{ms})$	1ms	0.9982	twqubit_C1234690withPC_Ufid.mat	28.3%, 7070.8Hz	26.9%, 6717Hz
$R_x^{2,7}(\pi)$	2ms	0.9979	twqubit_C27180_Ufid.mat	29.1%, 7285.3Hz	21.7%, 5414Hz
$R_y^2(\pi/2)R_x^5(\pi/2)$	1ms	0.9975	twqubit_C2Y5X90_Ufid.mat	28.9%, 7233.9Hz	29.2%, 7292Hz

MAY 09, 2015 TO MAY 10, 2015: RECALCULATION OF GRAPE PULSES WITH 0US BUFFER

Mar 09, 2015:

The reason of this recalculation is described here.

According to the fact that 4us cannot be absorbed into a 10us shape pulse, I decided to delete the buffer delay in the GRAPE calculation. An AU program called 'GRAPE_recal_nobuffer.m' was written to do it. The directory in Ordi2 is '\pulseexam_12qubit\C_rotations\GRAPE_nobuffer\' or in the SVN '\Matlab\Twqubit\Pulse_NoBuffer\'.

There are 18 pulses required for recalculation with 8 $\pi/2$ rotations and 10 π rotations. In the program, I first regenerate Bx and By from the shaped pulses, and use them as the initial guess, saved in files titled like 'twqubit_C190_InitialGuess.mat'. Then generate the params setting file with 0us buffer. **Note the power limit of C channel is set as 12.5KHz instead of 25KHz.** Both the initial guess file and params setting file are transferred to Feynman and Ordi2 by the scp command. A cluster sample is also written and saved in Ordi2 pulsefinder folder.

In calculation on Feynman in the weekend. All the 18 pulses

```
Pulse_Name{1} = 'twqubit_C790';
Pulse_Name{2} = 'twqubit_C290';
Pulse_Name{3} = 'twqubit_C234790';
Pulse_Name{4} = 'twqubit_C234790withPC';
Pulse_Name{5} = 'twqubit_C134690andH90';
Pulse_Name{6} = 'twqubit_C1234690withPC';
Pulse_Name{7} = 'twqubit_C2Y5X90';
Pulse_Name{8} = 'twqubit_C590';
Pulse_Name{9} = 'twqubit_C2180';
Pulse_Name{10} = 'twqubit_C6180';
Pulse_Name{11} = 'twqubit_C4180';
Pulse_Name{12} = 'twqubit_C57180';
Pulse_Name{13} = 'twqubit_C1180';
Pulse_Name{14} = 'twqubit_C23180';
Pulse_Name{15} = 'twqubit_C156180';
Pulse_Name{16} = 'twqubit_C2347andH180';
Pulse_Name{17} = 'twqubit_C23456180';
Pulse_Name{18} = 'twqubit_C27180';
```

Got almost all pulses with fidelity over 0.995. The exception is pulse Number 10, C6180. The reason should be the power limit 12.5KHz is not enough. So change it to 25KHz again and recalculate this GRAPE. All the GRAPE .mat files are transferred to '\Matlab\Twqubit\Pulse_NoBuffer\'.

An AU program 'make_shape_12qubit.m' for making the Bruker shape is used to generate all Burker files. Then the maximal power is checked using 'check_power_NoBuffer.m'. The powers are almost the same as the ones from the initial guess, and **are updated everywhere in this draft**.

DEC 16, 2014 TO FEB 12, 2015: COMBINING GRAPE PULSES TO BIG SHAPE FILES

Dec 16, 2014:

Combine pulses in the PPS preparation into big shape files, which should be easy for calibrations and pulsefixing. **Note later I fixed each pulse independently and combine the fixed ones together, as the optimal fix number for each pulse can be different.**

The code is in the SVN server for Matlab named '\Twqubit\pulse_combine.m'.

First read all the powers and phases for the $\pi/2$ and π rotations. The code is in Appendix I.

Then combine them with the free evolutions. Here I set the time step dt = 10us. The code is in Appendix I.

The two output files are 'twqubit_encoding1_C' and 'twqubit_encoding1_H'. The calibrations are 25000Hz.

Feb 12, 2015:

Solved the problem that how to change the phase of GRAPE pulses to other phases. For example, if we already have a X rotation, how can we rewrite the phase column to make it realize a Y rotation.

When combining all pulses into a large shape file, one has to know how to change a shape for X rotation to Y rotation. It should be a $\pi/2$ phase difference for every segment in the shape. I am going to check it.

The checking uses 4-qubit Crotonic in the folder 'F:\matlab\pulseexam_7qubit\4 qubit pulse check'. The target unitary is $R_x^1(\pi/2)$, and the pulse is 'Croton_90x1.txt' with length 1ms, 500 segments and amplitude 6000Hz.

When compared with $R_x^1(\pi/2)$, the fidelity is 0.9996. Then I changed the target to $R_y^1(\pi/2)$ with the GRAPE pulse unchanged. The fidelity goes to 0.4998 which is reasonable.

In order to produce a $R_y^1(\pi/2)$ from the original X rotation pulse, I added 90 to all phases in all segments. However, the fidelity goes to almost 0. Again all phases are reduced by 90, and this time the fidelity is 0.9996, which is what we want!

Conclusion: If you want to realize a Y rotation based on a X rotation pulse, just change the phase to phase-90 in each segment, and mod by 360 for the spectrometer.

```
phase = phase - 90;
phase = mod(phase, 360);
```

I wrote a program 'grape_phase' to generate the new phase in the folder 'F:\matlab\pulseexam_7qubit\'. Used in this manner 'phase_new = grape_phase(phase, initial_phase, end_phase)', where the initial_phase and end_phase can be X, Y, -X, or -Y.

The way to get the new operator is through the equation $R_z(\theta) = XR_y(\theta)\bar{X}$. If you know the unitary U_x of the X rotation pulse, and when you are realizing Y pulse from that one, the new unitary U_y is thus

$$U_y = R_z(\pi/2)U_xR_z(-\pi/2); \tag{1}$$

Generated the first encoding part, which will evolve Z7 to Z24567. The files 'twqubit_encoding1_C' and 'twqubit_encoding1_H' are in Ordi2 '\pulseexam_12qubit'. The total length is 32.98ms. Next I have to check whether the final state after this pulse will be Z24567 or not. So 'check_encoding.m' is written. The directory is Ordi2 '\pulseexam_12qubit'. This function will load all necessary .mat files to get the unitaries and calculate the final state based on these unitaries from Z7.

The final fidelity is 0.9832 (the same for with or without gradient) for Z24567. Two files 'U_encoding1.mat' and 'rho_encoding1.mat'. Now go on to the second Encoding part.

The fidelity for the second Encoding part is -0.9692 (the same for with or without gradient) for Z1234567. Two files 'twqubit_encoding2_C' and 'twqubit_encoding2_H' are in Ordi2 '\pulseexam_12qubit'. The total length is 21.29ms. Then the last piece in Encoding!

The fidelity for the third Encoding part is -0.9160 (the same for with or without gradient) for Z123456789101112. Two files 'twqubit_encoding3_C' and 'twqubit_encoding3_H' are in Ordi2 '\pulseexam_12qubit'. The total length is 7.36ms.

Update on Mar 04: In the last code, the free evolution time is not the integer times of 10us, but it should be. So the function 'F.m' is modified as

```
function F=F(Hamiltonian, time)
F=diag(exp(diag(-i*Hamiltonian*10e-6*round(time/10e-6))),0);
```

And all the fidelities are recalculated. They are 0.9831 for Encoding 1, -0.9717 for Encoding 2, -0.9124 for Encoding 3.

