



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

SEMESTERTHESIS

Automatic Multi-Qubit Gate Calibration

Author:

Johannes HEINSOO

Supervisor:

Yves SALATHÉ

Prof. Andreas WALLRAFF

spring 2013

Contents

1	Introduction	3
1.1	Motivation	3
1.2	Short history of the field	3
1.3	Experiment	4
2	Experiment control software	6
2.1	Operation and data-flow	6
2.1.1	Usage of standard output	8
2.2	Configuration files	8
2.2.1	QubitCalib.ini	9
2.2.2	PatternConfig and PulsesConfig	9
2.2.3	Cleansweep config	10
2.2.4	Backups, logs and paths	11
2.2.5	Proposal for naming convention	11
2.3	Additional improvements	12
3	Multiple qubit operations	13
3.1	Fluxpulse controlled qubit-qubit interaction	13
3.2	C-phase gate calibration measurements	14
3.2.1	Relative phase measurement	15
3.2.2	Relative phase landscape	16
3.2.3	Dynamic phase measurement	19
3.3	Bloch sphere interpretation of C-PHASE gate	20
	Summery and acknowledgements	24
	References	25

Abstract

Transmon qubits coupled to microwave cavities have been recognised as a convenient platform of quantum information technology (QIT) experiments. To investigate the properties of these systems and to show predicted quantum phenomena various experiments are done, where state transitions are driven with microwave pulses. There are a number of drifts of parameters in such experiments, which need daily calibration. To calibrate eigenfrequencies, pulse shape and timing various standard measurements are done. As some of them are done quite often, automation of these operations pays off. After a short introduction, this thesis describes the improved data flow and configuration files of the automatic calibration software QubitCalib. In the second part multi-qubit measurements now embedded in QubitCalib are discussed. This includes comparison of experimental and simulated landscape of the relative phase measurement and visualisation of the C-PHASE gate operation on pair of Bloch's spheres.

1 Introduction

1.1 Motivation

Quantum computing is hoped to help us in several ways. Most famous examples of its applications are the factorization of large numbers using Shor's algorithm and exchanging secret data using quantum cryptography. Thus it makes classical cryptography crackable and at the same time provides a secure quantum communication channel as a replacement. But there are also a large number of other problems which are hard to solve on classical computers and which have faster quantum analogues. These include in addition to NP-complete and combinatoric optimization problems, also problems of scientific calculations including the simulation of complex quantum systems. The latter is complicated for classical computers due to rapid growth of the number of dimensions of the Hilbert space when free parameters are added. In contrast this is not a problem, in principle, for quantum computers which thereby would be an important tool also for physicists. In addition to quantum simulations there are other ways in which investigating quantum systems for information technology (QSIT) provides us a better understanding of quantum world in general. These include deeper understanding of decohering process, quantum measurements, non-locality of reality and borders of macroscopic classical and microscopic quantum worlds.

1.2 Short history of the field

The earliest proposal for using the laws of the quantum world for information technology purpose is considered to be by Stephen Wiesner in late 60s [Nielsen 00, Ch. 1.1.1], who proposed early ideas of quantum cryptography and superdense coding. By today many physical implementations have been proposed for QSIT purposes. Photons were used for first experiments, as they couple to their environment weakly keeping the quantum state's purity. Also polarization states map naturally to the Bloch sphere, a representation of the state-space of single quantum bits or qubits in short. But the weak photon-photon interaction in transparent mediums has so far prevented realising useful quantum algorithms in photonic systems. Due to well developed experimental techniques used in nuclear magnetic resonance (NMR) first QSIT experiments of quantum algorithms were possible [Chuang 98]. But as coherence time decreases with the size of the molecules [Gershenfeld 97] and since in liquid state room temperature experiments one always has to fight with thermal population in addition to unwanted couplings this approach is probably not scalable [Warren 97] or suitable for generic quantum computation [Menicucci 02].

Today there are several more promising platforms for QSIT. Among experiments with semiconductor quantum dots [Hanson 08] and trapped ions [Haffner 08] in the last ten years superconducting qubits have quickly proven to be one of the most promising approaches [Clarke 08]. They can be considered to be artificial atoms in resonators which is an important and convenient setup of quantum optics. Instead of optical resonators on-chip coplanar wave-guides and instead of timing flying atoms qubits are fabricated onto the same chip. Thus the experiments are easier to control and upscale. The qubits themselves consist of two superconducting islands separated with Josephson junctions, which provide the needed non-linearity. The ones designed with large capacitance for making them insensitive to charge noise are referred to as transmon qubits [Koch 07]. The same design also gives them a large dipole moment for better coupling to one or more resonators. The formed loop couples to a magnetic field, which is used to tune energy levels of the artificial atoms into resonance with other qubits in the resonator or with the resonator itself for coherent exchange of photons. By design the qubits are controllable with microwave frequency range pulses, which enable good control of timing, strength and phase of the pulses - something which is much harder in the optical frequency range.

Coherent control of superconducting qubits was first achieved in 1999 [Nakamura 99]. In 2004 strong coupling to coplanar microwave resonator was realized [Wallraff 04]. Since then inter-qubit coupling [Steffen 06], multiple qubit tomography [Filipp 09], error correction [Reed 12] and even 3 qubit Shor's code have been demonstrated [Lucero 12]. Some of the limiting factors are still qubit coherence times and fidelity of microwave photon detection. For a more detailed overview of late developments refer to [Eichler 13, Ch. 1.3].

1.3 Experiment

The experiments in this thesis were carried out with the setup and sample used for the quantum teleportation experiment [Steffen 13]. As can be seen from Figure 1, the chip has four transmon qubits in three resonators two of which are connected to input-output ports. Each qubit has its charge line, which is used for voltage biasing and drive pulses. The drive pulses are microwave pulses with controlled phase and amplitude. These enable qubit state rotation on the Bloch sphere around arbitrary σ_{xy} plane axes, where σ represents Pauli matrices. The σ_z rotations can be done using a sequence of σ_x and σ_y pulses or by detuning (analogue of Larmor precession in the rotating wave approximation) [Baur 12][Ch. 2.2.1]. Using the described rotations, all single qubit gates can be realised. For best gate fidelities parameters like pulse amplitude, frequency and ratio of quadratures are calibrated. For multiple qubit gates each qubit also has a magnetic flux line such that it could be tuned

into resonance with other qubits or with the resonator by fast current pulses. This requires calibration of the flux pulse amplitude, length and also relative timing and crosstalk effects of flux pulses.

Fine-tuning these parameters becomes a tedious task for more complicated pulse sequences and this is why the software QubitCalib has been developed. In the next Chapter I will describe the improved data flow and some other new features of the software together with a few principles which should be kept in mind in future software development. Then I will discuss the way how multiple qubit operations are done in more detail and also what kind of measurements reveal the optimal parameters for the gates. The aim of all of it is to reduce experiment time and thus make it easier to try out more sophisticated quantum algorithms and measurements.

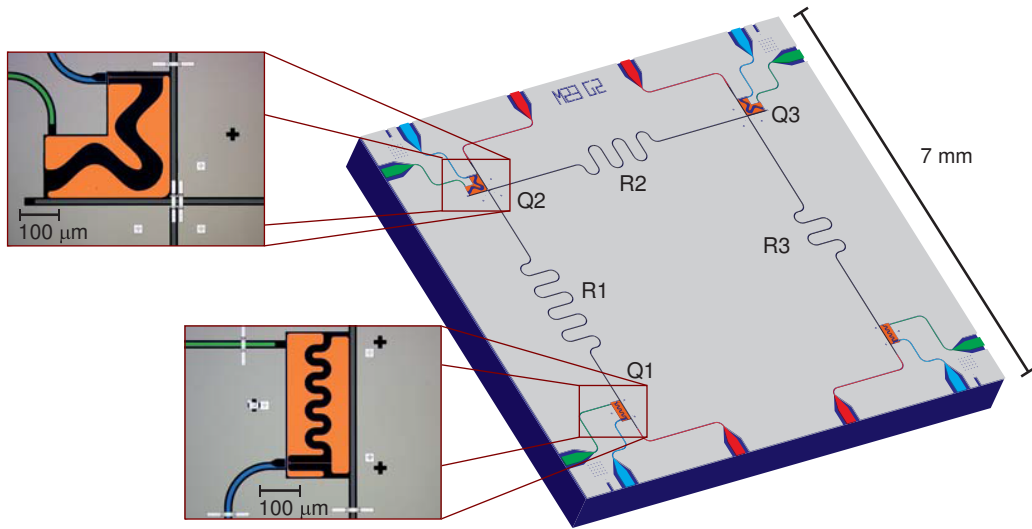


Figure 1: Rendered image of the sample used to test software and methods developed in this thesis. There are four qubits with voltage bias lines (green) and magnetic flux lines (blue). Qubits are connected with three resonators, out of which R1 and R3 can be read out using the readout ports (red). Sample fabrication, characteristics and image courtesy [Steffen 13].

