



Eidgenössische Technische Hochschule Zürich  
Swiss Federal Institute of Technology Zurich

SEMESTER THESIS

# Qubit State Population from Linearly Independent System Response

*Author:*

Axel DAHLBERG

*Supervisor:*

Johannes HEINSOO

February 16, 2016



## **Abstract**

In quantum computing, the capability to do accurate operations on the system being used, is an important ingredient. To test the accuracy of these operations, the state of the system has to be determined. This can be accomplished by coupling the system to a resonator, since the resonator frequency of the resonator will depend on the state of the system. The main focus of this project has been to investigate different methods of extracting information about the state of the system from the changes in a signal transmitted through the resonator.

The readout of the state of the system is done by comparing the signal response of the resonator with reference signals. The reference signals are the responses given when the system is set to some specific states. This analysis need to be robust to various errors in the preparation of these specific states. We have investigated two different approaches to compensate for the specific error of thermal excitation of the system.

We tested these readout methods in our experimental setup where we use superconducting qubits coupled to resonators. The results we found fluctuated during different runs, which prevented us to achieve the accuracy needed for readout. We believe that the main reason for this is low frequency noise in our instruments.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Cooper pair box . . . . .	1
1.2	Single qubit rotations . . . . .	2
1.3	Readout and interactions . . . . .	3
1.4	Superconducting chip with four qubits . . . . .	3
<b>2</b>	<b>Theory</b>	<b>5</b>
2.1	Signal quadratures . . . . .	5
2.2	From signal response to populations . . . . .	5
2.2.1	Least squares problem with constraints . . . . .	6
2.3	Extracting reference traces from thermally excited states . . . . .	7
2.3.1	From superposition states . . . . .	8
2.3.2	From single-shot readout . . . . .	8
2.4	State Tomography . . . . .	10
<b>3</b>	<b>Results</b>	<b>11</b>
3.1	Extracting populations from transmission signal . . . . .	11
3.2	From superposition traces . . . . .	12
3.3	From single-shot traces . . . . .	14
<b>4</b>	<b>Conclusion</b>	<b>18</b>
<b>A</b>	<b>Mathematica code</b>	<b>19</b>
A.1	FCNNLS . . . . .	19
A.2	Using JOptimizer in Mathematica . . . . .	21
<b>B</b>	<b>Fluctuations</b>	<b>23</b>
	<b>References</b>	<b>24</b>

# 1 Introduction

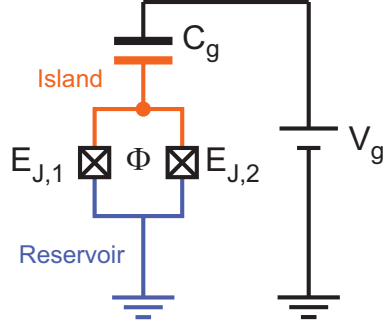
The field of quantum information science has great potential for producing useful applications [Stajic 13] but also to provide deeper understanding of nature [Hensen 15]. Applications include cryptography, computation, simulation and more. But there is still quite some work to be done before we have a universal quantum computer at our hands. To have a more clear goal of what is needed for achieving quantum computing, D. P. DiVincenzo stated five criteria that need to be fulfilled [DiVincenzo 97]. Later he also added two which are required for quantum communication:

- Quantum Computing
  - Well-defined qubits
  - Initialization to a pure state
  - Universal set of quantum gates
  - Qubit-specific measurement
  - Long Coherence times
- Quantum Communication
  - Interconvert stationary and flying qubits
  - Transmission of flying qubits

Many different physical realizations of a qubit are currently being researched. These include trapped ions, NV-centers and superconducting qubits, where the latter is used in our experimental setup. Superconducting qubits have been shown to fulfill the Divincenzo criteria and they make a promising candidate for the building block of quantum computers [Devoret 04]. In the remaining part of this section I will explain how quantum computing is done with superconducting qubits and in particular using transmons. We will not cover all details here but the interested reader can for example refer to [Steffen 13] for a more in depth discussion.

## 1.1 Cooper pair box

A superconducting qubit can be realized by using a Cooper pair box, which is schematically illustrated in Figure 1. In the Cooper pair box a capacitor  $C_g$  and two parallel Josephson junctions  $E_{J,1}$  and  $E_{J,2}$  are connected to a voltage source  $V_g$ . The Josephson junctions consist of two superconducting electrodes with an insulating layer in between. The part in between the capacitor and the Josephson junctions consists of a superconducting island (orange) where electrons pair up and make what are called Cooper pairs [Bardeen 57]. When a voltage is applied over the Josephson junctions these Cooper pairs can tunnel through the insulating layer and there will then be either an excess or a shortage of Cooper pairs on the



**Figure 1:** Circuit illustration with a Cooper pair box consisting of two Josephson junctions with Josephson energies  $E_{J,1}$  and  $E_{J,2}$  which are connected to a voltage source  $V_g$  via a capacitor  $C_g$ . Figure adapted from [Baur 12].

island. Due to the tunneling of Cooper pairs through the Josephson junctions, the junctions will effectively act as nonlinear inductors. The Cooper pair box acts as an artificial atom with a Hamiltonian given by

$$H(n_g) = \sum_{n=-\infty}^{\infty} \left( 4E_C(n - n_g)^2 |n\rangle \langle n| - \frac{E_J}{2} (|n\rangle \langle n+1| + |n+1\rangle \langle n|) \right), \quad (1.1)$$

where  $n$  is the number of excess Cooper pairs on the island,  $n_g = -C_g V_g / (2e)$  is the charge on the gate capacitor in units of  $2e$ ,  $E_J$  is the Josephson energy and  $E_C$  is the charging energy which describe how much energy is needed to add a Cooper pair to the island [Baur 12]. Both  $n_g$  and  $E_J$  can be controlled externally, by varying the gate voltage  $V_g$  and the magnetic flux  $\Phi$  through the cooper pair box and respectively.

## 1.2 Single qubit rotations

By considering only the two lowest energy levels of the Cooper pair box with dynamics described by the Hamiltonian in (1.1), we effectively have a superconducting qubit. To manipulate the state of the qubit we can apply an oscillating voltage on the gate, such that  $V_g = V_g^{\text{DC}} + V_D \cos(\omega_D t)$ . The Hamiltonian of the driven qubit is given by

$$H \approx \frac{1}{2} \omega_{eg} \sigma_z + \Omega \cos(\omega_D t) \sigma_x \quad (1.2)$$