MAR 04, 2015: RE-CHECK AND SIMPLIFIED THE PPS CIRCUIT IN SIMULATION

I found the decoding part can be simplified. Check it in Matlab.

All the files are saved in Ordi2 '\twqubit\'. For the encoding which consists of three parts, the fidelities have been checked before.

TABLE I: Encoding in 12-qubit PPS

Target State	State Fidelity	Density Matrix	Unitary Operator
IZIZZZXIIIII	1.0000	rho_encoding1.mat	U_encoding1.mat
ZXZZZZZIIIII	-1.0000	rho_encoding2.mat	U_encoding2.mat
ZZZZZZZZZZZZ	-0.9500	rho_encoding3.mat	U_encoding3.mat

For the phase cycling it has been done too.

TABLE II: Phase Cycling in 12-qubit PPS

Target State	State Fidelity	Density Matrix	Unitary Operator
$\|00...0\rangle\langle 00...0\| + \|11...1\rangle\langle 11...1\|$	0.9511	rho_phasecycling.mat	NAN

For the decoding part, the simulation is as follows.

TABLE III: Decoding in 12-qubit PPS

Target State	State Fidelity	Density Matrix	Unitary Operator
$A_1A_5A_6\|+---++-00000\rangle+$ $A_1'A_5'A_6'\|-+++--+00000\rangle$	0.8717	rho_decoding1.mat	U_decoding1.mat
$A_1A_5A_6A_7\|0+00-0-00000\rangle+$ $A_1'A_5'A_6'A_7'\|0-00+0+00000\rangle$	0.8570	rho_decoding2.mat	U_decoding2.mat
$A_1A_5A_6A_7A_5^{new}\|000000-00000\rangle+$ $A_1'A_5'A_6'A_7'A_5^{new'}\|000000+00000\rangle$	0.8570	rho_decoding3.mat	U_decoding3.mat

A_1, A_5 and A_6 are phases produced by the chemical shift evolutions of C1, C5 and C6 during $1/2\mathrm{J}_{\mathrm{C7H5}}$. A_7 is the phase by the chemical shift evolution of C7 (**Note: C7 is x-iy so the phase is the conjugate**) during $1/2\mathrm{J}_{\mathrm{C2C3}}$, and the coupling evolutions between C7 and all protons. A_5^{new} is the phase of C5 again in $1/2\mathrm{J}_{27}$, including the chemical shift evolution and coupling evolutions (J25 and J57 are switched off). So the phases are

$$\begin{aligned}
A_1 &= cos(2\pi(\omega_1-O_1)/2\mathrm{J}_{\mathrm{C7H5}}) - isin(2\pi(\omega_1-O_1)/2\mathrm{J}_{\mathrm{C7H5}}),\\
A_5 &= cos(2\pi(\omega_5-O_1)/2\mathrm{J}_{\mathrm{C7H5}}) - isin(2\pi(\omega_5-O_1)/2\mathrm{J}_{\mathrm{C7H5}}),\\
A_6 &= cos(2\pi(\omega_6-O_1)/2\mathrm{J}_{\mathrm{C7H5}}) - isin(2\pi(\omega_6-O_1)/2\mathrm{J}_{\mathrm{C7H5}}),\\
A_7 &= cos(2\pi(\omega_7-O_1)/2\mathrm{J}_{23}) + isin(2\pi(\omega_7-O_1)/2\mathrm{J}_{23}) *\\
&\prod_{k=8}^{12}(cos(\pi\mathrm{J}_{7k}/\mathrm{J}_{23}) + isin(\pi\mathrm{J}_{7k}/\mathrm{J}_{23})),\\
A_5^{new} &= cos(2\pi(\omega_5-O_1)/2\mathrm{J}_{27}) + isin(2\pi(\omega_5-O_1)/2\mathrm{J}_{27}) *\\
&\prod_{k\neq 2,5,7}(cos(\pi\mathrm{J}_{5k}/\mathrm{J}_{27}) + isin(\pi\mathrm{J}_{5k}/\mathrm{J}_{27}))
\end{aligned} \tag{2}$$

When calculating the evolutions in the Decoding part, using the following equations (the evolution is 1/2J with two π

pulses inserted in the middle. **Note: without π in the end**)

$$
\begin{aligned}
(X+iY)\otimes(X+iZ) &\longrightarrow (X-iY)\otimes(I+Z),\\
(X-iY)\otimes(X+iZ) &\longrightarrow (X+iY)\otimes(I-Z),\\
(X+iY)\otimes(X-iZ) &\longrightarrow (X+iY)\otimes(I-Z),\\
(X-iY)\otimes(X-iZ) &\longrightarrow (X+iY)\otimes(I+Z).
\end{aligned} \tag{3}
$$

MAR 10, 2015 TO MAR 11, 2015: SIMULATED SPECTRA FOR THE 12 QUBIT PPS CIRCUIT AND WITH GRAPE PULSES

Mar 10, 2015:

Assume all pulses are perfect and simulated the PPS circuit for 12 qubits. The imperfection comes from the simplification of the circuit.

The final state 'rho_decoding3.mat' includes a phase on C7. 'rho_decoding3(1,33) = 0.1979 + 0.5728i', and I used a Z rotation $R_z^7(\theta)$ to rotate it into X. The rotating angle θ is 1.2381, and the element becomes 0.6060 after the rotation. Note 0.6060/0.7071 (the ideal value) = 0.8570 which is exactly the fidelity for the final state 'rho_decoding3'. The new state is named 'rho_12pps_circuit', and is saved in Ordi2 '\twqubit\rho_12pps_circuit.mat'.

The simulation is implemented in Ordi2 '\NMR\Experiments\twqubit'. The main file is 'sim_twqubit.m'.

Some settings: o1_C = 20696, o1_H = 2894, td = 281684, swh = 30030.

The decoherence time is set in '\SRC\simulator\spectrumfast.m' and the value for C7 is chosen as 450ms.

1) For thermal, the spectrum file is saved in 'F:\matlab_full\IQC_simulation\PPS 12 qubit\sim_thermal_450ms.mat'.

2) For PPS of the circuit, copy 'rho_12pps_circuit.mat' to the folder '\SRC\simulator\' and change the function 'mkstate.m'. The spectrum file is saved in 'F:\matlab_full\IQC_simulation\PPS 12 qubit\sim_pps_obC7_phasefixed.mat' (phase fixed means the final coherence is on X axis by applying a Z rotation).

The comparison (**Note the PPS data should be divided by 24**) is shown in the following Fig. 1.