2 Experiment control software

For more complicated experiments software becomes an important factor determining the experimental capabilities. Software used in the Quantum Devices group for experiment control can be divided into two parts. The first part called the Cleansweep is a program written in LabView and has been used with several setups in the Quantum Devices group. Its task is to control all of the physical devices such that the user or higher level software does not need to handle communication and control of the hardware. QubitCalib is another LabView program used for automated calibration using standard measurements. On the front end one can configure the calibration sequence and choose which devices are used to control physical qubits. Detailed configuration of devices and physical characteristics of the experiment are stored in a number of configuration files. Based on the settings QubitCalib

1. calls the Mathematica pattern generation script,
2. runs the experiment using Cleansweep and
3. calls the Mathematica data analysis script

for each calibration operation in the sequence. In the end of the sequence a log file is assembled from the output of the analysis scripts and shown to the user. Automatic calibration of single qubit operations in QubitCalib was completed by Tim Menke [Menke 13] and a generalization of this to include also transitions to the second excited state has been implemented by Andreas Landig [Landig 13]. More detailed description of software architecture regarding data flow and configuration files follows.

Below many configuration files are mentioned and as most of the files have a different operation specific name and even file name extensions of the same file vary they are referred using nicknames. These nicknames are important to help people understand better, which files are meant in conversations and documentations. More or less agreed nicknames are written in `typewriter` font and sometimes alternatives are also mentioned using *italics*.

2.1 Operation and data-flow

When considering where to add new parameters or where to look for solving problems in operation it is important to understand how different parts of QubitCalib interact and how information is passed. The input-output relations of different software components are schematically represented on Figure 2.

Calibration operation begins with pattern generation but it cannot start without initial parameter values. In addition to `QubitCalib.ini` which contains more abstract parameters, `PatternConfig` file is needed with details of the physical properties of pulses, such as the

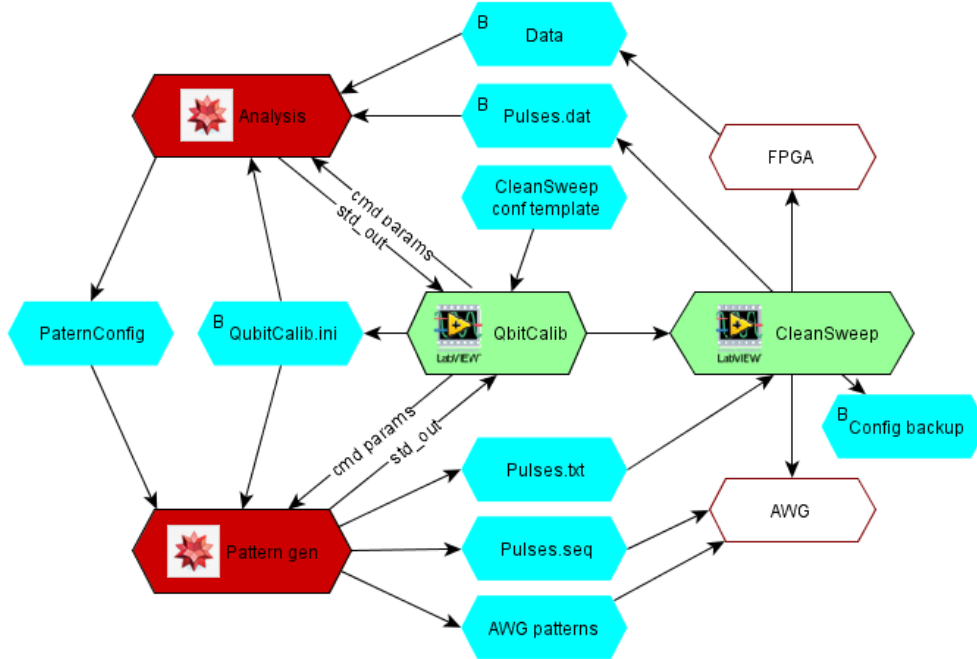


Figure 2: Description of data movement between components of QubitCalib. Red and green colors indicate Mathematica and LabView scripts and VIs. Blue indicates files. Those files marked with 'B' are backed up in the `data` folder. White coloured blocks mark hardware.

length and amplitude. This file is changed by hand when the experiment is initially calibrated after the cool-down and later the values are automatically updated during the routine calibrations. The path of the `QubitCalib.ini` file is passed to the pattern generation script via a command-line parameter. The path of the used `PatternConfig` is chosen on QubitCalib front-end and passed to pattern generation inside `QubitCalib.ini`.

The pattern generation Mathematica script outputs following files:

AWG patterns - `.pat` binary file of integers representing every waveform and `.seq` file describing the sequence of `.pat` files to be executed by the AWG.

PulsesConfig - described in Section 2.2.2.

All of these files are saved into the folder of `PatternConfig` file sometimes referenced as *AWG drive* which is the network share the AWG can access. AWG patterns have subfolders with operation name and AWG number. While patterns are directly read by the AWG (controlled by CleanSweep) the `PulsesConfig` gets renamed and moved¹ to `data` folder by

¹If pattern generation Mathematica script adds some variables in the right format, CleanSweep can use it to show ongoing experiment output on the front end with appropriate axis but this is not mandatory.

CleanSweep, such that the Mathematica script used for pattern generation would not need to know anything about the `data folder` path or structure (controlled by CleanSweep) and at the same time all of the configuration and log files would end up in the same place by the end.

When the experiment is done, the data is stored to the `data folder`. For archiving purpose CleanSweep configuration and screenshot of its front-end are also saved there.

Analysis file gets the paths of `data folder` containing `pulsesConfig` and raw data from measurements again from `QubitCalib.ini`. It performs the analysis and sets new values of pattern parameters in `PatternConfig` unless `update PatternConfig` option on QubitCalib front end was disabled.

2.1.1 Usage of standard output

Both Mathematica scripts send string formatted debugging data and measurement results back to QubitCalib through standard output. In LabView it is parsed line by line. First white-space separated string of every line is considered to be a label. At this point there are 3 fixed labels.

ErrorMsg is the only mandatory one. It contains string formatted verbal error message. If there was no error, **ErrorMsg** `none` should be returned, otherwise QubitCalib thinks, that Mathematica script had an unexpected error and will not continue.

Error carries the error code. This can be used to indicate a class of errors. For example **Error** `6004` indicates that there was inconceivable parameter in the configuration file.

A list of used error codes can be found from `error_codes.txt` in the QubitCalib folder.

Rerun - if it is returned with value other than 0, the calibration sequence will be repeated.

This can be used for example in case the accuracy is not yet good enough. At this moment only multiple qubit operations support this feature.

In addition to this an arbitrary label can be used to send back any sort of string formatted debugging data from Mathematica to the QubitCalib front end. If one wants to enable or disable the labelled messages shown in the status box during the calibration the corresponding label has to be added to "Debug tag list" on the QubitCalib frontend.

2.2 Configuration files

There are many configurable parameters which give the software the flexibility needed for physics experiments. There is not only more parameters than could fit onto the front-panel of the software but also too many to hold in a single type of configuration file. As complexity of the experiment setup is growing vastly it is important to introduce certain principles

of modularity into the software development. One component of this is stating the clear meaning and purpose of the different configuration files such that they would be used only for one purpose and for one kind of data. This might make the implementation of some new features more inconvenient, but it pays off later. For example while debugging one knows in which configuration file to look for certain kind of parameters without detailed knowledge of the software architecture. Below there is an attempt to describe meaning and purpose of existing files and of the ones added as part of the semester project.

2.2.1 QubitCalib.ini

`QubitCalib.ini` is generated by QubitCalib and contains parameters for Mathematica scripts. It includes only general QubitCalib operation and software specific parameters including

- paths for data input and output,
- paths to other configuration files (eg. `PatternConfig`),
- output channel indexes and corresponding logical qubits the pulses will be generated for,
- run number to indicate how many times the same operation has already been run, and
- logging index to give script a unique index in calibration sequence.