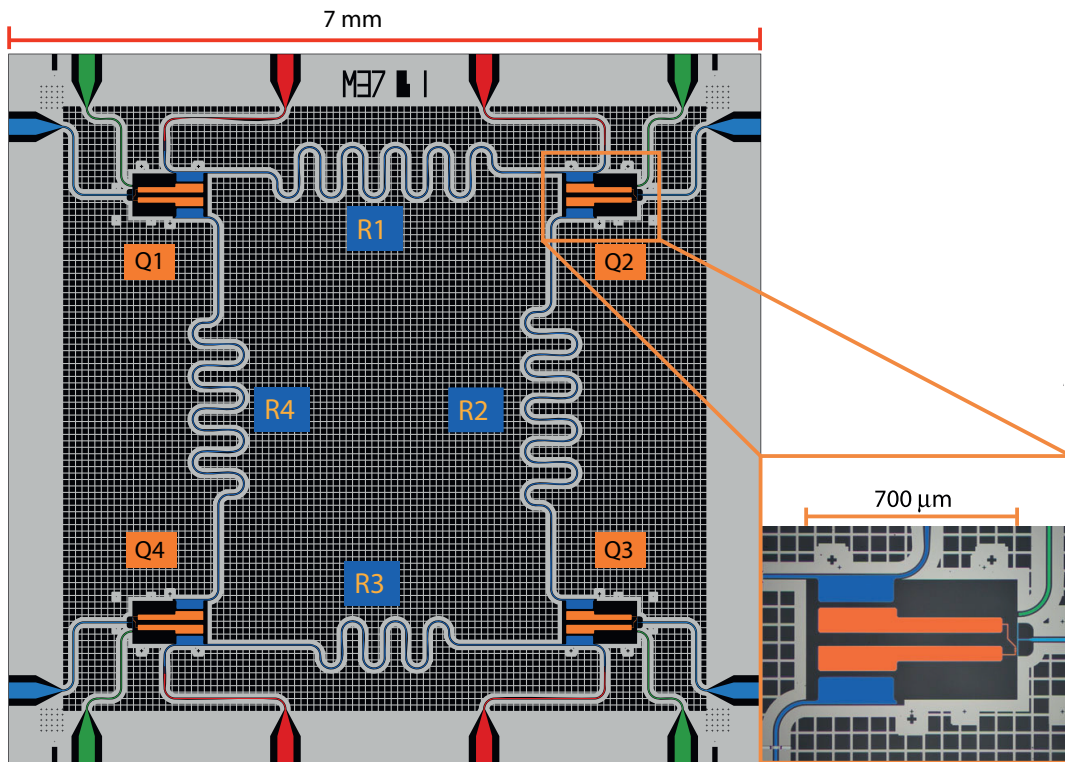
where  $\omega_{eg}$  is the transition frequency between the ground and excited state,  $\Omega$  depends on the amplitude and phase of the drive signal and  $\sigma_z, \sigma_x$  are the Pauli operators [Baur 12]. The state of the qubit can therefore be manipulated by applying voltage-pulses on the gate.

### 1.3 Readout and interactions

The superconducting qubit is coupled to an electromagnetic field in a resonator consisting of a planar waveguide. This is done for a number of reasons. Firstly this allows for non-demolition readout of the state of the qubit. The coupling between the resonator and the qubit will shift the resonance frequency of the resonator, either to a higher or to a lower value, dependent on the state of the qubit. Through spectroscopy on the resonator, the state of the qubit can be determined. Another reason for the resonator is that it allows for interactions between multiple qubits, which can be used for two-qubit manipulations. The resonator can also improve the decay time of the qubit, which determines how long the qubit can stay in the excited state without spontaneously decaying to the ground state. This is because the resonator restricts the possible modes of the electromagnetic field due to its finite length. The qubit is therefore only coupled to a discrete set of modes and does not decay as easily.

### 1.4 Superconducting chip with four qubits

In our setup we use a chip with four superconducting qubits coupled via four resonators as shown in Figure 2. All resonators are used for interactions between the qubits but only resonator R1 and R3 are used for readout by measuring transmission through the red ports. Each qubit is coupled to a flux bias line and a microwave drive line, coloured blue and green respectively. These lines are used to perform operations on the state of the qubits.



**Figure 2:** Four-qubit-chip with resonators coupled to the qubits R1-R4. Transmission and readout of signals through the resonators R1 and R3 are done via the red ports. The blue and green ports are flux bias lines and microwave drive lines used to do operations on the qubits. Figure adapted from [Salathé 15].



## 2 Theory

In this section I describe the theoretical aspect of how information of the state of the qubit can be extracted from signals transmitted through the resonator, which is coupled to the qubit. To extract this information the transmission signal will be decomposed into inphase and quadrature components.

### 2.1 Signal quadratures

Any signal  $s(t)$  can be decomposed as follows

$$s(t) = I(t) \cos(\omega_{\text{IF}}t) + Q(t) \sin(\omega_{\text{IF}}t). \quad (2.1)$$

where the  $I$  and  $Q$  are called the inphase and quadrature components. The quadrature components can be found by multiplying the original signal with  $\cos(\omega_{\text{IF}}t)/2$  and  $\sin(\omega_{\text{IF}}t)/2$  respectively, which gives

$$s(t) \cos(\omega_{\text{IF}}t)/2 = I(t) + I(t) \cos(2\omega_{\text{IF}}t) + Q(t) \sin(2\omega_{\text{IF}}t) \quad (2.2)$$

$$s(t) \sin(\omega_{\text{IF}}t)/2 = Q(t) - Q(t) \cos(2\omega_{\text{IF}}t) + I(t) \sin(2\omega_{\text{IF}}t) \quad (2.3)$$

The factors  $I(t)$  and  $Q(t)$  are found by running the signals in (2.2) and (2.3) through a low pass filter.

### 2.2 From signal response to populations

In this section I explain how the transmission signal quadratures of a resonator coupled to the qubit can be used to determine the state populations of the qubit.

Assume we have a set of eigenstates  $\{|\psi_i\rangle\}_i$ , for example  $\{|g\rangle, |e\rangle, |f\rangle\}$  in the case of a qutrit. These states will give rise to a set of transmission signals which I call the reference traces. In the IQ-plane these reference traces are parametrized by time and I denote them  $\{\vec{\Theta}_{\psi_i}(t)\}_i$ . For an arbitrary state with density matrix  $\rho$ , there will be a set of probabilities  $\{p_{\psi_i}\}_i$  associated with the different measurement outcomes corresponding to the eigenstates  $\{|\psi_i\rangle\}_i$ . This set of probabilities are also commonly called the populations of the state. For an average readout, the trace in the IQ-plane corresponding to the state  $\rho$  will be given by a weighted sum of the reference traces where the coefficients are the populations,

$$\vec{\Theta}_{\rho}(t) = \sum_i p_{\psi_i} \vec{\Theta}_{\psi_i}(t). \quad (2.4)$$

In practice the value of the traces will only be known for a discrete set of time values  $\{t_k\}_k$ . To find the set of populations that solve equation (2.4), both components of the traces at each timepoint can be treated as independent variables. Given  $n$  timepoints, this will give rise to a system of  $2n$  linear equations which usually will be much larger than the number of populations. This overdetermined system of  $2n$  equations can be solved with the method of least squares. To write this system of equation in matrix-form I denote  $\theta_\rho$  as the in-phase components at each timepoint concatenated with the quadrature components. The populations can then be found by solving the following problem

$$Mp' = \theta_\rho \quad \text{for } p'_{\psi_i} \geq 0, \sum_i p'_{\psi_i} = 1, \quad (2.5)$$

which stated as a least squares problem can be written

$$\min_{p'_{\psi_i}} \|Mp' - \theta_\rho\|_2^2 \quad \text{for } p'_{\psi_i} \geq 0, \sum_i p'_{\psi_i} = 1, \quad (2.6)$$

where  $M$  is a matrix which columns are the concatenated reference traces  $\theta_{\psi_i}$  and  $p'$  is a vector with the variables  $p'_{\psi_i}$ . The two constraints of non-negativity and normalization are needed for the solution to be a true probability distribution.

### 2.2.1 Least squares problem with constraints

There has been a number of articles written on methods for solving non-negative least squares, starting with the work of Lawson and Hanson in 1974 [Lawson 74]. In this project we have implemented a method called Fast-Combinatorial-Non-Negative-Least-Squares (FCNNLS), first presented by Benthem and Keenan [Van Benthem 04]. This method is in particular fast when simultaneously solving multiple systems of equations which only differ by their right hand sides. The method have been implemented in Mathematica during this project and the code can be found in appendix A.