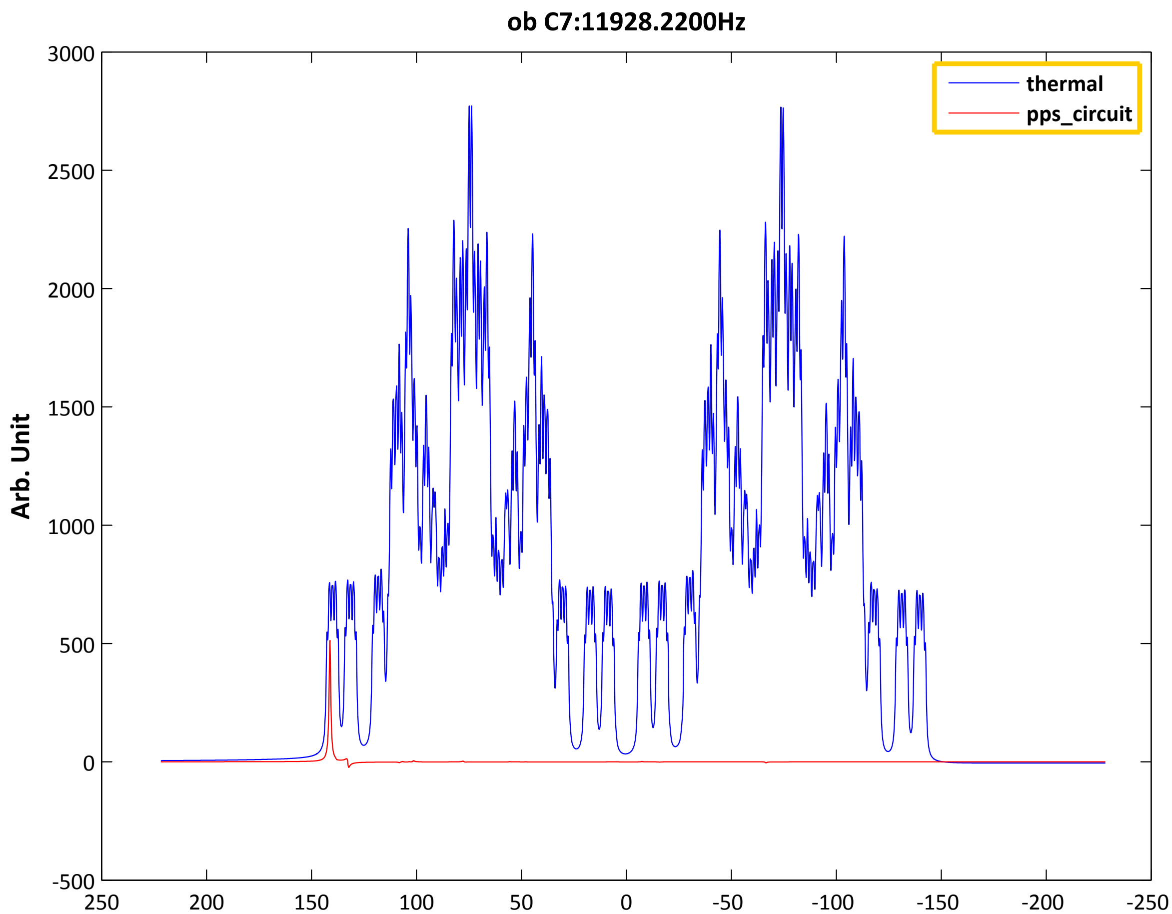

FIG. 1: Comparison of the thermal and PPS produced by the circuit.

Mar 11, 2015:

Use the GRAPE unitaries to simulate the 12-qubit PPS circuit again. The imperfection of the GRAPE pulses will contribute to some minor error in spectra.

The unitaries for all rotations have been saved in Ordi2. So I only combined them with the free evolutions. **Note the time step is 10us, which means the free evolutions are integer times of 10us with the unwanted chemical shift refocusing (important!!!).**

Evolution times for every part.

Encoding1:

Name	Total	$1/4J_{27}$	$1/4J_{67} - 1/4J_{27}$	$1/4J_{47} - 1/4J_{67}$	$1/4J_{57} - 1/4J_{47}$	$1/4J_{57}$
Encoding1	32.98ms	6670us	560us	1380us	2880us	11490us

Encoding2:

Name	Total	$1/4J_{12}$	$1/4J_{23} - 1/4J_{12}$	$1/4J_{23}$
Encoding2	21.28ms	4340us	3300us	7640us

Encoding3:

Name	Total	$1/4J_{CH}$	$1/4J_{CH}$
Encoding3	7.36ms	1680us	1680us

Decoding1:

Name	Total	$1/4J_{CH}$	$1/4J_{CH}$
Decoding1	6.36ms	1680us	1680us

Decoding2:

Name	Total	$1/4J_{12}$	$1/4J_{23} - 1/4J_{12}$	$1/4J_{23}$
Decoding2	20.28ms	4340us	3300us	7640us

Decoding3:

Name	Total	$1/4J_{27}$	$1/4J_{27}$	$1/4J_{57}$	$1/4J_{57}$
Decoding3	42.32ms	6670us	6670us	11490us	11490us

Updated on April 6: Please ignore the following sentences as Annie found a mistake in check_grape.m. The unitary of the GRAPE pulse should be U(4us)*Ugrape*U(4us), but I wrote Ugrape*U(8us) by mistake! This will introduce a serious phase error and trigger the following problem. I have fixed it. *It is surprising that after applying phase cycling, the state will introduce a phase. The ideal state after phase cycling will have two elements 0.7071 at the top off-diagonal positions, but these two numbers change to 0.5018-0.4057i and 0.5018+0.4057i after GRAPE phase cycling pulse. The absolute value is 0.6453, which is reasonable. Moreover, the state in the decoding part will also involve an unexpected phase in every step. I guess the reason is the imperfections of the GRAPE pulses (some of them are pretty low like 80% fidelity), and the 90 or 180 pulse cannot rotate or refocus the state to the desired axis completely. Anyway, in the following table, I will write both fidelities, with and without this phase correction.*

Oct 13, 2015:

Simulated the unitaries of the fixed GRAPE pulses. Not surprisingly, the PPS looks still good.

Since we have to fix all GRAPE pulses in the real experiment, the measured pulses in the lab need to be simulated too. At the first sight, the measured pulse is very close to the ideal pulse after pulse fixing. To be more precise I still followed Ray's suggestions and simulated the circuit with the really measured pulses.

First, all pulses are refixed as the environment may change. Then the main function 'pulsefixer_grape.m' is changed slightly to save the measured pulse after every fixing step. The typical name is for example 'twqubit_C2180_C_25000_NoBufferfix5_measured' in the folder 'H:\Matlab\Twqubit\Pulse_NoBuffer_Refix\'. The total number of refixed pulses is 22, and 'check_grape_measured.m' is used to calculate all of the measured pulses' unitaries and fidelities, with each saving in say 'twqubit_C2180_Ufid_measured.mat'. The fidelity of each pulse is about 99.8%. So it is reasonable to guess that the final PPS will be fine, and the simulation gives good result. See Fig. 3. Do not care about the minus peak as I forgot to adjust it to plus.

All fidelities $\mathrm{tr}(\rho_{\mathrm{ideal}}\rho_{\mathrm{real}})$ calculated by the circuit itself and by the GRAPE pulses.

Part	Length	Circuit Fidelity	GRAPE Fidelity	Fix Fidelity
Encoding1	32.98ms	1.0000	0.9831	0.9833
Encoding2	21.28ms	1.0000	0.9717	0.9706
Encoding3	7.36ms	0.9500	0.9124	0.9111
Phase Cycling	1ms	0.9511	0.9126	0.9110
Decoding1	6.36ms	0.8717	0.8693	0.8682
Decoding2	20.28ms	0.8570	0.8430	0.8387
Decoding3	43.32ms	0.8570	0.8234	0.8137