`QubitCalib.ini` should not contain information about physical properties of the experiment (for example frequency of qubit) nor setup specific information (e.g. model of the microwave generators). If it contains some information like this, it is due to convenience of exporting the data. Mathematica scripts should not make use of such parameters.

As this file contains paths of all of the other files and logs, it is in principle the only thing one needs to pass to Mathematica to run it by hand in lab or when testing Mathematica scripts for QubitCalib.

This file is saved in LabView cluster output format. It is also easy to read numerical and string values from it in Mathematica scripts, but some care has to be taken when passing arrays conveniently.

2.2.2 PatternConfig and PulsesConfig

`PatternConfig` contains information about the physical properties of the sample and experimental setup, which is needed to generate pattern files for AWG pulse sequences. `PulsesConfig` has information about already generated pulses and is needed to interpret the obtained data after the experiment. Here are some more important parameters contained in the files:

- π and $\frac{\pi}{2}$ pulse amplitudes,
- IF frequencies,

- DRAG pulse parameters,
- flux pulse parameters,
- AWG types (needed for exporting patterns in right format)
- AWG channel assignment (either mixer input quadratures or pair of fluxlines) and
- mixer calibration parameters (quadrature phase and amplitude).

`PatternConfig` and `PulsesConfig` files have basically the same contents. The only difference is that pulse generation Mathematica scripts add some description of the generated pattern to the file. These additional parameters are needed to interpret raw measurement data by analysis script and some of them are also used by the `CleanSweep` front-end (scales of plot axes for example).

These files are written in Mathematica syntax with few additions. Import functions provided by `PulsesGeneration` Mathematica library interpret lines beginning with `#` as comments and arrays as channel-pair-wise parameters². This file is also meant for running the experiment without central control of `QubitCalib` and needs to remain in a somewhat human readable for this purpose.

2.2.3 Cleansweep config

This file was first intended to be just a copy of the `CleanSweep` front-end settings so that it could be backed up after every measurement or reloaded with configurations for specific purposes. This also specifies the format used: LabView specified text file similar to `QubitCalib.ini`. Regarding `QubitCalib` it is used as a template file containing the full set of common configurations one uses in `Cleansweep` when doing such calibration measurements. Among many parameters the file contains the experiment's repetition rate, AWG output amplitudes, LO and RF generator frequencies, DC offsets for mixers and FPGA configuration. `QubitCalib` loads the file prepared for such usage - often called *CleanSweep dummy* or *Rabi-Ramsey-config*. After reading the file in, operation specific things are overwritten in the *experiment settings* cluster. For example which RF generators should be turned on. This is done for every step in the calibration routine. Then the cluster is passed to `CleanSweep` as Sub-VI parameter. While the experiment runs, one can double-check the parameters from the `Cleansweep` front end. The experiment settings cluster contains in addition to settings from `Cleansweep config` also a VISA communication configuration which is read in from another file `Cleansweep communication config`.

As `Cleansweep` controls all the details of the devices and electronics, this configuration is what

²In case this behaviour is not wished, one should add an extra layer of `{}` brackets around the array to make the outermost dimension to be 1.

this file should contain. Since Cleansweep is used for many purposes and without QubitCalib, the file should never contain some QubitCalib or higher level operation information.

2.2.4 Backups, logs and paths

There are few other files not shown on the data-flow diagram shown in Section 2.1 as they are not read by any program but are stored for logging purpose. In the QubitCalib log folder concatenated output from analysis scripts QubitCalib analysis log is saved as a pdf and also in nb format. In QubitCalib backup folder a backup of PatternConfig is saved³. The CleanSweep configuration for each experiment is saved for later reference into the so-called data folder. This is also the place for storing measurement output data. Names of sets of files in this folder corresponding to one experiment run by CleanSweep are prepended with an experiment identifier number referred to as *CleanSweep number*. Paths to all of the folders just named can be set on the QubitCalib front end. Automatically the data folder will contain subfolders corresponding to the date of the start of the experiment, but this behaviour can be overridden using QubitCalib front end options.

2.2.5 Proposal for naming convention

Most of the file names make sense, if the historical reason and content of the file is known. But to make it more intuitive and to force the principle of distinction and classification of parameters and stress the format of the file some unification should be made.

Most confusing of all is the nickname of QubitCalib.ini which instead could be named *QCalibOperationConf* as it contains settings for the pattern generation and analysis scripts used in a particular calibration operation. In contrast the file used to store all front end settings of the QubitCalib could be called *QCalibConf*.

Currently there is no consistency regarding extensions. As an example PulsesConfig has the extension .txt when generated by the Mathematica pulses generation script and later when renamed by Cleansweep it gets the extension .dat although the content is the same. Furthermore the .ini is often the nickname of QubitCalib.ini. Instead the extension should represent the file format. Here is one possible classification of formats:

.ini files generated using native format of LabView. Both reading and writing VIs are provided in the library OpenG. In Mathematica reading is also implemented and used in several places to read the QubitCalib.ini file. While it works well for numbers and string, some tricks are needed for convenient representation of arrays.

³After finishing the experimental work for current thesis also dumping QubitCalib front end settings and status log has been implemented.

`.cfg` are written in a Mathematica convenient format. For exporting one uses `InputForm` of Mathematica. After reading the tokens in they can be interpreted with `ToExpression`. `.bin` extension should be used for binary data files. `.dat` files should contain raw data tables in text form. `.txt` marks free formatted text. For example a human readable log file.

Although files already exported by current software will not be renamed due to compatibility issues with older data, other setups and large amount of hard-coded naming-conventions, this convention should be followed when naming new kind of files.

2.3 Additional improvements

Here a few additional important improvements are described which were realised as part of an assistant contract after the semester thesis.

In the pattern generation libraries there are special functions to evaluate list of expressions which describe exported AWG patterns. A usual experiment has a long sequence of patterns and often some AWG channels are always filled with zeros or other repeated patterns. To make pattern generation more efficient an Optimization step was added which reuses pulses already generated by referring to the same pattern file in the sequence description. To check, if such pattern is already used, a hashtable is used. For some sequences this saves more than two orders of magnitude in used HDD space, RAM usage, pattern generation and uploading time.

Existing mixer calibration subVIs were embedded such that it can be a step in the automated calibration routine. Also a framework for calling Mathematica commands in LabView was built to make use of exiting routines to export data plots for calibration log and to update the `PatternConfig` file in case for mixer calibration.

Sometimes things go wrong in the early part of the calibration routine. To make it easier for the user to notice the error before the whole routine is carried out, output plots are now shown on front end of the `QubitCalib` directly after analysing of every measurement. This feature makes use of the notification system developed to update the front panel from subVIs before they finish.

As multi-qubit operations needed number of new features and the single-qubit operations were improved at the same time, two branches of code emerged. To avoid making future changes and testing twice, these branches were joined. This required generalisation of several existing VIs and changes in `QubitCalib.ini` format, which also meant changing Mathematica scripts. In the future a similar merge should be done in Mathematica scripts as many scripts have big parts of identical code.

3 Multiple qubit operations

In addition to single qubit operations, multi-qubits gates are needed to realise quantum algorithms. There are two distinct ways to achieve these in the superconducting circuits [Baur 12, Ch. 6]: using sideband transition [Wallraff 07] or fast flux pulses [Majer 07]. The second one gives faster gates but is experimentally more complicated. I will shortly describe this method: how it is used to realise multi-qubit gates and the measurements done to calibrate this operation.

3.1 Fluxpulse controlled qubit-qubit interaction

Transmon qubit state energies depend on the magnetic flux and are given by following relation [Devoret 04]:

$$E_k = E_C \mathcal{M}_A \left[k + 1 - (k + 1) \bmod 2 + 2N_g (-1)^k, -2 |\cos \theta| \frac{E_{Jmax}}{E_C} \right] \quad (1)$$

where k is the state index with $k = 0$ being the ground state, E_C is the charging energy determined by the qubit island capacitance, N_g is the DC bias in units of cooper pair charges, E_{Jmax} is the maximum Josephson energy determined by the qubit design, θ is the phase difference of the superconducting condensates which is proportional to the magnetic flux and \mathcal{M}_A is the Mathieu characteristic function. It is useful to fix the DC bias to $N_g = 0.5$ as this gives first order insensitivity to charge noise and thus longer coherence times.