What usually is done to solve a least squares problem with a linear equality constraint, as the one for normalization in equation (2.6), is to add a Lagrange multiplier. The system of equations then get one more equation and one more variable. Since the FCNNLS-method forces all variables to be non-negative, including the Lagrange multiplier, this cannot be done in this case. What might be possible is to modify the non-negative least squares method to be able to choose which variables should be under the non-negativity constraint. Instead we can add the normalization equality as an equation to the system of equations with some weight  $w$ . Effectively we add one row to  $M$ , which is  $(w, \dots, w)$  and one element  $w$  in  $\theta_\rho$  at the corresponding position. The weight will normally be 5-10 times the sum of the average

of the absolute value of each row. The problem reduces to

$$\begin{pmatrix} (w, \dots, w) \\ M \end{pmatrix} p' = \begin{pmatrix} w \\ \theta_\rho \end{pmatrix} \quad \text{for } p'_{\psi_i} \geq 0, \quad (2.7)$$

which we can solve with the method FCNNLS.

To check that the results we get by adding a weighted equation are correct, these have been compared with the results from a more powerful optimization function called JOptimizer. This is a package for Java which can be imported by using Mathematicas JLink, also this code can be found in appendix A.

### 2.3 Extracting reference traces from thermally excited states

If the qudit<sup>1</sup> receive thermal excitation from the environment, it is hard to prepare the qudit in its eigenstates. For example if the ground state is intended to be prepared by initially not driving the qudit at all, the state of the qudit will be a mixed state given by

$$\rho_{g^{\text{th}}} = (1 - \sum_{i=1}^{\infty} p_i) |0\rangle \langle 0| + \sum_{i=1}^{\infty} p_i |i\rangle \langle i| \quad (2.8)$$

where  $p_i$  is given by

$$p_i = \exp\left(-\frac{\hbar\omega_{0i}}{k_B T_{\text{eff}}}\right) \quad (2.9)$$

where  $\hbar$  is the reduced Planck constant,  $\omega_{0i}$  is the transition frequency between the energy levels 0 and  $i$ ,  $k_B$  is the Boltzmann constant and  $T_{\text{eff}}$  is the effective temperature. For sufficiently small anharmonicity<sup>2</sup> the transition frequencies are approximately  $\omega_{0i} \approx i\omega_{01}$  so the populations can be estimated by

$$p_i \approx (p_1)^i. \quad (2.10)$$

If the eigenstates cannot be prepared, the reference traces have to be found in some other way rather than using the resonator response from the eigenstates directly. In the sections below we present two possible methods for extracting the reference traces even though there is thermal excitation in the setup.

---

<sup>1</sup>A unit of quantum information in a  $d$ -level quantum system.

<sup>2</sup>Anharmonicity can be defined as  $\omega_{12} - \omega_{01}$ .

### 2.3.1 From superposition states

In this section I describe how superposition states can be used to extract reference traces in the presence of thermal excitation for the case of a qutrit. To suppress effects of noise, we use traces that are the average of normally 60 000 experiment repetitions. From (2.4) and due to thermal excitation our thermally excited ground trace will be given by

$$\theta_{g^{\text{th}}} = (1 - p_e - p_f)\theta_g + p_e\theta_e + p_f\theta_f. \quad (2.11)$$

where the subscripts of the traces denote which state they correspond to. To prepare equal superposition states we can apply the following three operations to the thermally excited ground state

$$X_{ge}(\pi/2) \rightarrow \theta_{g+e} = \frac{(1 - p_f)}{2}(\theta_g + \theta_e) + p_f\theta_f \quad (2.12)$$

$$X_{ef}(\pi)X_{ge}(\pi/2) \rightarrow \theta_{g+f} = \frac{(1 - p_f)}{2}(\theta_g + \theta_f) + p_f\theta_e \quad (2.13)$$

$$X_{ge}(\pi)X_{ef}(\pi)X_{ge}(\pi/2) \rightarrow \theta_{e+f} = \frac{(1 - p_f)}{2}(\theta_e + \theta_f) + p_f\theta_g, \quad (2.14)$$

where  $X_{ij}(\varphi)$  means a rotation by  $\varphi$  around the  $X$ -axis of the Bloch sphere for the energy levels  $i$  and  $j$ . The rotations are written as operators, so the order in which they are applied should be read from right to left in the above notation. Given these traces we can estimate the real ground trace by the following combination

$$\theta_{g+e} + \theta_{g+f} - \theta_{e+f} = \theta_g + p_f(\theta_e + \theta_f - 2\theta_g). \quad (2.15)$$

Since  $p_f$  can be assumed to be very small and in particular a lot smaller than  $p_e$ , this is a much better estimate of the ground trace than the one given in (2.11). The same can be done with the other two eigenstates with the following combinations

$$\theta_{g+e} - \theta_{g+f} + \theta_{e+f} = \theta_e + p_f(\theta_g + \theta_f - 2\theta_e) \quad (2.16)$$

$$-\theta_{g+e} + \theta_{g+f} + \theta_{e+f} = \theta_f + p_f(\theta_g + \theta_e - 2\theta_f), \quad (2.17)$$

for the  $e$  and  $f$  levels correspondingly. Populations obtained from experimental data using this method can be found in section 3.2.

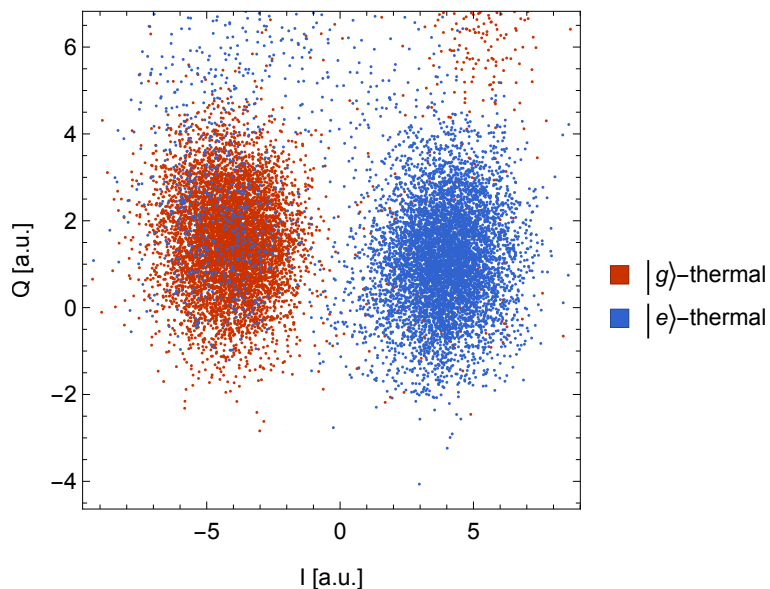
### 2.3.2 From single-shot readout

In a single-shot measurement the state of the qudit gets projected to the eigenstate corresponding to the measurement outcome. The reason why single-shot traces cannot be used

as reference traces directly is that these are in general very noisy. To suppress the noise, the traces could be averaged over many experimental repetitions. The problem is that, for example for a thermally excited ground state, a fraction of the single-shot traces will correspond to an excited state. A solution to this is to determine what eigenstates the single-shot traces correspond to. The traces not corresponding to the eigenstate which reference trace we wish to find, can be removed and not be included in the averaging.

To determine which state a single-shot trace corresponds to, we integrate out time such that the trace gets mapped to a point in the IQ-plane. In the rest of this section we will only consider the ground and first excited state by assuming that  $p_f \ll 1$ .

The integrated traces mainly get mapped to two areas in the IQ-plane, which is due to the projective measurement, see Figure 3. For example, a trace with an integrated point close to the middle of the red area, will with high probability be a ground trace. Therefore the ground and excited traces can be classified and averaged over separately, into reference traces with lower noise.