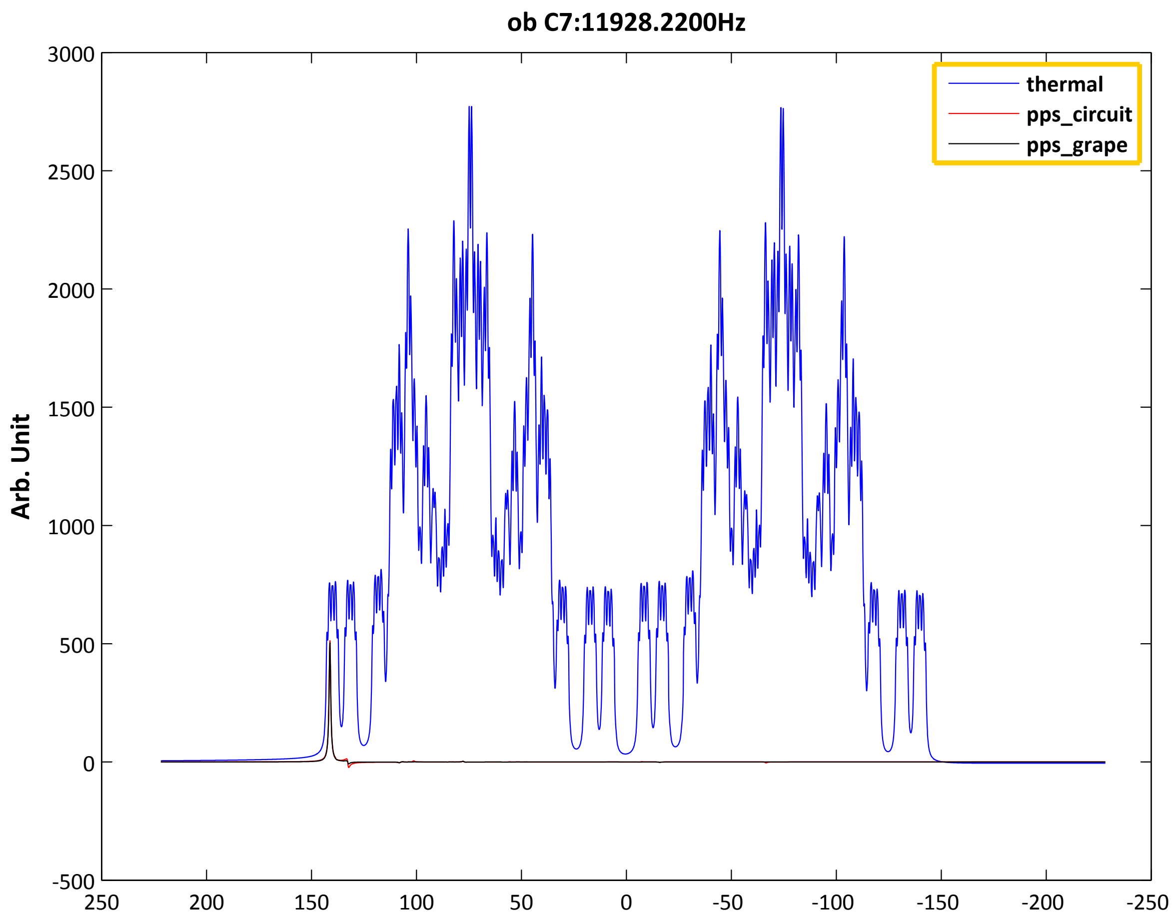

FIG. 2: Comparison of the thermal and PPS produced by the circuit and GRAPE.

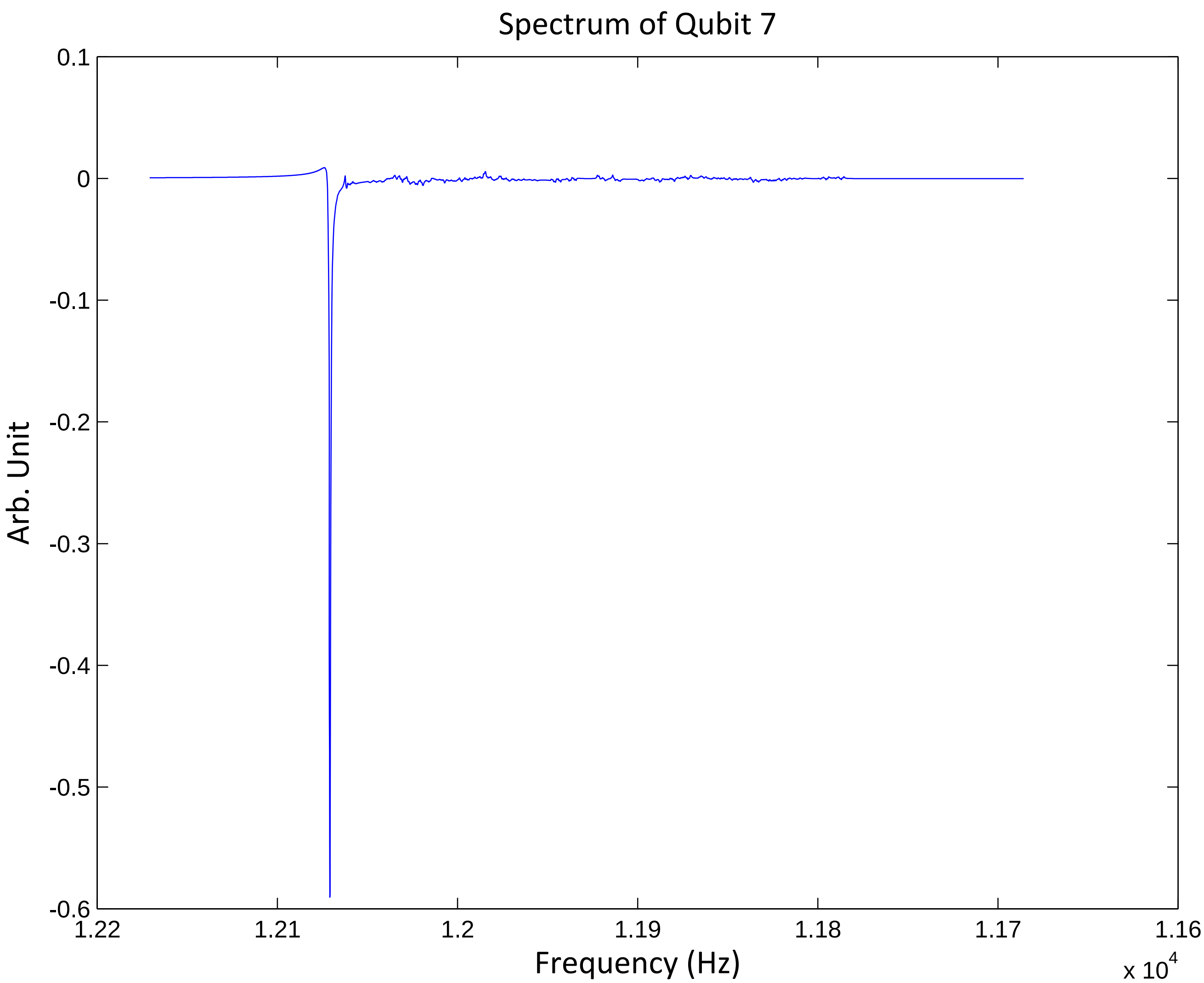

FIG. 3: PPS spectrum of the measured pulses after fixing. The negative direction is meaningless.

JUNE 09, 2015: WRITE A UNIVERSAL PROGRAM FOR SPECTRUM SIMULATION

Fast spectrum generating code spec_plot.m in \subfunction

The previous way to simulate the NMR spectrum by giving the density matrix, Hamiltonian, T2* is via Fourier transform. The code is written by Colm Ryan and it is powerful. However, for 12 qubits it takes more than 40 minutes to get a spectrum. So I write another program to simulate the spectrum. It is universal, and only takes less than 1 minute for a 12-qubit simulation.