In the usual schemes qubits are *parked* far away from resonances with the resonator and other qubits using off chip flux coils. By using local flux lines of single qubit they can be switched into resonance with selected transition of another qubit. Even when qubits are off-tuned from the resonator they can still be coupled through exchange of virtual photons. Such interaction of two degrees of freedom is often called J -coupling and the interaction part of the Hamiltonian has the following form:

$$\hat{H}_{int}/\hbar = J_{00}^{AB} \left[(|0\rangle\langle 1|_A + \sqrt{2}|1\rangle\langle 2|_A) \otimes (|1\rangle\langle 0|_B + \sqrt{2}|2\rangle\langle 1|_B) + h.c. \right] \quad (2)$$

where coupling constant J_{ij}^{AB} upper indexes A and B represent involved qubits and lower indexes i and j ground states of transitions involved, constant $\hbar = h/2\pi$ with h being a Planck constant and $h.c.$ stands for hermitian conjugate. Using Equations 1 and 2 it is possible to predict transition frequencies for qubits with different flux bias. For qubit parameters from [Steffen 13]⁴ the transition frequency dependence on qubit A flux bias is plotted on Figure 3. The same parameters will be used for illustrative estimations throughout the thesis.

⁴Parameters for qubits 2 and 3 are used in experimental data which is showed later.

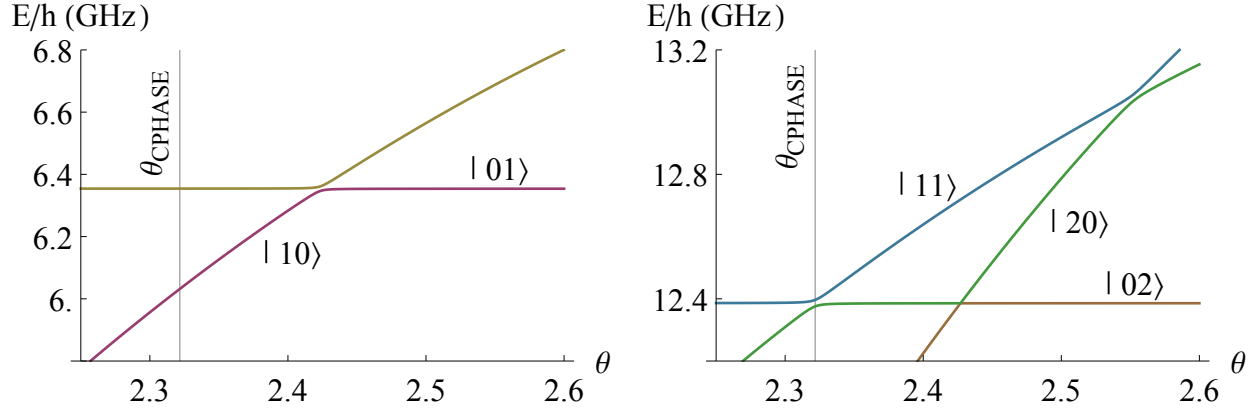


Figure 3: Energy dependence of two qubit states on the flux bias of the qubit A. The flux bias used for C-PHASE operation is marked with a vertical line. When the energies of two states are tuned to the same frequency, they are shifted by J_{ij}/π due to qubit-qubit coupling. The splitting is smaller for the second order process at the anti-crossing of $|02\rangle$ and $|20\rangle$.

When qubits states are tuned into resonance the excitation oscillates between two qubits at frequency $J_{ij}/2\pi$. By controlling the length of the flux pulse and choosing the resonant transitions different multiple qubit states can be produced.

When $|01\rangle$ and $|10\rangle$ are tuned into resonance for $t = \pi/2J_{00}$, an i -SWAP gate operation is obtained [Bialczak 10, Dewes 12]. This is because $|00\rangle$ and $|11\rangle$ remain eigenstates but single excitations are swapped $|01\rangle \leftrightarrow |10\rangle$ with an additional phase factor of i .

To get the controlled-PHASE (c-PHASE) gate, the $|11\rangle$ and $|02\rangle$ states have to be tuned to resonance for $t = \pi/J_{01}$ [Strauch 03]. If the length of the interaction enabling flux pulse is exactly right there will be an additional 180° phase shift turning state $|11\rangle$ to $-|11\rangle$. The goal of the calibration is to find out θ and t which would give the best C-PHASE gate.

Notice that on Figure 3 other transitions seem to be far enough from $|02\rangle$ and $|11\rangle$ transition used for C-PHASE operation. In the case shown on the plot the loss is 0.02^2 to state $|20\rangle$ and 0.03^2 to state $|01\rangle$. This seems to have a negligible effect, but as shown later it can be observed in our measurement.

3.2 C-phase gate calibration measurements

The qubit response to a flux pulse is approximately known from spectroscopy measurements. From the location of the anti-crossing one gets the amplitude for the flux pulse and the width of the splitting at this point corresponds to J/π . Here I will describe more sensitive and direct measurements.

For better estimation of the optimal flux pulse length and amplitude which would give the required operation the so-called Chevron pattern is measured. This consists of setting the operation qubits to state $|11\rangle$ using π -pulses on single qubits and applying flux pulses with varying length and amplitude. Before measuring the population of one of the qubits another set of π -pulses is done to convert $\{|11\rangle, |20\rangle\}$ superposition to a $\{|00\rangle, |20\rangle\}$ superposition for increased visibility. Pulse scheme and observed pattern of coherent oscillations are illustrated on Figure 4. For C-PHASE operation we are interested in finding the flux pulse amplitude and length which would end the initial state.

The Chevron pattern measurement has several downsides when it comes to calibrating the C-PHASE gate. First it is slow, as one needs to measure many points to accurately estimate the position of the sweet-spot. But it also lacks phase-sensitivity: from Figure 4b it can be seen, that for any flux pulse amplitude there is a point, where we get back to $|11\rangle$ state. Instead we get phase-factor of 180° only in case we have passed $|02\rangle$ pure state. This point will be discussed in more detail in Section 3.3 after having described phase-sensitive measurement methods.

3.2.1 Relative phase measurement

For daily recalibration a Ramsey-type of phase-sensitive measurement is done. It consists of two parts. In the first case qubit B is not pulsed and in the second it is excited to the

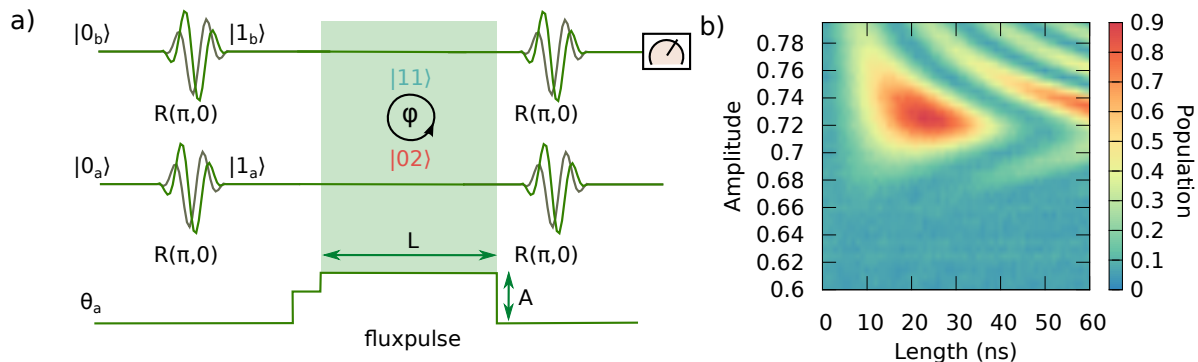


Figure 4: Pulse scheme (a) and measurement result (b) of the more accurate estimation of flux pulse length and amplitude for the C-PHASE gate. DRAG π -pulses are applied before and after the flux pulse. The flux pulse tunes $|11\rangle$ and $|02\rangle$ states into resonance and the population is exchanged between them. The measured population of one of the qubits depends on the length and amplitude of the flux pulse used and forms the so-called Chevron pattern.

state $|1\rangle_B$. Qubit A is in both cases $\pi/2$ -pulsed to turn it into superposition state. If now qubit A is tuned to qubit B using a flux pulse corresponding to the C-PHASE operation $1/\sqrt{2}(|01\rangle + |11\rangle) \rightarrow 1/\sqrt{2}(|01\rangle - |11\rangle)$ a state dependent swap of population should take place. Such a pulse scheme is illustrated on Figure 5a.

Thus a gathered phase of qubit A in the superposition state is different whether the qubit B is excited or not. To get a C-PHASE gate operation the flux pulse amplitude and length have to be such that phase difference of these two cases would be exactly 180° . To measure this relative phase an additional $\pi/2$ -pulse is applied on qubit A around $\sin\phi\sigma_x + \cos\phi\sigma_y$ similar to a Ramsey-type frequency measurement. Numeric phase value is obtained by fitting a cosine to population dependence of the phase ϕ . An illustration of such a measurement result is on Figure 5b.