**Figure 3:** Integrated values for single-shot traces measured for thermally excited ground state in red and after a  $\pi$ -pulse in blue.

The benefit of this method is that it does not matter what state the qubit was in before the measurement. If we find two distinct gatherings of integrated traces we can classify the traces and average over the ones corresponding to the relevant eigenstate. Reference traces extracted using this method on experimental data can be found in section 3.3.

## 2.4 State Tomography

To completely characterize an arbitrary state of a qudit, its density matrix should be determined. The populations of the state in the eigenstates is then not sufficient. For example the populations of a qubit is described by one real number, since the two populations are constrained to be normalized. However, its density matrix should be a normalized hermitian matrix of dimension 2 and therefore described by 3 real numbers.

If we do not assume normalization the density matrix of an arbitrary qubit state can be written,

$$\rho = \sum_{i=0}^3 \alpha_i \sigma_i, \tag{2.18}$$

where  $\sigma_0$  is the identity and  $\sigma_i$  for  $i = 1, 2, 3$  are the Pauli matrices. So to completely characterize the state of a qubit, it is sufficient to determine the coefficients  $\alpha_i$ . These coefficients can be found from the populations of the state itself and the states prepared by initially performing the rotations  $X_{ge}(-\pi/2)$ ,  $Y_{ge}(-\pi/2)$  and  $X_{ge}(\pi)$ . For derivation of this and further discussion on state tomography, refer to for example [Maurer 08].

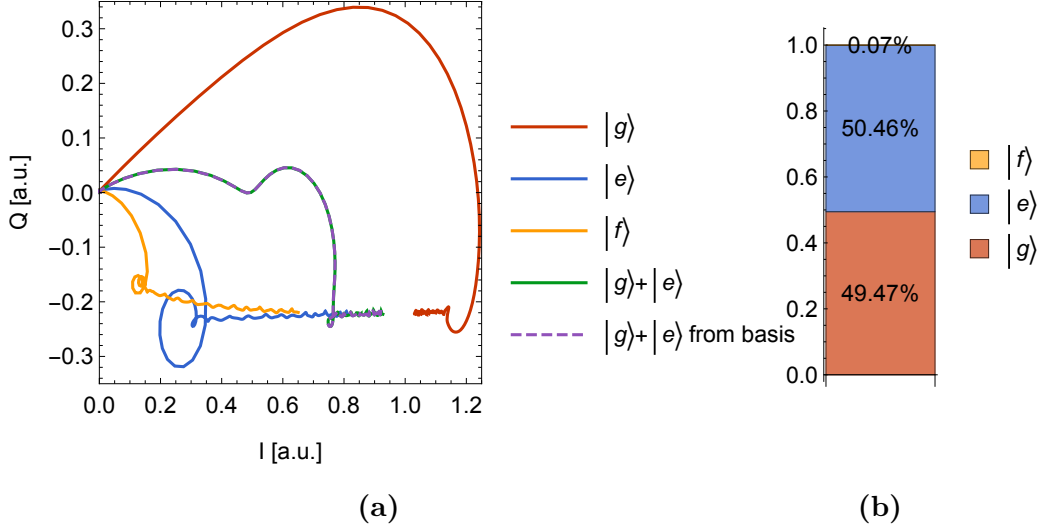
### 3 Results

In this section I present experimental result from applying the methods for extracting populations of the state of a qudit, from signals transmitted through the resonator, introduced in section 2. First I present results for solving equation (2.5) by using NNLS in section 3.1, which gives the populations of a state, if correct reference traces are provided. In section 3.3 I continue with results for extracting reference traces by using superposition states and lastly by using single-shot traces in section 3.3.

#### 3.1 Extracting populations from transmission signal

In section 2.2 and equation (2.4) I stated that the trace from a superposition state will be a combination of the reference traces. The experimental results in Figure 4 agree with this in the case of a qutrit and the superposition state being  $|g\rangle + |e\rangle$ . In the figure we see that all traces start at the origin when the measurement begins. This means that there is no transmission signal. When the measurement starts, the quadrature points move away from the origin and draw curvy traces. These shapes show the characteristic dynamical features of a driven harmonic oscillator. After some time the traces settle and more slowly approach a common point, this is the squiggly part of the traces in the figure. This approach to a common point is due to the decay, from the excited states to the ground, of the qubit.

The purple trace, which is a combination of the reference traces as in equation (2.4) with the populations given in Figure 4b, completely covers the green which is the superposition trace found experimentally. A decomposition this good is impossible for most curves in the plane. Therefore we can be confident that our assumptions in section 2.2 are correct. In this particular case I have not taken into account thermal excitation, which is not necessary for the equal superposition state. This is because this state is symmetric in the two relevant eigenstates. Furthermore, the prepared  $e$ -state is the thermally excited ground state with the populations in the  $g$  and  $e$  levels swapped. Therefore the decomposition of the equal superposition state will give populations  $(p_g, p_e, p_f) \approx (0.5, 0.5, 0)$ , independently of the thermal excitation in the reference traces.



**Figure 4:** Quadrature components of a signal transmitted through the resonator coupled to the transmon when this was in four different states (a). The dashed trace is a combination of the traces from the eigenstates using the populations in (b), which are found by decomposing the equal superposition trace (green) in terms of the traces from the eigenstates.

### 3.2 From superposition traces

In this section I present experimental results from using the method presented in section 2.3.1 to extract reference traces corresponding to the eigenstates. In Figure 5, traces from the three first thermally excited energy levels of the transmon are shown. We have used traces from prepared superposition states and the method in section 2.3.1 to extract reference traces, shown as dashed traces in Figure 5. These extracted traces are plotted in the same figure. These traces also show the same dynamics as discussed in the previous section. Specifically they have the dynamics of a driven resonator coupled to a system that decays.

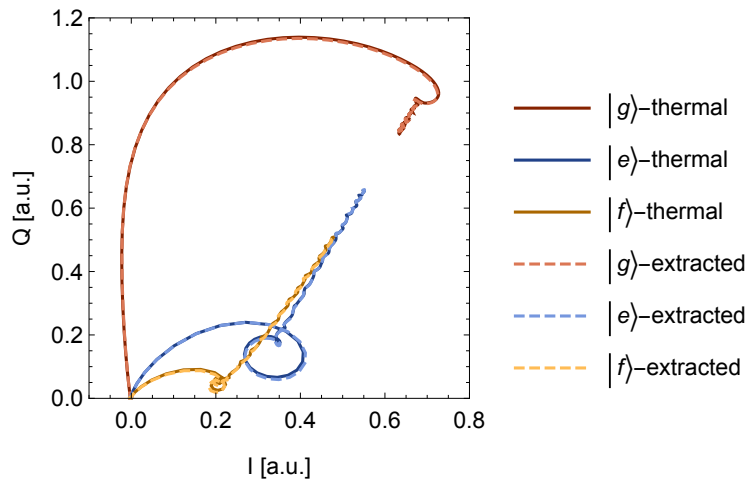
When we used this method on experimental data we found two problems. One of the problems can be seen in Figure 5: the trace from the thermally excited ground state is just outside the extracted trace. This should not be the case. The trace from an arbitrary state should be given as a combination of the reference traces, with non-negative coefficients which sum up to one<sup>3</sup>. The  $|g\rangle$ -thermal trace does not satisfy these constraints, if the extracted traces are used as reference traces. The second problem was that the traces fluctuated almost on the order of the thermal population during different runs, even though we made large amount of averages.

<sup>3</sup>Populations in the energy levels above the first three can be neglected, at the temperature and accuracy we currently have.