The function is 'spec_plot.m' saved in the '\subfunction' folder. There are four input parameters: Hamiltonian, density matrix, observed spins and T2*. By giving the Hamiltonian, we can know the frequencies of all possible peaks. For example, in the 12-qubit sample we would like to observe C7. There should be 2^{11} peaks as 11 spins have interactions with C7. The central frequency is $\omega(7)$, which is the chemical shift of C7. So the frequency of all peaks would be

$$\omega(7) + \sum_{i=1, i\neq 7}^{12} \pm J_{i7}/2. \tag{4}$$

By the signs of J, we can also know the state of the other qubits. I define +J/2 is $|0\rangle$ and -J/2 is $|1\rangle$ to make the $|00..0\rangle$ appear at the leftmost side in Bruker (highest frequency). After confirming the state of the other qubits, we can convert this state to decimal number, and then know the locations of it in the density matrix, because the density matrix can be understood through states (row 1 is $|00..0\rangle$, row 2 is $|00..1\rangle$ etc.). The element in that position gives the real part and imaginary part of the related peak. Then a Lorentzian shape is plotted with this element as the amplitude and phase, and T2* as the half-height width. Sum over all the 2^{11} peaks will give the spectrum of C7. And if you want to save the figure and data points, just input a name after running the code. If you input 0, no data will be saved but the spectrum will be shown in a separate window.

C2 fitting code fitting_C2.m in \Twqubit_Circuit_PPS\simulation

When I compared the thermal spectrum of C2 and experimental thermal, they do not match at the small J-couplings. This program is written to fit the small J-couplings of C2 by brutal force. I only need to fit the frequency 8900 to 8920 region as all small couplings are involved in it. There are 7 of them: C2C4, C2C5, C2C6, C2H1, C2H2, C2H3, and C2H5. Moreover, C2C4, C2C5, and C2C6 are fixed by the H-decoupled C spectrum. The second assumption is C2H2 and C2H3 have the same couplings to save time. The constraint is the width of this regions equals to the sum over all couplings, that is, only two variables then. C2H2 and C2H1.

The experimental spectrum is saved in 'thermal_exp_C_200.mat' and the scaled factor is defined as the integral (sum) of the region [8900 8920]. There are 13 peaks in total and each peak corresponds to an intensity. In the fitting, we consider the 13 peak frequencies as the target function f, defined as the sum over square error of each peak. If this f is small then 1Hz which means the peak locations are pretty close, go to check the intensities of every peak. The definition of this value g is the same as f. When g is small, it means the peak intensities have the best fit as well as the locations are close too. The T2*=310ms gives the best match. Refer to the original code for details and the output spectrum is shown below (plotted by 'compared_C2.m').

Besides, C7 is also compared with T2*=420ms. The figure plotted by 'compared_C7.m' is shown below.

Now it is ready to go to check every experimental spectrum of C2 and C7. Indeed, we will just observe these two.

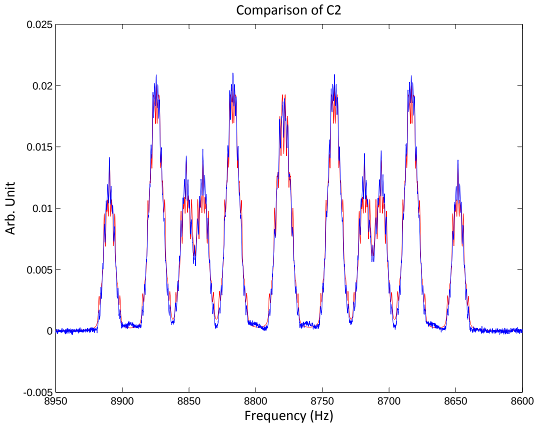

FIG. 4: Comparison of simulated and experimental C2 thermal.

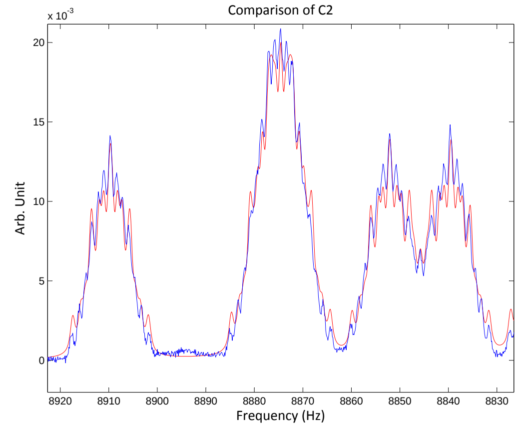

FIG. 5: Detailed comparison.

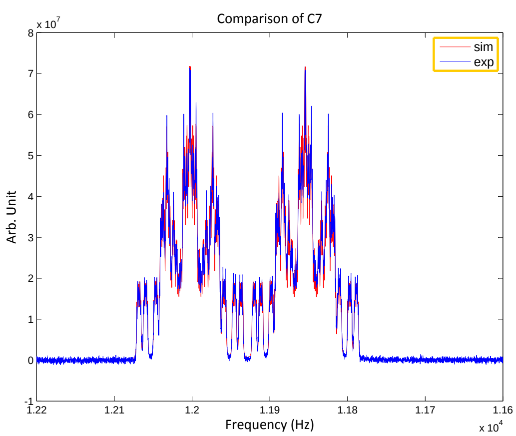

FIG. 6: Comparison of simulated and experimental C7 thermal.

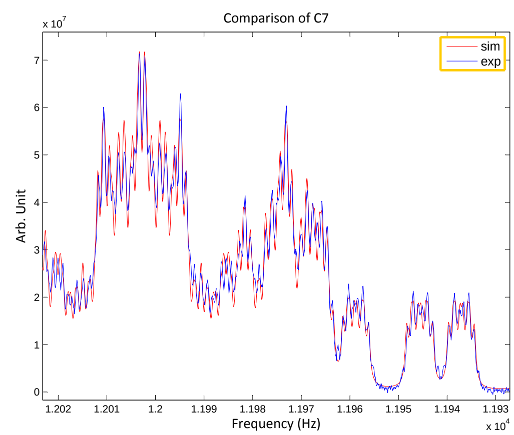

FIG. 7: Detailed comparison.

APR 16, 2015: 12-QUBIT CALIBRATION IN EXPERIMENT

Apr 16, 2015:

The calibration is used for getting the dB of $\pi/2$ and 2π pulses for a fixed length. The au program is written by Annie, including 3 files 'calib.m', 'calibrate.m' and 'getspec.m'. First upload all the three files to Ordi2. Note 'calib.m' is for redirection and it should be in the root folder when opening Matlab in Ordi2. The path is thus '\d29lu\fixpulse\' because Matlab will go to this folder to show the last result of pulse fixing.

The calibration folder is 'twqubit_calib'. Then select the experiment you want to calibrate, and run 'edau twqubit_calib'. **You have to edit Line 6 'host_resultpath', Line 24 'TIMES' which means number of loops, the following 'p1', 'd1' and 'pl3', and the starting power. Then in 'calibrate.m', change the targetfolder, ob_min, ob_max and figure_name.**

The experiments for calibrating C is titled 'twqubit_calib_C', and for calibrating H is 'twqubit_calib_H'. For H, I calibrated through the C channel as in experiment we only observe the C channel. The way is to transfer the H signal to C via XZ terms, so I can only calibrate the $\pi/2$ and $3\pi/2$ pulses. The results are summarized in the following table (**forget the column of June 25 as the setting is different and has been changed back**)

Rotation	Length	Frequency	Exp. No	dB (Apr 16)	dB (Apr 17)	dB (June 25)
C360	40us	25000Hz	11-23 (-3.5dB to -5.9dB)	-4.51dB	-4.38dB	-1.18dB
C360	80us	12500Hz	51-63 (3.0dB to 0.6dB)	1.82dB	1.74dB	—
C90	20us	12500Hz	101-113 (3.0dB to 0.6dB)	—	1.00dB	
H270 (by C)	30us	25000Hz	XXXX	—	6.20dB	6.20dB
H360 (by H)	40us	25000Hz	XXXX	—	5.40dB	

Actually only C360 and H270 are usable. The two powers are -4.38dB and 6.2dB for 25KHz.