3.2.2 Relative phase landscape

Relative phase values for different flux pulse lengths and amplitudes define a landscape and our goal is to find the extrema. When we assume, that we are not too far from the optimal flux pulse we can always assume power law dependence of pulse amplitude and length. Figure 6a shows a landscape obtained with 9 measurements and contours and center of 2D parabola fitted on the values. Such method is currently implemented in QubitCalib, the software used

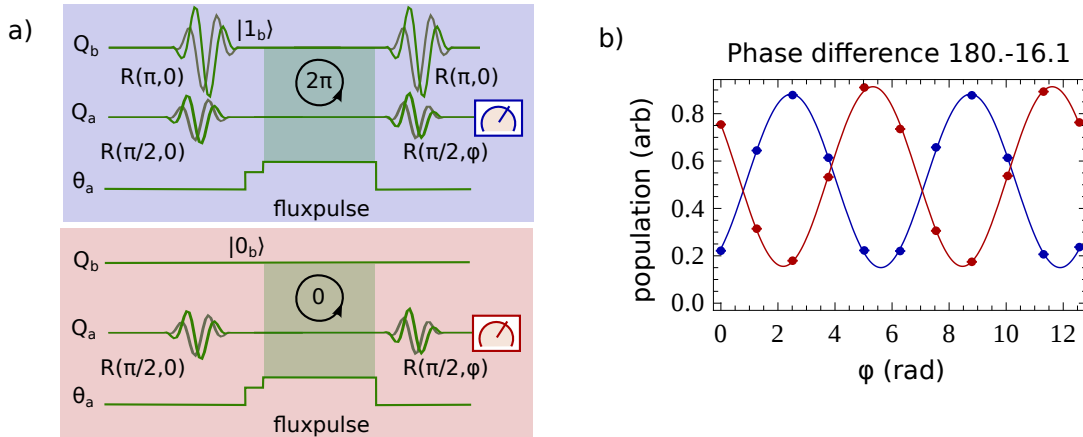


Figure 5: Pulse scheme (a) and measurement result (b) of the relative phase measurement. $\pi/2$ -pulses are applied to qubit A before and after the flux pulse, which tunes the $|11\rangle_{AB}$ state to resonance with $|02\rangle_{AB}$ but not to the states $|10\rangle_{AB}$ and $|01\rangle_{AB}$. Phase of the second $\pi/2$ -pulse is varied to measure the phase difference of qubit A in the cases when B was initially excited (blue) and when not (red). The population is scaled to the result of reference measurements.

for automatic calibration.

This method only works in case of good starting values. To improve that situation the relative phase landscape was studied in more detail. On Figure 6b the same landscape is shown obtained from a larger number of measurements. One can see, that the relative phase measurement is a lot more sensitive to flux pulse amplitude, than length. Also the formed valley seems to be tilted.

The features of the experimental data can be reproduced by first principle simulations as can be seen on Figures 7b and 7d. The valley-like shape of the relative phase measurement is not so intuitive and is discussed in more detail in Section 3.3 using Bloch sphere interpretation of C-PHASE gate. The tilt of the valley is due to weak coupling to other nearby energy levels. Plot 7d becomes symmetric when other non-diagonal elements of interaction Hamiltonian are removed.

As by the algorithm described in the beginning of the present section, only relative phase was optimised and it is insensitive to flux pulse length the algorithm leaves it uncalibrated or even might walk away from the optimal pulse length. As Figures 7a and 7c show, this corresponds to a smaller amplitude of the oscillations in relative relative phase measurement due to leaving some of the state population into $|02\rangle$ state.

So the automatic C-PHASE gate calibration method using relative phase measurements could be improved to make it converge into an optimal configuration from larger space of initial

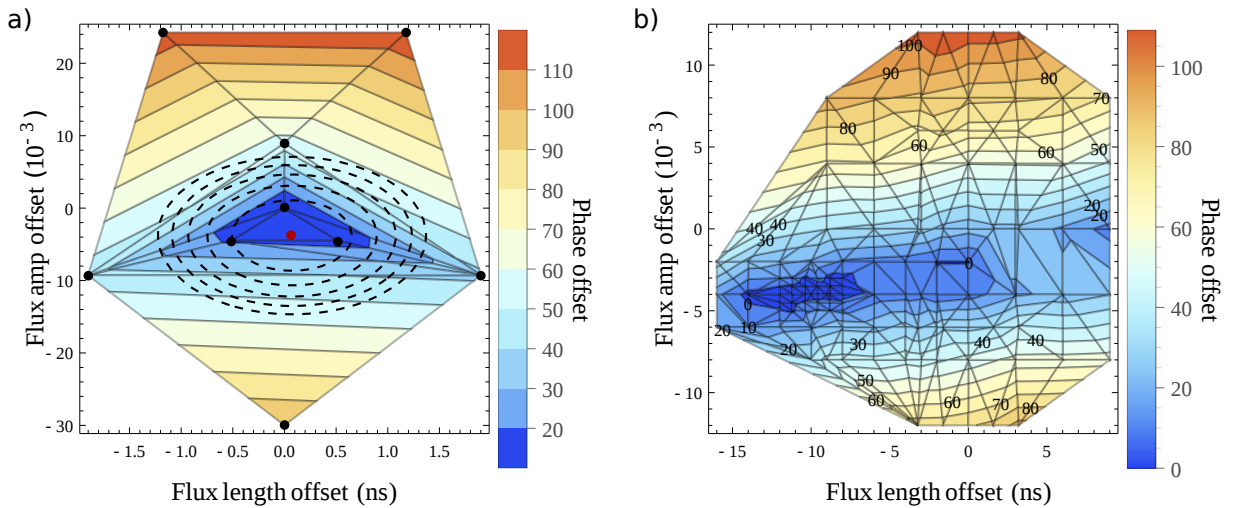


Figure 6: Experimental data of the offset of the relative phase from 180° dependence of flux pulse length and amplitude. In case of routine calibration (a) minima is easily found. More data points (b) reveal only a small sensitivity to flux pulse length. Flux pulse amplitude axes is in AWG output scale.