The extracted trace was sometimes outside the trace from the thermally excited ground and sometimes inside. Figure 12 in appendix B illustrate these fluctuations by showing the traces from nine experimental repetitions.

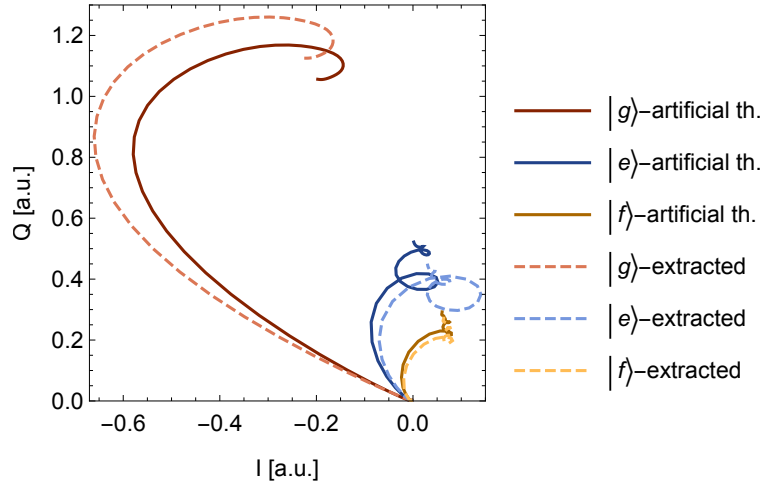
Finding the reason to why we see these inconsistencies, is still an unsolved problem. There are a couple of possible issues in our experimental setup that could give rise to these faulty outcomes and which need to be investigated. Firstly the single qubit rotations we use to prepare our states might lack the needed accuracy. Since our method crucially depend on that we prepare equal superposition states, inaccurate single qubit rotations can lead to a problem. Another problem might be that we have low frequency noise in the instruments we use in our experimental setup. The frequency might be such that the noise matters on the timescale we use for averaging but is perhaps not low enough to be compensated by calibration measurements.



**Figure 5:** Solid curves show quadrature components of signals transmitted through the resonator coupled to the transmon for the first three energy levels in presence of thermal excitation. The dashed curves are extracted traces from superposition traces.

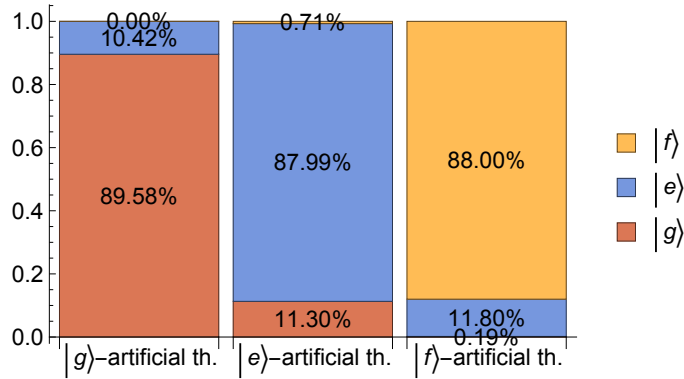
Since we had these inconsistencies we wanted to try our method in a setting where the accuracy was not as crucial. What we did was to essentially introduce artificial thermal excitation by initially rotating our states by  $\frac{\pi}{5}$  around the  $Y$ -axis of the Bloch sphere. Because we prepared all states by performing rotations around the  $X$ -axis, this initial rotation around the  $Y$ -axis essentially acts as  $\sin^2(\pi/10) \approx 9.5\%$  extra thermal excitation. In Figure 6 we show the traces we measure when we apply this artificial thermal excitation plus the traces we extract from superposition states. By using the extracted as references traces in our

analysis we can calculate the populations of the artificial thermal traces in terms of these reference traces.



**Figure 6:** Same setting as in Figure 5 but during all state preparations an initial  $\pi/5$ -pulse around the  $Y$ -axis was applied.

The computed total thermal excitation was found to be approximately 11 %, see Figure 7. The total thermal excitation should be the real thermal excitation plus the artificial and since we expect the real thermal excitation to be 5 – 6 %, what we found is too low.

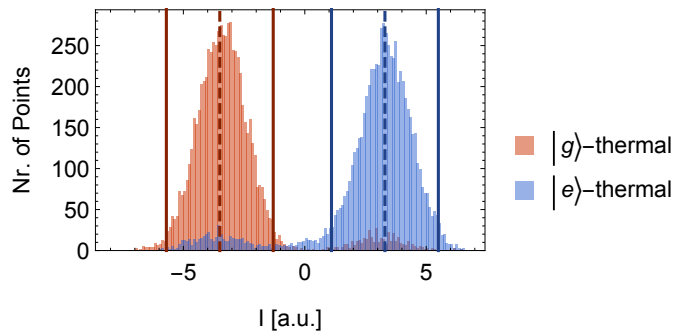


**Figure 7:** Populations of the three solid traces in Figure 6 in terms of the three dashed.

### 3.3 From single-shot traces

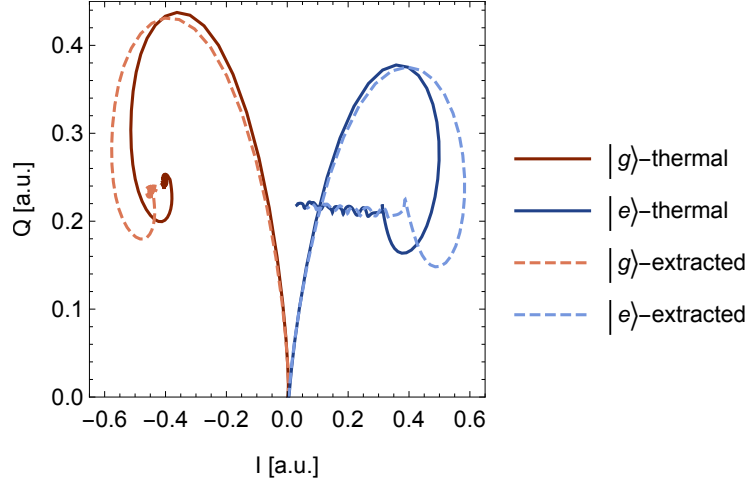
Here I present experimental results from using the method presented in section 2.3.2 to extract reference traces corresponding to the eigenstates. In Figure 3 we saw that the points

from the integrated traces gather in two regions in the IQ-plane which correspond to getting a measurement outcome  $g$  or  $e$  respectively. The coordinate system in the IQ-plane has been rotated such that the separation between these regions is mainly in one quadrature. This allow us to look at these points in a histogram over this quadrature, as shown in Figure 8. The red histogram is bimodal with a higher peak to the left and a smaller to the right. The smaller is due to the thermal excitation. We can see a similar shape for the blue histogram but mirrored. This is because the  $|e\rangle$ -thermal state is prepared by swapping the  $p_g$  and  $p_e$  populations of the thermally excited ground state. What also can be seen for the blue histogram is a slight tail on the left side of the higher peak. This is due to decay from the excited state to the ground. In the histogram we have also specified two intervals in red and blue. Traces with an integrated point inside for example the red interval are with high probability traces corresponding to the eigenstate  $|g\rangle$  and the same for the blue interval and the state  $|e\rangle$ .



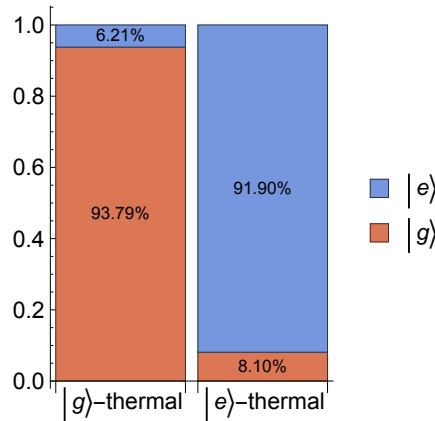
**Figure 8:** Histogram of the data points in Figure 3 along the I quadrature.