Sep 01, 2015:

This part is to study the PHC0 and PHC1 parameters in Bruker. It is indeed important especially when people want to extract the phase information from NMR spectra.

PHC0 is known as the global phase. When PHC0 is changing, all the peaks in a spectrum are undergoing the same phase change. In other words, for peaks with different phases, PHC0 can never adjust them to the same phase.
PHC1 is a relative phase. It means the phase of one point (a given frequency) is fixed, and the phases of the other frequencies are changing relatively. The changing amount is proportional to the distance, that is, the frequency difference from the reference point. In fact, in Bruker spectrometer, when changing PHC1, the phase of the left-most point (largest frequency) is fixed and used as the reference. When PHC1 goes to 2π, the phase of the right-most point (smallest frequency) changes by 2π, and all the other points in the middle change phases accordingly based on their distances to the left-most point.

The formula when correcting phases by PHC0 and PHC1 is

$$\begin{aligned} R_0(i) &= R(i)cos[a(i)] - I(i)sin[a(i)], \\ I_0(i) &= I(i)cos[a(i)] + R(i)sin[a(i)], \end{aligned} \tag{5}$$

where $a(i) = PHC0 + (i-1)PHC1$ with $i > 0$. R and I represent the corrected values, and R_0 and I_0 are the original data. We can clearly see the relationship between the two phases.

For an NMR experiment, first we need to set the reference phase. One traditional way is to tune the phase of the thermal spectrum. From Eq. 5 we know a 20us hard pulse will induce the chemical shift evolutions for different spins, so there is a phase difference for different spins depending on how much angle they evolve under chemical shift evolutions. However, since this phase difference is proportional to the chemical shifts, they can be simultaneously corrected by PHC1. As PHC1 is changing, PHC0 will change accordingly because O1 (the middle point) feels a 0.5*PHC1 change and is no longer in absorption.

For a GRAPE pulse, it is totally different. GRAPE pulse has already considered the chemical shift evolution, so all the spins will feel the same rotating angles under one GRAPE pulse. In principle, PHC0 and PHC1 must be 0 to keep the phases of all spins pointing up after a $R_y(\pi/2)$ on thermal.

I used the NMRsim program provided by Topspin to check the above analysis. It is proved to be correct. For a hard pulse readout, we have to use 'apk' command to tune all phases up, whereas for a GRAPE pulse readout, PHC0 = PHC1 = 0 will give all phases up.

In experiment, even for GRAPE pulses, PHC0 cannot be 0 as there is a machine phase. This phase should be a constant if the spectrometer is stable. PHC1 should be 0 for a GRAPE pulse as there is no relative phase for different spins. But we can actually see by setting PHC1 = 0 we can never get all peaks up. So I guess there is a space between the end of the sequence and the start of acquisition and makes it similar to a free evolution before we read out the spectrum. We can adjust the phase of a GRAPE pulse spectrum to all peaks up and record this phase as reference. For future experiments they all follow this phase setting.

I also checked in simulation that PHC0 and PHC1 depend on TD and SWH indeed.

Conclusion if we want to rotate the thermal to X and observe it.
A hard pulse readout will give PHC0 and PHC1. PHC1 comes from the Hamiltonian evolution during the hard pulse, and the free evolution between the sequence tail and acquisition; PHC0 comes from PHC1 and machine phase.
A GRAPE pulse readout will give PHC0 and PHC1. PHC1 comes from the free evolution between the sequence tail and acquisition; PHC0 comes from PHC1 and machine phase.

APR 18, 2015 TO AUG 18, 2015: ALL PULSES TEST IN EXPERIMENT

Apr 18, 2015:

Tested all single-qubit pulses by a set of experiments for each. It is notable that when observing H channel as we have to set H channel as f1, the power calibration for H should be different from the setting that C is f1.

The experimental folder is 'twqubit_pps'. Exp 1 and 2 are thermal spectra.

For every pulse I have 4 observations. For C channel, for H channel, pulse-gradient-pulse for C channel, and pulse-gradient-pulse for H channel. Refer to the Appendix to see what the pulses are.

Polarization Crush
11-14: all90butC7

Phase Cycling
21-24: all90

Encoding
101-104: C57180
105-108: C23180
109-112: C234790
113-116: C156180
117-120: C234790withPC

Decoding
201-204: C2347andH180
205-208: C134690andH90
209-212: C23456180
213-216: C1234690withPC
217-220: C27180
221-224: C2Y5X90

Single $\pi/2$ rotations
301-314: C1 to C7 90 rotation (two experiments for one pulse)

Single $\pi/2$ rotations
315-342: C1 to C7 180 rotation (four experiments for one pulse)

Apr 23, 2015:

First try on the entire 12-qubit PPS circuit. One problem is found that the default 4us delay when calculating GRAPE pulses should be waived. So this part is not useful any more.

Exp 1000: zg for the thermal without H decoupled
Exp 1001: observe C7 after crushing all the other 6 carbons and a gradient
Exp 1002: the same as 1001 but decoupled H before observing C7
Exp 1003: Encoding 1. observe C7 and decouple H. Almost the same as the 7-qubit experiment.
Exp 1004: Encoding 2. observe C7 and decouple H. Almost the same as the 7-qubit experiment.
Exp 1005: PPS, gradient and observe C7 but no signal...
Exp 1006: PPS, without gradient and observe C7, no signal either...

There is a problem when I calculated the GRAPE. There is a default 4us pre-delay and post-delay on the target unitary, as in experiment when applying the GRAPE pulse these two delays are required (not the minimal value but since GRAPE has considered that it has no effect). However, we do not have enough slots (only SP0 to SP31) for GRAPE pulses so we have to combine them to a big shape. The problem is the time step in the big shape is chosen as 10us, which cannot absorb the 4us delay. If we get rid of the 4us delays during the big shape, the experimental result is totally different.

I tried to use 2us time step size in the big shape instead of 10us, so I can write the 4us delay. Then for every 5 steps

(2us*5=10us) the amplitude and the phase should be the same. In principal it is fine, but when I am fixing the pulse on the H channel, the random noise is so serious that we have to use a low-pass filter to remove the noise. As in the big shape the shape is not smooth any more, some wanted points will be removed by the filter too. I guess no way to solve this except the pulse is smooth.

So maybe need recalculation with 0us pre-delay and post-delay. Please refer to here to know the details of the new 0us delay pulses.

May 13, 2015:

New test with the No Buffer Pulses. Found a problem in pulse fixing and explained here. This part is no longer useful.

Tested in experiment with the new no-buffer GRAPE pulses. I can still get the 7 coherence by decoupling H channel. To my surprise, if I fixed all the pulses, the performance with the fixed ones is even worse. I guess the reason is when pulse fixing, the baseline is a little bit below 0 for the free evolution. It does not make sense to fix them but the fixing program is indeed trying to fix it. The solution might be fixing each pulse one by one, and then combine all of them with the 0 points for free evolution. It will be done soon.