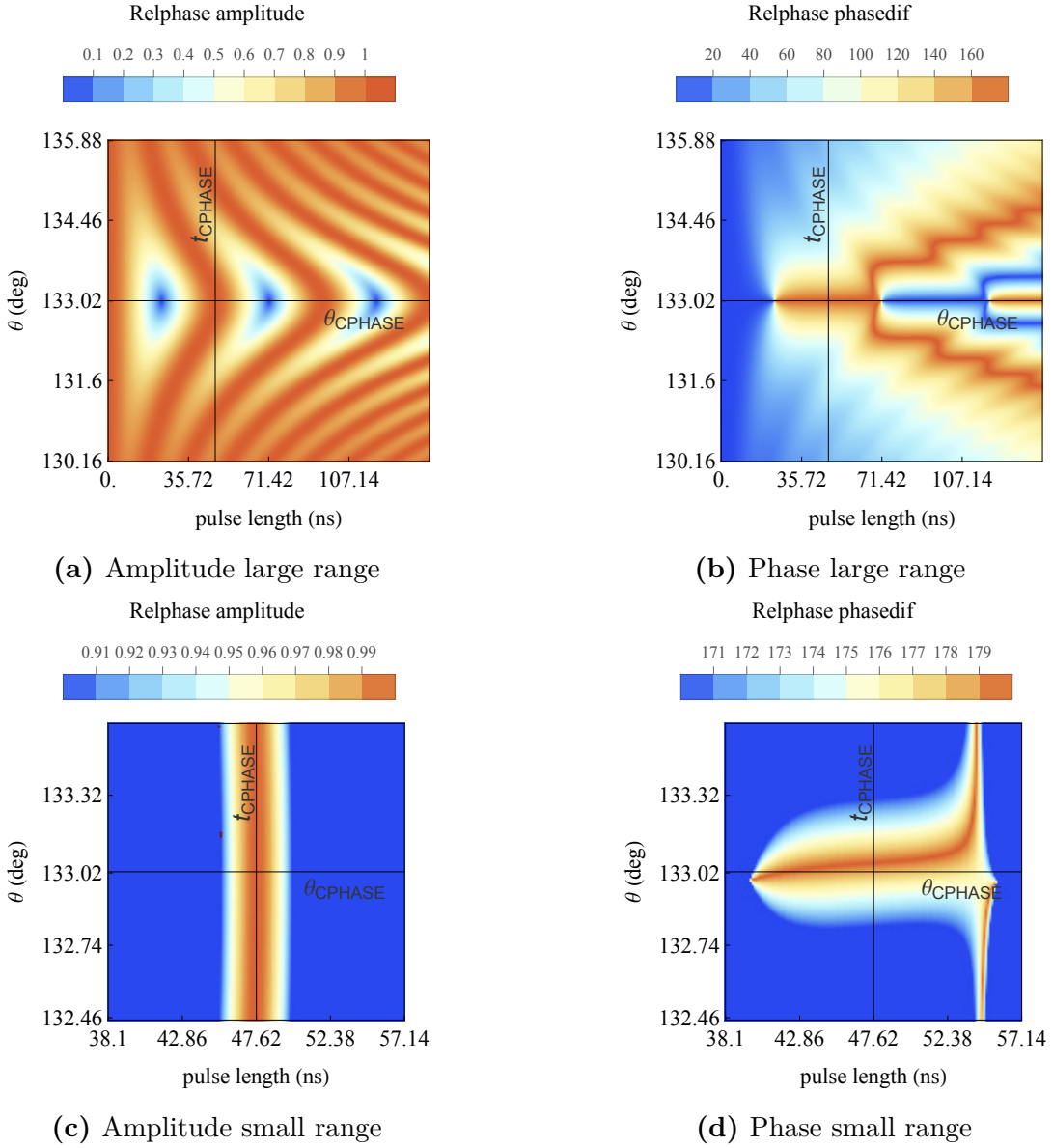


Figure 7: First principle simulation predictions of relative phase measurement amplitude and phase dependence on flux pulse length and amplitude. Grid lines mark the optimal length t_{CPHASE} and amplitude θ_{CPHASE} for C-PHASE gate operation. Relative phase measurement amplitude is given as proportion to maximum and relative phase is in degrees. On Figures 7c and 7d color range is clipped to increase contrast of the interesting range.

configurations. One way to improve the method would be to investigate how phase and amplitude from relative phase measurements correspond to gate fidelity and this might lead to more useful objective function compared to relative phase alone.

Another idea Figures 7a and 7b hint to is the possibility of doing C-PHASE gate using smaller or higher flux pulse amplitude θ . This might be needed, if there is some other resonance one wants to avoid. The big downside of this scheme is significantly longer gate time.

3.2.3 Dynamic phase measurement

Note, that if the relative phase was the only thing one needed to calibrate for the C-PHASE gate there should be no phase shift in the case when qubit B was not excited. Thus the measured population for $\phi = 0$ should be 1 in this case and 0 in the case qubit B was initially excited and the relative phase well calibrated. As Figure 5b illustrates, in both cases there is an additional absolute phase shift in addition to the 180° relative phase. This is due to flux pulse caused detuning of $|0\rangle \rightarrow |1\rangle$ transition resonance frequency $\nu_{0\rightarrow 1}^A(\theta) = (E_1^A(\theta) - E_0^A(\theta)) / h$. As transition energies determine the superposition evolution speed the phases of measured states are shifted in relation to rotating reference frame of corresponding parking frequency⁵. Both the $|1\rangle_A$ and the $|1\rangle_B$ states gather additional phase shift due to detuning from the reference. After successful relative phase calibration operation matrix is

$$U_{\text{C-PHASE}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{i\phi_A} & 0 & 0 \\ 0 & 0 & e^{i\phi_B} & 0 \\ 0 & 0 & 0 & -e^{i(\phi_A + \phi_B)} \end{pmatrix} \quad (3)$$

in the basis of $\{|00\rangle, |10\rangle, |01\rangle, |11\rangle\}$. This needs to be compensated by either single qubit phase gates or redefinition of the computational basis. Both require ϕ_A and ϕ_B to be known. For the C-PHASE gate qubit A has to be detuned by $\nu_{0\rightarrow 1}^A(\theta_{\text{C-PHASE}}) - \nu_{0\rightarrow 1}^A(\theta_{\text{parking}})$ which in our case⁶ corresponds to an oscillation period of 1.38 ns. This time-scale is of the order of relative phase calibration routine step determined by AWG sampling rate and also small enough to make predicting the dynamic phase inaccurate. Thus the relative phase has to be calibrated before dynamic phase measurement such that the used flux pulse length or amplitude would not change any more. Note also that not knowing ϕ_A and ϕ_B is the very reason why the relative phase measurement has to be done by varying the phase of the second

⁵In experiments we stay in this rotating reference frame frequency, as this is the one where we do single qubit pulses. It corresponds to sum of the local oscillator frequency ν_{LO} and intermediate frequency ν_{IF} .

⁶According to first principle simulation based on the same parameters as discussion before.

pulse and fitting the sine on to the data. Otherwise one could minimize the relative phase offset from 180° by doing the measurements with a single phase of the second $\pi/2$ -pulse.

Dynamic phase measurement itself is just another Ramsey-type of measurement. The qubit is $\pi/2$ -pulsed into a superposition state and after a given time $\pi/2$ -pulsed again. When the pulse measurement is done with several phases of the second $\pi/2$ -pulse phase offset of the qubit can be determined like in relative phase measurement. This is still a multiple qubit operation, as we are interested in the phase offset of the qubit in presence of local fluxpulse on a particular qubit. Dynamic phase for each fluxpulse has to be measured for for all qubits used in the experiment algorithm unless corresponding crosstalks could be neglected.

3.3 Bloch sphere interpretation of c-phase gate

There are the states $\{|00\rangle, |10\rangle, |01\rangle, |11\rangle, |02\rangle\}$ involved in relative phase measurements when unwanted coupling can be neglected. Thus there are still too many two-qubit state combinations needed to describe all features of the state evolution on Bloch spheres. On the other hand, if we project two qubit states into the basis $\{|0_A\rangle, |1_A\rangle\}$ and $\{|1_B\rangle, |2_B\rangle\}$ we can describe all important features on just two Bloch spheres although a lot of state space dimensions have been traced out.

On all off the Bloch spheres shown in Figures 8 and 9 there are two timelines with length $2 \times t_{\text{C-PHASE}} = 2\pi/J_{01}$ both corresponding to a part of the relative phase measurement. The time is represented by change of colors. Note also, that as here all transition frequencies correspond to flux pulse amplitude $\theta_{\text{C-PHASE}}$ we are in a rotating reference frame where we do not see a dynamic phase.

On Figures 8a and 8b a well tuned C-PHASE gate action is shown. We can only see the time line of the state starting from $\sqrt{1/2}(|01\rangle + |11\rangle)$ as the state $\sqrt{1/2}(|00\rangle + |10\rangle)$ does not evolve. It stays in an eigenstate of σ_x on Figure 8a and in the center of the sphere on the Figure 8b corresponding to a lack of population in the basis states which is a special case of a fully mixed state. On the other hand the population from state $\sqrt{1/2}(|01\rangle + |11\rangle)$ is moving into the space spanned by $\{|1_B\rangle, |2_B\rangle\}$. Note how the timeline in the basis formed by $\{|0_A\rangle, |1_A\rangle\}$ passes through the $|0_A\rangle$ state at the same moment as in $\{|1_B\rangle, |2_B\rangle\}$ the state makes it to equator. In the middle of the $2 \times \text{C-PHASE}$ evolution the population has left the $\{|1_B\rangle, |2_B\rangle\}$ bases and has exactly the opposite phase to the initial state in $\{|0_A\rangle, |1_A\rangle\}$. As in $\{|0_A\rangle, |1_A\rangle\}$ the timeline passes the $|0_A\rangle$ state. On the way back the state arrives in $\{|1_B\rangle, |2_B\rangle\}$ to a superposition once more but with opposite phase. In the end it arrives almost exactly back to the initial state.

For comparison on Figures 8c and 8d there is the same time evolution shown as in the figure

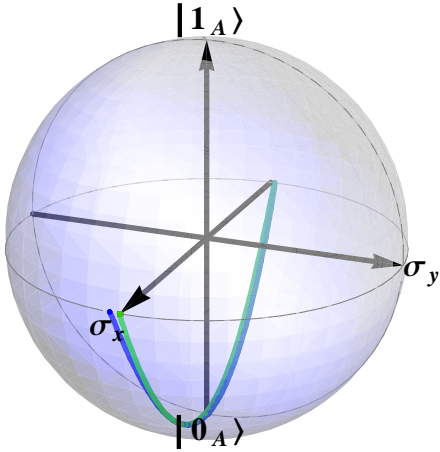
above, but now the coupling frequency is added to the rotating reference frame frequency. The choice of the timelines length results the timeline of the $\sqrt{1/2}(|00\rangle + |10\rangle)$ state making exactly one revolution on equator of Bloch sphere 8c. At the same time timeline of $\sqrt{1/2}(|01\rangle + |11\rangle)$ makes two circles through $|0_A\rangle$. Again in the middle of the evolution the two timelines have a relative phase of 180° . Note also, that the relative phase of XY plane projections of two timelines flips from to opposite phase each time $|0_A\rangle$ is passed. This explains the singularity-like jump on 7b and 7d on $t_{C-PHASE}$ slice and also the weak sensitivity to flux pulse length near $\theta_{C-PHASE}$ and $t_{C-PHASE}$ crossing.