In Figure 9 I show the traces we find when we average over the traces in the red and blue regions in Figure 8 respectively. Also included in Figure 9 are the traces from the thermally excited  $g$  and  $e$  states. Notice that the  $|g\rangle$ -extracted trace is further from the  $|e\rangle$  traces than the  $|g\rangle$ -thermal. This is what we expect and the problem we had with the data in Figure 5 regarding this, is not present here. Another thing to note is that the  $|e\rangle$ -extracted trace will not compensate for decay. The reason for this is that we do not include the tail of the blue histogram in our averaging.



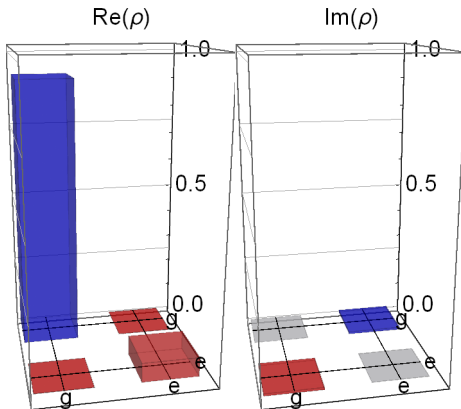
**Figure 9:** Quadrature components of averaged traces. The solid curves are the average of the red and blue sets of traces in Figure 8 respectively. The dashed are the same but when only using the traces inside the red and blue intervals in Figure 8.

Since the  $|e\rangle$ -extracted trace does not compensate for decay, we will be able to see the decay when we compute the populations of a state. This can be seen in Figure 10, where we used the extracted traces in Figure 9 as reference traces to calculate the populations of the thermally excited states. The decay can be seen by comparing the  $p_e$  population of the  $|g\rangle$ -thermal state with the  $p_g$  population of the  $|e\rangle$ -thermal state. Since their difference is  $8.10\% - 6.21\% = 1.89\%$ , we can see that we roughly have a 2% decay during our measurement.



**Figure 10:** Populations of the two solid traces in Figure 9 in terms of the two dashed.

As discussed in section 2.4, we need to find the density matrix to fully characterize a state. To do this, we need the populations in the eigenstates for the state itself and for the states we get by performing the following rotations  $X_{ge}(-\pi/2)$ ,  $Y_{ge}(-\pi/2)$  and  $X_{ge}(\pi)$ . We have done this for the thermally excited ground state by decomposing the traces from the prepared states in terms of the extracted reference traces in Figure 9. Using these populations we could calculate the density matrix, which is illustrated in Figure 11.



**Figure 11:** Density matrix of the thermally excited ground state of the qubit.

In matrix-form the density matrix was found to be,

$$\rho \approx \begin{pmatrix} 0.922 & 0.0 \\ 0.0 & 0.067 \end{pmatrix} + i \begin{pmatrix} 0.0 & -0.008 \\ 0.008 & 0.0 \end{pmatrix}. \quad (3.1)$$

The density matrix we found is not normalized, that is  $Tr(\rho) < 1$ . This is because the population in the  $g$ -state of the thermally excited  $e$ -state is larger than the population in the  $e$ -state of the thermally excited  $g$ -state. Which in turn is due to decay, as pointed out earlier in this section.

## 4 Conclusion

In this project we have investigated how to extract state populations of a qubit from signals transmitted through a resonator coupled to a qubit. We have looked at the quadratures of the signal and used linear algebra to calculate the populations. Our method works well on experimental data if one has access to correct reference traces, that is if we can find the traces corresponding to the eigenstates. The two methods we have proposed to find correct reference traces work theoretically but when we use them on experimental data we found some problems and in particular a lack of sufficient accuracy. For the method where we extract reference traces from superposition traces, we could clearly see fluctuations between experimental repetitions. We did not however, have the same amount of data for the method using single-shot traces, due to lack of time. This prevented us from investigate if fluctuations is also a problem for single-shot method. Although we suspect this to be the case, it is something that should be studied further. We believe that the origin of these problems might be due to inaccurate single qubit rotations or low frequency noise in our instruments. In future projects these problems should be investigated further, since these inconsistencies between theory and experiment might be important in other aspects of our experimental setup.

## Acknowledgements

I would like to thank Prof. Andreas Wallraff for giving me the opportunity to work on this project. Furthermore I would also like to thank Johannes Heinsoo for supervising me with great knowledge, humour and patience. Lastly I wish to thank Paul Magnard, Theodore Walter, Mintu Mondal and Samuel Haberthür for interesting discussions and all the help.