Exp 1103: Ob Z24567 with H decoupling and non-fixed No Buffer pulses. Look good.
Exp 1104: Ob Z1234567 with H decoupling and non-fixed No Buffer pulses. Look good.
Exp 2103: Ob Z24567 with H decoupling and fixed No Buffer pulses. Worse than 1103, which is surprising.
Exp 2104: Ob Z1234567 with H decoupling and fixed No Buffer pulses. Worse than 1104, which is surprising.

For PPS, there is still no signal. See Exp 1999 and 2000. Exp 1999: PPS by non-fixed No Buffer pulses. No signal.
Exp 2000: PPS by fixed No Buffer pulses. No signal.

I tried 'fp' with about 200 scans to get a signal after one phase cycling step, and 80 scans with 'efp' and 'LB=0.3Hz'. So I divided the PPS into 24 experiments, and 80 scans for each experiment. D1 = 60s, so the total time for trying a PPS now is 34 hours...

Note: Explanations to Exp 2103 and 2104. The No Buffer pulses after fixing are even worse because the fixing program has a problem. The pulses itself is a big shape, with tons of 0 points inside to realize free evolutions. However, the pickup coil will detect slight signals for those 0 points (slightly over or below 0). The fixing program tries to correct these points by modifying these 0 to non-zero, which is not reasonable. One way to solve this problem is by fixing all pulses individually and combine them at last. So I fixed all the 18 pulses either 1ms or 2ms, and then used programs called 'pulse_combine_encoding_nobuffer_fixed.m' and 'pulse_combine_decoding_nobuffer_fixed.m' in '\Twqubit\Pulses_NoBuffer\' to combine all of them with 0 points.

Unfortunately the result is not good. It turns out that the signal is still very low with 80 scans. So maybe improve the number of scans can help a little bit.

May 22, 2015:

Re-run PPS with Scan=200. No signal again but the 24 experimental settings (Exp 2001 to Exp 2024) will be used in the future.

As there are three days that we will go to Ray's cottage, I set 200 scans for each phase cycling and the total length is thus 3d10hrs. I used a MAC program to automatically set all 24 experimental parameters. The Matlab program to generate this MAC file is 'Genmultizg.m' in '\Twqubit\'. Put the generated file to '\lists\mac\'. The 24 experiments are from Exp 2001 to Exp 2024.

Unfortunately there is no FID after the multizg. I tried some of the 24 experiments and they cannot do fp or efp. Need to figure out why. Another problem is during a long multizg, the sample will unlock at some time. Xinhua said the reason is gradient field. Because the gradient field is applied in z direction, it will affect the locking signal. But we cannot avoid the many gradient fields in between. Try experiment one by one? Anyway, do simulation now. That is, getting the simulated spectrum of each phase cycling, and then compare.

Re-run the 24 experiments again and now they have spectrum. Almost all 0. The sum is not a single peak. Meanwhile, the simulation is done. The folder is '\Twqubit_Circuit_PPS\'. First I ran 'rho_phasecycling_24steps.m' to get all MAT files of the 24 states, with each one corresponding to a phase cycling step. Then upload all the 24 MAT files to Ordi2 where the 'mkstate.m' function is. Run 'sim_12pps_step.m' in '\NMR\EXPERIMENTS\twqubit' to get all the simulated results and transfer them to my laptop. Plot all the 24 figures with the name such as '24_step_7.fig' for each one. Every spectrum out of the 24 figures is oscillating seriously which means if the T2* is not long enough (the case in experiment), we cannot resolve it.

June 11, 2015:

Found the best parameters for 12 qubits.

Before experiments, I want to find the best setting for 12 qubits.

SWH = 30030Hz. DW = 16.65us.
SWH is a fixed value. DW is related to the sampling rate, and reciprocal (inversely proportional) to SWH.

TD = 65536 and SI = 262144(max).
The real spectrum will have SI points. So if SI¡TD, it will cut the first SI points from TD when doing FFT. Otherwise, it will add the number of SI-TD points to the tail when doing FFT. Because SWH is very big, we need many many points to keep a high resolution. That is why SI is so big. Meanwhile, if TD is much smaller, we will use a lot of zero points added to the tail of TD, which means the noise is canceled a lot. Small TD gives us a good SNR, but it loses the resolution of small couplings. **(For H-decoupled experiments, TD = 281684 for a high resolution.)**

AQ = 1.09s.
AQ is fixed as AQ = DW*TD.

Aug 10, 2015

Implement every encoding and decoding part and compare with simulation.

The experiments are (Spectra are shown in Appendix II).

Exp 1398: Observe C7 by Gaussian with H decoupled. NS = 1. TD=281684.
Exp 1399: Observe C2 by Gaussian with H decoupled. NS = 1. TD=281684.
Exp 1401: Observe C7 by GRAPE as the reference. NS=10. See Fig. 9
Exp 1402: Observe C2 by GRAPE as the reference. NS=10. To my surprise the phase is different with 1401.
Exp 1403: Observe C7 after encoding1. NS=10. See Fig. 10
Exp 1404: Observe C2 after encoding1. NS=10.
Exp 1405: Observe C7 after encoding1 and decouple H. NS=1. See Fig. 11
Exp 1406: Observe C2 after encoding1 and decouple H. NS=1.
Exp 1407: Observe C7 after encoding2. NS=10. See Fig. 12
Exp 1408: Observe C2 after encoding2. NS=10.
Exp 1409: Observe C7 after encoding2 and decouple H. NS=1. See Fig. 13
Exp 1410: Observe C2 after encoding2 and decouple H. NS=1.

For encoding3 which generates ZZZZZZZZZZZZ, we tested that it cannot be observed. The H channel cannot be decoupled as the state of H is not I. So we directly applied encoding 3 to thermal state, and C-H couplings will evolve.

Exp 1411: Observe C7 after encoding3, applied on thermal. NS=1. See Fig. 14
Exp 1412: Observe C2 after encoding3, applied on thermal. NS=1.
Exp 1413: Observe C3 after encoding3, applied on thermal. NS=1.
Exp 1414: Observe C4 after encoding3, applied on thermal. NS=1.

Prepare the 7-qubit PPS and replace some of the pulses with the 12-qubit GRAPE. This can check the performance of these 12-qubit GRAPE pulses. All the sequences have a C-H SWAP in the very beginning to improve the C signal.

Exp 1887: Observe C7 on thermal. NS=7. See Fig. 15
Exp 1888: Observe C7 after C-H SWAP applied on thermal. NS=7.
Exp 1889: Observe C7 of the 7-qubit PPS. NS=7. See Fig. 16
Exp 1891: Observe C7 of the 7-qubit PPS, with the Encoding replaced by 12-qubit pulses. NS=7.
Exp 1893: Observe C7 of the 7-qubit PPS, with the Encoding and Phase Cycling replaced by 12-qubit pulses. NS=7. See Fig. 17

Sep 09, 2015

The implementations of every encoding and decoding part are repeated with C-H SWAP gate in the beginning. Just increase the Exp No by 1000, for example, 1401 to 2401.

A program 'Compare_EveryStepWithSWAP.m' in the folder '\Twqubit_Circuit_PPS\simulation\' is written to read all spectra data and plot figures. The operators are saved in '\All_Exp_Operators' and the figures in '\All_Sim_and_Exp_Compare_Figures'.

All the following experiments have a C-H SWAP gate in the beginning, except Exp 2399 and 2402, as we cannot observe C2 after the SWAP gate.