The small gap between initial and final state on Figure 8a represents the same effect as the tilt of the valley on Figure 7d. It disappears when the coupling of states $|01\rangle$ and $|10\rangle$ is removed the from interaction Hamiltonian. If the coupling is there, the state $|0_A\rangle$ is not an exact eigenstate anymore, the frequency ν_{10} is slightly shifted and a small rotation of the Bloch sphere prevents the state from evolving exactly through $|0_A\rangle$. It can be seen from the Bloch spheres on Figure 9 that the same thing happens for detuned gates and results in having an imperfect $2 \times C-PHASE$ operation.

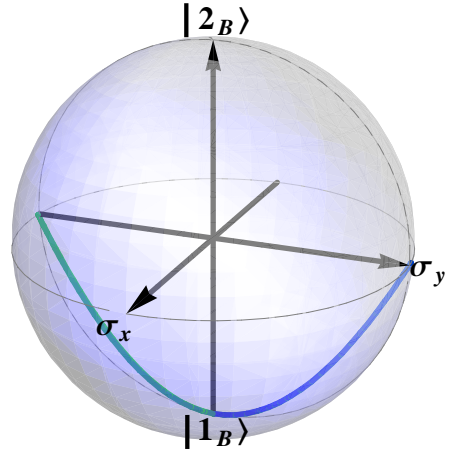
One thing which happens, if the flux amplitude is wrong, is that the evolution does not go through pure $|02\rangle$ state. It is seen from both Figures 9b and 9d that the first time the timeline arrives to the equator it is not with a 180° relative phase to initial state and the second time it does not get back to the initial state. The sign of the fluxpulse amplitude offset from the optimal value determines in which direction the timeline misses the $|0_A\rangle$ state and the amplitude of the amplitude offset obviously determines the amount. As $|0\rangle$ is missed the timeline does not reach the equator on Figures 9b and 9d.

The σ_z component on Figures 9a and 9c corresponds to the amplitude of the relative phase measurement. It is easy to see that the amplitude depends on the length and amplitude of the flux pulse in the same way as on Figure 7a.

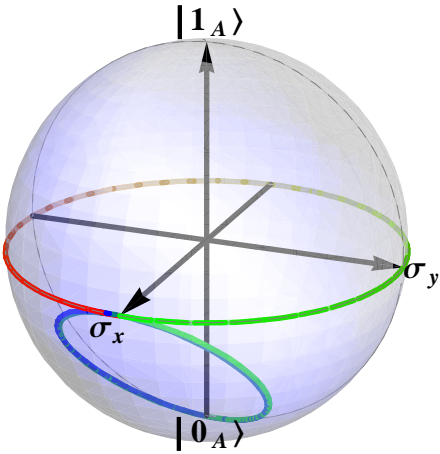
When the XY plane projection (the phase of the state) is now compared to one of the non evolving state $|00\rangle + |10\rangle$ it is clear that relative phase now depends on time. This relation can also be seen from Figures 7b and 7d.



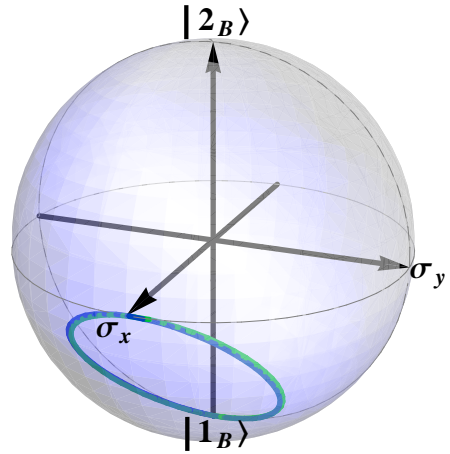
(a) $\nu_{\text{RRF}} = \nu_{10}/2$



(b) $\nu_{\text{RRF}} = \nu_{02} - \nu_{01}/2$

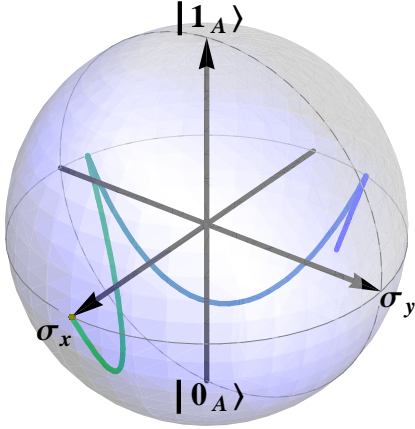


(c) $\nu_{\text{RRF}} = \nu_{10}/2 + J_{01}/2\pi$

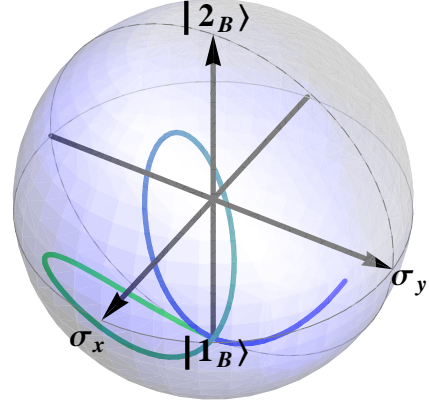


(d) $\nu_{\text{RRF}} = \nu_{02} - \nu_{01}/2 + J_{01}/2\pi$

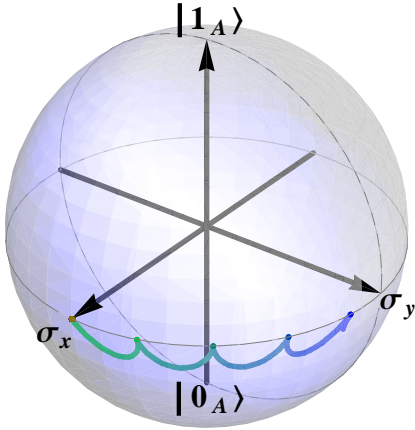
Figure 8: Nearly perfect $2 \times \text{C-PHASE}$ gate evolution visualised on pairs of Bloch spheres in two different rotating reference frames. Left ones are in bases $\{|0_A\rangle, |1_A\rangle\}$ and the right ones in $\{|1_B\rangle, |2_B\rangle\}$. The evolution starts with the green line and goes over to red in the case $|\psi(0)\rangle = \sqrt{1/2}(|00\rangle + |10\rangle)$ and to blue in the case $|\psi(0)\rangle = \sqrt{1/2}(|01\rangle + |11\rangle)$. Two consequent perfect C-PHASE gates would give back exactly the same state. As can be seen from all of the Figures the beginnings and ends of time-lines almost completely overlap. Notice also, that on Subfigures 8c and 8d blue time-lines do two overlapping loops and are thus hard to distinguish.



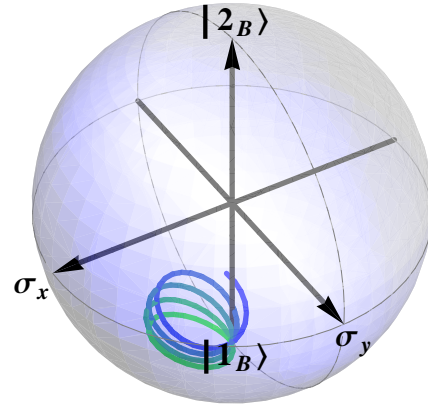
(a) $\nu_{\text{RRF}} = \nu_{10}/2$



(b) $\nu_{\text{RRF}} = \nu_{02} - \nu_{01}/2$



(c) $\nu_{\text{RRF}} = \nu_{10}/2$



(d) $\nu_{\text{RRF}} = \nu_{02} - \nu_{01}/2$

Figure 9: Off-tuned $2 \times \text{C-PHASE}$ gate evolution on pairs of Bloch spheres. On Subfigures 9a and 9b flux pulse amplitude is just slightly too big and on 9c and 9d it is too small and detuning is $3\times$ bigger.