# A Mathematica code

## A.1 FCNNLS

The Mathematica function  $fcnnls(M, A)$  (line 1-84) returns a matrix  $X$ . The  $i$ -th column of  $X$  are the non-negative least-squares solution to the equation  $Mx = a_i$ , where  $a_i$  is the  $i$ -th column of  $A$ . The function  $fcnnls$  uses a subfunction  $cssls$  (line 86-113) also presented below. Both functions have been translated into Mathematica code from the MATLAB code presented in a paper by Benthem and Keenan [Van Benthem 04].

```
1 fcnnls[M_, A_] /; Dimensions[M,1]==Dimensions[A,1] :=
2 Module[{nObs, lVar, pRHS, W, iter, maxiter, CtC, CtA, K, Pset,
3 Psetindex, D, Fset, Hset, nHset, alpha, i, j, hIdx, negIdx,
4 alphamin, minIdx, idx2zero, Jset, mx, mxidx},
5
6 {nObs, lVar} = Dimensions[M];
7 pRHS = Dimensions[A][[2]];
8 W = ConstantArray[0, {lVar, pRHS}];
9 iter = 0; maxiter = 3 lVar;
10
11 (*Precompute parts of pseudoinverse*)
12 CtC = M\[Transpose].M; CtA = M\[Transpose].A;
13
14 (*Obtain the initial feasible solution and corresponding passive set*)
15 K = cssls[CtC, CtA, {}];
16 Psetindex = Position[K, _?Positive];
17 Pset = Normal[SparseArray[Psetindex -> 1]];
18 K = ReplacePart[K, Position[K, _?NonPositive] -> 0];
19 D = K;
20 Fset = Position[FreeQ[#, 0] & /@ (Pset\[Transpose]), False] // Flatten;
21
22 (*Active set algorithm for NLS main loop*)
23 While[Length[Fset] > 0,
24 (*Solve for the passive variables (uses subroutine below)*)
25 K[{:; , Fset}] = cssls[CtC, CtA[{:; , Fset}], Pset[{:; , Fset}]];
26
27 (*Find any infeasible solutions*)
28 Hset = Fset[[Position[MemberQ[#, _?Negative] & /@ (K[{:; , Fset])\[Transpose]), True] // Flatten]];
29
30 If[Length[Hset] > 0,
31 nHset = Length[Hset];
32 alpha = ConstantArray[0, {lVar, nHset}];
33
34 While[Length[Hset] > 0 && (iter < maxiter),
35 iter++;
36 alpha[{:; , ;}] = Infinity;
37
38 (*Find indices of negative variables in passive set*)
39 hIdx = Position[Pset[{:; , Hset}], 1] \[Intersection] Position[K[{:; , Hset}], _?Negative];
40 negIdx = Join[{hIdx[{:; , 1}], {Hset[hIdx[{:; , 2}]]}]\[Transpose];
41 Table[
42 alpha[[hIdx[[i, 1]], hIdx[[i, 2]]]] =
43 If[(D[[#[[1]], #[[2]]] - K[[#[[1]], #[[2]]]]) &[
44 negIdx[[1]]] == 0,
45 Infinity, (D[[#[[1]], #[[2]]])/(D[[#[[1]], #[[2]]] -
46 K[[#[[1]], #[[2]]]]) &[negIdx[[i]]];, {i,
47 Length[negIdx]};
48 {alphamin, minIdx} = Table[{Min[alpha[{:; , i}], Position[alpha[{:; , i],
49 ]][[1, 1]]], {i, nHset}}\[Transpose];
50 alpha = Table[alphamin, {i, lVar}];
51 D[{:; , Hset}] = D[{:; , Hset}] - alpha*(D[{:; , Hset}] - K[{:; , Hset}]);
52 idx2zero = Join[{minIdx}, {Hset}\[Transpose];
53 D = ReplacePart[D, idx2zero -> 0];
54 Pset = ReplacePart[Pset, idx2zero -> 0];
55 K[{:; , Hset}] = cssls[CtC, CtA[{:; , Hset}], Pset[{:; , Hset}]]];
```

```

55         Hset = Position[MemberQ[#, _?Negative] & /@ (K\[Transpose]), True] // Flatten;
56         nHset = Length[Hset];
57     ];
58 ];
59 (*Make sure the solution has converged*)
60 If[iter == maxiter, (*Error*)iter];];
61
62 (*Check solutions for optimality*)
63 W[{:; , Fset]] = CtA[{:; , Fset]] - CtC.K[{:; , Fset]];
64 Jset =
65     Position[
66         FreeQ[#,
67             False] & /@ (Map[# <= 0 &,
68                 Boole[Map[# == 0 &, Pset[{:; , Fset]], {-1}]]*
69                     W[{:; , Fset]], {-1}]\[Transpose]), True] // Flatten;
70 Fset = Complement[Fset, Fset[[Jset]]];
71
72 (*For non-optimal solutions, add the appropriate variable to Pset*)
73
74 If[Length[Fset] > 0,
75     mx = Boole[Map[# == 0 &, Pset[{:; , Fset]], {-1}]]*W[{:; , Fset]];
76     mxidx =
77         Table[Position[mx[{:; , i}], Max[mx[{:; , i}]]][[1]], {i,
78             Dimensions[mx][[2]]}] // Flatten;
79     Pset = ReplacePart[Pset, Join[{mxidx}, {Fset}]\[Transpose] -> 1];
80     D[{:; , Fset]} = K[{:; , Fset}];
81 ];
82 ];
83 K
84 ];
85
86 cssls[CtC_List, CtA_List, Pset_List] :=
87 Module[{K, lVar, pRHS, codedPset, sortedPset, sortedEset, breaks,
88     breakIdx, cols2solve, vars, varsindex},
89
90     K = ConstantArray[0, Dimensions[CtA]];
91
92     If[Pset == {} || FreeQ[Pset, 0],
93
94         K = Inverse[CtC].CtA;,
95
96         {lVar, pRHS} = Dimensions[Pset];
97         codedPset = 2^Range[lVar - 1, 0, -1].Pset;
98         sortedPset = Sort[codedPset];
99         sortedEset = Ordering[codedPset];
100        breaks = Differences[sortedPset];
101        breakIdx = Join[{0}, Position[breaks, _?Positive] // Flatten, {pRHS}];
102
103        Do[
104            cols2solve = sortedEset[[Range[breakIdx[[k]] + 1, breakIdx[[k + 1]]]];
105            vars = Pset[{:; , sortedEset[[breakIdx[[k]] + 1]]];
106            varsindex = Position[vars, 1] // Flatten;
107            K[[varsindex, cols2solve]] = PseudoInverse[CtC[[varsindex, varsindex]].CtA[[varsindex,
108                cols2solve]];
109            , {k, Length[breakIdx] - 1}];
110        ];
111
112        K
113    ];

```



## A.2 Using JOptimizer in Mathematica

The Mathematica function `nonnegLSJava(M, b)` (line 51-66) returns a vector which is the least-squares solution to the equation  $Mx = b$ . Options can be given to the function such that the solution is both non-negative and normalized by setting `positive -> True` and `normalized -> True`. Normalized means that the elements of the solution sum up to one,  $\sum_i x_i = 1$ . The function `nonnegLSJava` transform the least-squares problem into a quadratic programming problem and then calls a function `qpJava` (line 11-47) to solve this. The function `qpJava` utilizes a Java-package called JOptimizer<sup>4</sup>. To use Java-packages these have to be implemented to Mathematica through the JLink (line 1-7). Furthermore, the JOptimizer uses four other Java-packages called commons-math, commons-logging, commons-lang and Colt.

```

1 Needs["JLink"];
2 InstallJava[];
3 Javapackagesdir = "Q:\\USERS\\Axel\\Java\\Javapackages";
4 files = {"\\joptimizer-3.4.0.jar", "\\commons-math3-3.5.jar",
5 "\\commons-logging-1.2.jar", "\\colt.jar",
6 "\\commons-lang3-3.4.jar"};
7 AddToClassPath[StringJoin[Javapackagesdir, #]] & /@ files;
8
9 Options[qpJava] = {tolFeas -> 10.^-12, tol -> 10.^-12,
10  positiv -> True, normalized -> True};
11 qpJava[P_, q_, Aeq_, beq_, Gineq_, cineq_, initPoint_,
12  OptionsPattern[]] := JavaBlock[
13  Module[{nrOfVar, objectfunc, or, ineq, opt},
14  nrOfVar = Dimensions[P][[2]];
15
16  objectfunc =
17  JavaNew["com.joptimizer.functions.\
18  PDQuadraticMultivariateRealFunction", P, q, 0];
19
20  or = JavaNew["com.joptimizer.optimizers.OptimizationRequest"];
21  or@setF0[objectfunc];
22  or@setInitialPoint[initPoint];
23  If[OptionValue[normalized],
24  or@setA[Aeq];
25  or@setB[beq];
26  ];
27  or@setToleranceFeas[OptionValue[tolFeas]];
28  or@setTolerance[OptionValue[tol]];
29  If[OptionValue[positiv],
30  LoadJavaClass["java.lang.reflect.Array"];
31  ineq =
32  JavaNew["Lcom.joptimizer.functions.\
33  LinearMultivariateRealFunction;", nrOfVar];
34  Do[Array[set[ineq, i - 1,
35  JavaNew["com.joptimizer.functions.\
36  LinearMultivariateRealFunction", Gineq[[i]], cineq[[i]]]]; {i,
37  nrOfVar}];
38  or@setFi[ineq];
39  ];
40
41  opt = JavaNew["com.joptimizer.optimizers.JOptimizer"];
42  opt@setOptimizationRequest[or];
43  opt@optimize[];
44
45  opt@getOptimizationResponse[] @getSolution[]
46  ]
47 ]
48
49 Options[nonnegLSJava] = {tolFeas -> 10.^-12, tol -> 10.^-12,