Exp 2398: Observe C7 by Gaussian with H decoupled. NS = 1.
Exp 2399: Observe C2 by Gaussian with H decoupled. NS = 1.
Exp 2401: Observe C7 by GRAPE as the reference. NS=10. See Fig. 18
Exp 2402: Observe C2 by GRAPE as the reference. NS=10.
Exp 2403: Observe C7 after encoding1. NS=10. See Fig. 19
Exp 2404: Observe C2 after encoding1. NS=10.
Exp 2405: Observe C7 after encoding1 and decouple H. NS=1. See Fig. 20
Exp 2406: Observe C2 after encoding1 and decouple H. NS=1.
Exp 2407: Observe C7 after encoding2. NS=10. See Fig. 21
Exp 2408: Observe C2 after encoding2. NS=10.
Exp 2409: Observe C7 after encoding2 and decouple H. NS=1. See Fig. 22
Exp 2410: Observe C2 after encoding2 and decouple H. NS=1.

Exp 2432: Observe C7 after encoding3 applied twice which generates 7 coherence. NS=10. See Fig. 23
Exp 2433: Observe C2 after encoding3 applied twice which generates 7 coherence. NS=10.
Exp 2434: Observe C7 after encoding3 applied twice which generates 7 coherence, and decouple H. NS=1. See Fig. 24
Exp 2435: Observe C2 after encoding3 applied twice which generates 7 coherence, and decouple H. NS=1.
Exp 2436: Observe C7 after decoding1new which generates 7 coherence. NS=10. See Fig. 25
Exp 2437: Observe C2 after decoding1new which generates 7 coherence. NS=10.
Exp 2438: Observe C7 after decoding1new which generates 7 coherence, and decouple H. NS=1. See Fig. 26
Exp 2439: Observe C2 after decoding1new which generates 7 coherence, and decouple H. NS=1.
Exp 2444: Observe C7 after decoding2toend which generates 7 coherence. NS=10. See Fig. 27
Exp 2445: Observe C2 after decoding2toend which generates 7 coherence. NS=10.
Exp 2446: Observe C7 after decoding2toend which generates 7 coherence, and decouple H. NS=1. See Fig. **??**
Exp 2447: Observe C2 after decoding2toend which generates 7 coherence, and decouple H. NS=1.

Sep 10, 2015

Experimental result of the entire circuit without the phase cycling part.

1) Observed C7 after C-H SWAP gate. It is 3.39 (maximum 4) times higher than the C7 result without SWAP. This experiment is used as reference. Fig. 18 shows the spectra of C7 and C2. Note for C2 experiments, I cannot apply a SWAP gate first because after the SWAP, C2 is no longer at Z.

2) Apply Encoding1 and observe C7 and C2 with (Fig. 19) and without (Fig. 20) H decoupled. The length is 32.98ms and the left signal is 0.6974, 0,3313, 0.6498, 0.6090. From left to right the values are C7 with H coupled, C2 with H coupled, C2 with H decoupled, and C2 with H decoupled, respectively. The same for the following cases.

3) Apply Encoding1+Encoding2 and observe C7 and C2 with (Fig. 21) and without (Fig. 22) H decoupled. The length is 54.26ms and the left signal is 0.1133, 0.1563, 0.4498, 0.4844.

3) Apply Encoding1+Encoding2+Encoding3*2 and observe C7 and C2 with (Fig. 23) and without (Fig. 24) H decoupled. After Encoding3, we generated 12 coherence so it is not observable. The solution is we can double applied Encoding3 and then the state evolves to 7 coherence again which is able to observe. The length is thus 68.98ms and the left signal is 0.0864, 0.1764, 0.3148, 0.3443.

4) Apply Encoding1+Encoding2+Encoding3+Decoding1new and observe C7 and C2 with (Fig. 25) and without (Fig. 26) H decoupled. Decoding1new has two replacements compared to original Decoding1. First, the H hard pulse is replaced by the last pulse (C234790withPC) of Encoding3. Second, the last pulse (C134690andH90) of Decoding1 is replaced by

the first pulse (C234790) of Encoding 3. We have to do this replacement because direct application of original Decoding1 will give us no signal, but with Decoding1new we can still get 7 coherence. The length is 68.98ms which is the same as Encoding3*2 but the left signal is a bit smaller 0.0683, 0.1538, 0.2477, 0.2786.

5) Apply Encoding1+Encoding2+Encoding3+Decoding1new+Decoding2toEnd and observe C7 and C2 with (Fig. 27) and without (Fig. **??**) H decoupled. The final state is 7 coherence. The length 133.58ms is about equivalent to the length of the entire circuit. The final signal is 0.0280, 0.0585, 0.0805, 0.0816

Here is a summarized table of the remained signal after every step. We can see the H coupled signal is much less than H decouple signal. I think the reason is the H coupled signal is too complicated and the cancelation of up and down signal is too serious. So we can use the H decoupled signal as the meter of the signal attenuation for every step.

Case	Length	C7 Hcouple	C2 Hcouple	C7 Hdecouple	C2 Hdecouple
Encoding1	32.98ms	69.7%	33.1%	65.0%	60.9%
Encoding2	54.26ms	11.3%	15.6%	45.0%	48.4%
Encoding3*2	68.98ms	8.6%	17.6%	31.5%	34.4%
Decoding1new	68.98ms	6.8%	15.4%	24.8%	27.9%
Decoding2toEnd	133.58ms	2.8%	5.9%	8.1%	8.2%

Oct 09, 2015

A problem of applying many gradient fields in the spectrometer.

A problem was found during the PPS preparation. In the beginning, we planned to use 9 gradient fields in the sequence and implemented each phase cycling by hundreds of times. We found that during the running, the lock will be automatically off! Just before the second run, the spectrometer cannot stay locking any more, and the spectrum will have a obvious drift of chemical shifts. It is caused by applying too many gradient fields and the reason remains unknown. Probably the spectrometer thinks you are not friendly at all as you order him to do so many gradients...

Problem: Lock will be automatically off during a multiple scan of experiments with many gradient fields.
Solution: Reduce the number of gradients or the strengths; Reduce D1; Reduce NS.

In the new sequence, we have reduced the gradients to 5.
Gz1: After C-H SWAP gate.
Gz2: After Encoding1.
Gz3: After Encoding2.
Gz4: After Encoding3.
Gz5: Before C790 observation.

Two gradients (including killing all C polarizations before C-H SWAP and killing C1-C6, H1-H5 polarizations before Encoding1) are replaced by phase cycles. So a minimum NS=4 is required. Anyway, this is not a problem as we need much much more scans than 4 in 12 qubits.

Oct 10, 2015

Two changes for the PPS experiment.

Two changes were made during the test.

Change 1: 12 phase cycles are enough instead of 24. Actually Exp 1-12 are exactly the same as Exp 13-24. From now on we only use 12 different sequences.
Change 2: After the circuit, a pulse to rotate C7 back to Z was added. Gz5 and C790 observation followed this pulse. The advantage is the signal might be pure with this additional gradient.

The related Matlab programs of generating all sequences and spectrometer Mac commands are in 'H:\Matlab\Twqubit\'.
1. Gen24Sequences.m: Generate all 12 sequences and manually transfer them to the wave folder in the spectrometer computer.
2. Genmultizg.m: Generate one file 'twqubit_pps_phasecycling' and manually transfer it to the mac folder in the spectrometer computer. It will automatically generate all parameter settings of the 12 experiments, such as NS, D1, SP etc. Meanwhile, generate 'twqubit_pps_zg' which can be used to run all experiments. It also runs one thermal in between any two different phase cycles to check if the spectrum drifts or not.

Oct 20, 2015

Check different pulse fixing positions.

We refixed all pulses and the error of each fixed pulse is tiny. However, when we go back to experiments, combine all fixed pulses together, and run them, the spectrum is much worse than the previous result.