Summary

For automated calibration the software QubitCalib had been developed in the Quantum Devices group. Having part of the software realised in National Instruments LabView and part in Wolfram Mathematica scripts, best use of both programming environments was made having simultaneously a powerful mathematics engine and a framework which is easy to control and develop. As part of the semester thesis project the data flow between the parts was greatly improved and hereby described in detail. This enabled several new features and the description saves time for future development and debugging. One of the new features enabled are more complex calibration routines.

To achieve C-PHASE gate operation with flux pulses one first has to calibrate the amplitude and length of the pulse. Approximate parameters are known from spectroscopic measurements and can be better estimated with Chevron type measurement. For daily fine-tuning one wants to do a relative phase measurement, which takes into account in addition to the output state amplitudes also the phases and thus is more sensitive. After the relative phase calibration also the dynamic phase has to be measured in order to be compensated or taken into account. The relative and dynamic phase measurements are analysed in the thesis and realised as QubitCalib operations.

Acknowledgements

The software QubitCalib has been developed in cooperation with Andreas Landig [[Landig 13](#)] and Tim Menke [[Menke 13](#)]. Relative and dynamic phase calibration scripts were based on Mathematica notebooks by the Quantum Information team. I would like to thank everybody from the QuDev group for pleasant atmosphere and especially Yves Salathé, Markus Oppliger, Philipp Kurpiers for their kind help and friendly collaboration.

References

- [Baur 12] M. Baur, A. Wallraff & D. Vion. Realizing quantum gates and algorithms with three superconducting qubits. ETH, 2012.
- [Bialczak 10] R. C. Bialczak, M. Ansmann, M. Hofheinz, E. Lucero, M. Neeley, A. D. O’Connell, D. Sank, H. Wang, J. Wenner, M. Steffen, A. N. Cleland & J. M. Martinis. *Quantum process tomography of a universal entangling gate implemented with Josephson phase qubits*. Nat. Phys., vol. 6, no. 6, pages 409–413, June 2010.
- [Chuang 98] Isaac L. Chuang, Lieven M. K. Vandersypen, Xinlan Zhou, Debbie W. Leung & Seth Lloyd. *Experimental realization of a quantum algorithm*. Nature, vol. 393, no. 6681, pages 143–146, May 1998.
- [Clarke 08] John Clarke & Frank K. Wilhelm. *Superconducting quantum bits*. Nature, vol. 453, no. 7198, pages 1031–1042, June 2008.
- [Devoret 04] M. H. Devoret, A. Wallraff & J. M. Martinis. *Superconducting Qubits: A Short Review*. cond-mat/0411174, 2004.
- [Dewes 12] A. Dewes, F. R. Ong, V. Schmitt, R. Lauro, N. Boulant, P. Bertet, D. Vion & D. Esteve. *Characterization of a Two-Transmon Processor with Individual Single-Shot Qubit Readout*. Phys. Rev. Lett., vol. 108, page 057002, Feb 2012.
- [Eichler 13] Christopher Eichler. *Experimental Characterization of Quantum Microwave Radiation and its Entanglement with a Superconducting Qubit*. PhD thesis, ETH Zurich, 2013.
- [Filipp 09] S. Filipp, P. Maurer, P. J. Leek, M. Baur, R. Bianchetti, J. M. Fink, M. Göppl, L. Steffen, J. M. Gambetta, A. Blais & A. Wallraff. *Two-Qubit State Tomography Using a Joint Dispersive Readout*. Phys. Rev. Lett., vol. 102, no. 20, page 200402, 2009.
- [Gershenfeld 97] N. Gershenfeld & I. Chuang. *Bulk Spin Resonance Quantum Computation*. Science, vol. 275, pages 350–356, 1997.
- [Haffner 08] H. Haffner, C. Roos & R. Blatt. *Quantum computing with trapped ions*. Physics Reports, vol. 469, no. 4, pages 155–203, December 2008.

- [Hanson 08] Ronald Hanson & David D. Awschalom. *Coherent manipulation of single spins in semiconductors*. Nature, vol. 453, no. 7198, pages 1043–1049, June 2008.
- [Koch 07] Jens Koch, Terri M. Yu, Jay Gambetta, A. A. Houck, D. I. Schuster, J. Majer, Alexandre Blais, M. H. Devoret, S. M. Girvin & R. J. Schoelkopf. *Charge-insensitive qubit design derived from the Cooper pair box*. Phys. Rev. A, vol. 76, no. 4, page 042319, 2007.
- [Landig 13] Andreas Landig. Software for arbitrary single qubit and qutrit gate calibration. Master’s thesis, ETH Zurich, August 2013.
- [Lucero 12] Erik Lucero, Rami Barends, Yu Chen, Julian Kelly, Matteo Mariantoni, Anthony Megrant, Peter O’Malley, Daniel Sank, Amit Vainsencher, James Wenner, Ted White, Yi Yin, Andrew N Cleland & John M Martinis. *Computing prime factors with a Josephson phase qubit quantum processor*. arXiv:1202.5707, February 2012.
- [Majer 07] J. Majer, J. M. Chow, J. M. Gambetta, J. Koch, B. R. Johnson, J. A. Schreier, L. Frunzio, D. I. Schuster, A. A. Houck, A. Wallraff, A. Blais, M. H. Devoret, S. M. Girvin & R. J. Schoelkopf. *Coupling superconducting qubits via a cavity bus*. Nature, vol. 449, no. 7161, pages 443–447, September 2007.
- [Menicucci 02] N. C. Menicucci & Carlton M. Caves. *Local Realistic Model for the Dynamics of Bulk-Ensemble NMR Information Processing*. Physical Review Letters, vol. 88, no. 16, pages 167901+, April 2002.
- [Menke 13] Tim Menke. Realizing a calibration program for superconducting qubits. Master’s thesis, ETH Zurich, February 2013.
- [Nakamura 99] Y. Nakamura, Y. A. Pashkin & J. S. Tsai. *Coherent control of macroscopic quantum states in a single-Cooper-pair box*. Nature, vol. 398, no. 6730, pages 786–788, April 1999.
- [Nielsen 00] Michael A. Nielsen & Isaac L. Chuang. Quantum computation and quantum information. Cambridge University Press, 2000.
- [Reed 12] M. D. Reed, L. DiCarlo, S. E. Nigg, L. Sun, L. Frunzio, S. M. Girvin & R. J. Schoelkopf. *Realization of three-qubit quantum error correction with superconducting circuits*. Nature, vol. 482, pages 382–385, February 2012.

- [Steffen 06] M. Steffen, M. Ansmann, R. C. Bialczak, N. Katz, E. Lucero, R. McDermott, M. Neeley, E. M. Weig, A. N. Cleland & J. M. Martinis. *Measurement of the entanglement of two superconducting qubits via state tomography*. Science, vol. 313, no. 5792, pages 1423–1425, September 2006.
- [Steffen 13] L. Steffen, A. Fedorov, M. Oppliger, Y. Salathe, P. Kurpiers, M. Baur, G. Puebla-Hellmann, C. Eichler & A. Wallraff. *Realization of Deterministic Quantum Teleportation with Solid State Qubits*. ArXiv e-prints, February 2013.
- [Strauch 03] Frederick W. Strauch, Philip R. Johnson, Alex J. Dragt, C. J. Lobb, J. R. Anderson & F. C. Wellstood. *Quantum Logic Gates for Coupled Superconducting Phase Qubits*. Phys. Rev. Lett., vol. 91, no. 16, pages 167005–, October 2003.
- [Wallraff 04] A. Wallraff, D. I. Schuster, A. Blais, L. Frunzio, R.-S. Huang, J. Majer, S. Kumar, S. M. Girvin & R. J. Schoelkopf. *Strong coupling of a single photon to a superconducting qubit using circuit quantum electrodynamics*. Nature, vol. 431, pages 162–167, 2004.
- [Wallraff 07] A. Wallraff, D. I. Schuster, A. Blais, J. M. Gambetta, J. Schreier, L. Frunzio, M. H. Devoret, S. M. Girvin & R. J. Schoelkopf. *Sideband Transitions and Two-Tone Spectroscopy of a Superconducting Qubit Strongly Coupled to an On-Chip Cavity*. Phys. Rev. Lett., vol. 99, page 050501, 2007.
- [Warren 97] Warren S. Warren. *The Usefulness of NMR Quantum Computing*. Science, vol. 277, no. 5332, pages 1688–1690, 1997.