```

---

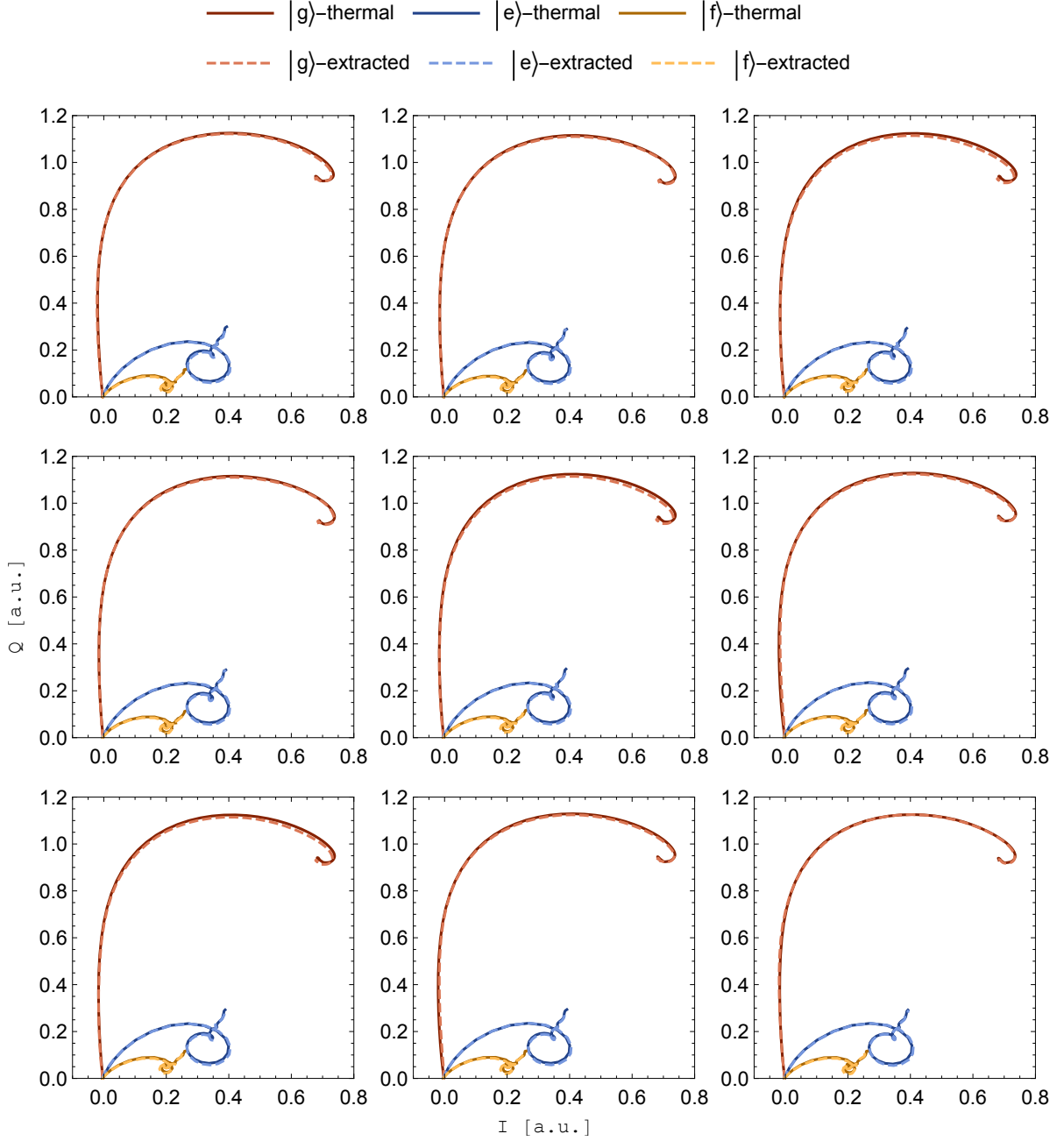
<sup>4</sup>Documentation about JOptimizer can be found under <http://www.joptimizer.com/>

```

50     positiv -> True, normalized -> True};
51 nonnegLSJava[M_, b_, OptionsPattern[]] :=
52 Module[{P, q, nrOfVar, Aeq, beq, Gineq, cineq, initPoint},
53   P = 2 M\[Transpose].M // N;
54   q = -2 M\[Transpose].b // N;
55   nrOfVar = Dimensions[M][[2]];
56   Aeq = ConstantArray[1, {1, nrOfVar}] // N;
57   beq = {1} // N;
58   Gineq = -IdentityMatrix[nrOfVar] // N;
59   cineq = ConstantArray[0, nrOfVar] // N;
60   initPoint = ConstantArray[1/nrOfVar, nrOfVar] // N;
61
62   qpJava[P, q, Aeq, beq, Gineq, cineq, initPoint,
63     tolFeas -> OptionValue[tolFeas], tol -> OptionValue[tol],
64     positiv -> OptionValue[positiv],
65     normalized -> OptionValue[normalized]]
66 ]

```

## B Fluctuations



**Figure 12:** Nine experimental repetitions to illustrate fluctuating results. Solid curves show quadrature components of signals transmitted through the resonator coupled to the transmon for the first three energy levels in presence of thermal excitation. The dashed curves are extracted traces from superposition traces.

## References

- [Bardeen 57] J. Bardeen, L. N. Cooper & J. R. Schrieffer. *Theory of superconductivity*. Phys. Rev., vol. 108, pages 1175–1204, 1957. 1
- [Baur 12] Matthias Baur. *Realizing quantum gates and algorithms with three superconducting qubits*. PhD thesis, ETH Zurich, 03 2012. 2
- [Devoret 04] M. H. Devoret, A. Wallraff & J. M. Martinis. *Superconducting Qubits: A Short Review*. cond-mat/0411174, 2004. 1
- [DiVincenzo 97] D. P. DiVincenzo. *Topics in Quantum Computers*. Mesoscopic Electron Transport (NATO Advanced Study Institute, Series E: Applied Sciences), vol. 345, 1997. 1
- [Hensen 15] B. Hensen, H. Bernien, A. E. Dreau, A. Reiserer, N. Kalb, M. S. Blok, J. Ruitenberg, R. F. L. Vermeulen, R. N. Schouten, C. Abellan, W. Amaya, V. Pruneri, M. W. Mitchell, M. Markham, D. J. Twitchen, D. Elkouss, S. Wehner, T. H. Taminiau & R. Hanson. *Loophole-free Bell inequality violation using electron spins separated by 1.3 kilometres*. Nature, vol. 526, pages 682–686, oct 2015. 1
- [Lawson 74] C. L. Lawson & R. J. Hanson. Solving least squares problems. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 1974. 6
- [Maurer 08] Peter Maurer. State tomography of multiple qubits in circuit qed. Master’s thesis, ETH Zurich, 07 2008. 10
- [Salathé 15] Y. Salathé, M. Mondal, M. Oppliger, J. Heinsoo, P. Kurpiers, A. Potočnik, A. Mezzacapo, U. Las Heras, L. Lamata, E. Solano, S. Filipp & A. Wallraff. *Digital Quantum Simulation of Spin Models with Circuit Quantum Electrodynamics*. Phys. Rev. X, vol. 5, page 021027, 06 2015. 4
- [Stajic 13] J. Stajic. *The Future of Quantum Information Processing*. Science, vol. 339, page 1163, 2013. 1
- [Steffen 13] Lars Steffen. *Quantum Teleportation and Efficient Process Verification with Superconducting Circuits*. PhD thesis, ETH Zurich, 2013. 1

[Van Benthem 04] M. H. Van Benthem & M. R. Keenan. *Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems*. Journal of Chemometrics, vol. 18, no. 10, pages 441–450, 2004. 6, 19



Eidgenössische Technische Hochschule Zürich  
Swiss Federal Institute of Technology Zurich

## Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

**Title of work** (in block letters):

Qubit State Population from Linearly Independent System Response

**Authored by** (in block letters):

*For papers written by groups the names of all authors are required.*

**Name(s):**

Dahlberg

**First name(s):**

Axel

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

**Place, date**

Zürich, 12.02.2016

**Signature(s)**

*For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.*