

Simon Dörrer
doerrers@ethz.ch

Impact of RF Noise on Trapped-ion Quantum Gate Fidelity

Master Thesis

Trapped-Ion Quantum Information Group
Swiss Federal Institute of Technology (ETH) Zurich

Supervisors:

Martin Stadler

Professor:

Dr Jonathan Home

February 21, 2024

Abstract

As quantum computers scale up to eventually reach a quantum advantage, their control systems have to scale accordingly. Therefore, it is important to avoid overspecification of hardware to be cost-effective. This thesis investigates the relationship between the phase noise of the RF-generation hardware and the gate fidelity of trapped-ion qubits, such that an appropriate phase noise specification can be derived. To achieve this goal, this work contains both theoretical analysis and simulations, as well as experimental evidence. I simulated both single- and multi-qubit quantum gates under noise influence. In order to precisely control the phase noise present in the experimental setup, I added a noise generator to the current control firmware. It enables the synthesis of signals with defined noise power and profile. This thesis is a collaboration with Texas Instruments. They provided us with a high-performance, wide-bandwidth multi-channel RF transceiver board, the AFE8000EVM. I integrated it into the existing hardware as the analogue front end. Finally, we measured the influence of noise on single-qubit operations on $^{40}\text{Ca}^+$ ions utilising the implemented noise generator. I find a linear dependence of the gate fidelity on the phase noise at the Rabi (trap) frequency for single (multi)-qubit gates.

Contents

1	Introduction	1
2	Theory	3
2.1	Trapped-ion qubit control	3
2.1.1	$^{40}\text{Ca}^+$ as a qubit	3
2.1.2	Ion-light interaction	5
2.1.3	Single-qubit rotations	6
2.1.4	Multi-qubit interaction with the MS gate	7
2.2	Impact of noise on trapped-ion qubits	9
2.2.1	Phase noise	11
2.2.2	Phase noise in single-qubit rotations	14
3	Quantum Experiment Control at TIQI	21
3.1	Programmable hardware	21
3.2	Hardware description languages (HDLs)	22
3.3	Standardised interface for data converters (JESD204)	22
3.4	The experimental setup	24
3.5	The control system	25
4	Implementation	28
4.1	DAC upgrade - AFE8000	28
4.2	Noise generation	29
4.3	On-board complex sine-wave synthesis	34
4.4	Clock generation and distribution	36
5	Results	39
5.1	Phase noise measurements	39
5.2	Simulation	40
5.2.1	Single-qubit gate	41
5.2.2	Mølmer-Sørensen (MS) gate	44
5.3	Gate fidelity measurements	46
5.3.1	Single-qubit rotations	46
6	Discussion	51
6.1	Comparison to other papers	52
6.2	Possible further investigation	53
	Appendices	61

Chapter 1

Introduction

The idea of utilising quantum physics for computation was first conceived by Paul Benioff, Yuri Manin and Richard Feynman in the early 1980s [1, 2, 3]. They conjectured that using the quantum nature might be more efficient to simulate complex quantum systems than classical computers. The first quantum algorithm proposed by David Deutsch confirmed that [4]. This advantage comes from the way quantum computers process information. While a classical computer stores its computational state in a bit with states 0 and 1, a qubit, the quantum equivalent, is the linear combination of two quantum states $|0\rangle$ and $|1\rangle$. On measurement, either one or the other is measured with a certain probability. It is defined by the probability amplitude, which is the coefficient of the state in the linear combination. Quantum algorithms work by increasing the probability amplitude of the state representing the correct solution by applying a sequence of quantum gates. Thus, the solution to the problem can be measured with high probability in the end. In some cases, these quantum algorithms require fewer operations than their classical counterpart, even up to an exponential speedup. Some examples are Grover's algorithm [5] for unstructured search or most prominently, Shor's algorithm [6] for integer factorisation.

However, a big challenge for quantum computers to overcome is the limited "accuracy" or fidelity of aforementioned quantum gates. Although measurement results are binary, the probability amplitudes are complex numbers. Therefore, quantum gates are more like operations on an analogue computer than digital operations, suffering from small disturbances or imperfect gate operations. Since these minor errors propagate and add up over the several applied gates, it is important to keep them as low as possible to get a meaningful output. Currently, quantum computing is in the noise intermediate-scale quantum (NISQ) era [7]. Thus, we are considerably limited by the error rates of current quantum systems, which are not capable of correcting these errors yet.

Another major obstacle is the number of qubits in a system. Since the gate time for classical computers is in the range of nanoseconds or even picoseconds compared to a gate time of microseconds in the case of trapped-ion quantum computing, and classical bits are much easier to realise than qubits, classical computers still outperform quantum computers. Therefore it is important to scale up the number of qubits such that quantum computers can take advantage of the speedup they promise. This poses a challenge, however, since with a rising number of qubits, error rates increase. The system couples more to the environment and loses its quantum state (it decoheres) and it becomes harder to keep up the same gate fidelity. Also, the cost per qubit has to go down so that such a big system is even economical.

It is important to know the influence of different parts of the system on the final gate fidelity, such that the focus can be set on the most influential aspects. This is especially important with an increasing amount of qubits in order to spend resources per qubit most efficiently. In this thesis, I focus on the influence of control electronics, which all quantum information processing relies on. In the case of trapped ions, the control system is essentially a well-orchestrated set of signal generators. These electrical pulses in the radio frequency (RF) domain control lasers, which in turn control the quantum state of the qubits. All electrical signals however are subject to noise, which impacts the gate fidelity of the qubits negatively as will be shown here. One can aim to reduce this noise as much as possible, but only at the cost of complexity and therefore size and price. Also, at some point the fidelity will no longer be limited by the noise introduced by control hardware, but by other error sources. One has to find a balance between the gate fidelity increase and less complex and affordable and therefore scalable hardware.

The goal of this thesis is to derive a qualitative and quantitative dependency of the fidelity of trapped-ion gates on the noise introduced by control hardware. Some prior studies [8, 9, 10] have been conducted on which this thesis will build. This includes mathematical derivations of the influence of noise, as well as simulations and measurements thereof. However, all of these papers focus on intrinsic laser noise, not noise introduced by control electronics in particular. Therefore, I present noise requirements that help specify an RF generation platform for trapped-ion quantum information experiments. In return, scientists working on the improvement of gate fidelities get a picture of when they are limited by RF noise.

To achieve this goal, this work contains both theoretical analysis and simulations, as well as experimental evidence thereof. The theoretical background of trapped-ion quantum control and derivations of the influence of RF phase noise on single-qubit gates can be found in chapter 2. Based on this, I simulated both single and multi-qubit quantum gates under noise influence in the programming language Julia [11] with the quantum simulation package *QuantumOptics.jl* [12]. To confirm the results retrieved from both theory and simulation, we measured the influence of noise on $^{40}\text{Ca}^+$ ions. For this, I implemented a noise generator in the current control firmware, such that signals with defined noise power and shape can be synthesised directly. To convert the digital signal to electrical pulses, Texas Instruments has provided us with an RF transceiver card, the AFES000EVM. I integrated this card into the existing hardware as part of the thesis. The current hardware platform and technical knowledge needed for this implementation are explained in chapter 3. Chapter 4 covers the realisation of the aforementioned integrated noise generation and adaption of the TI card, together with accompanying changes to hardware and firmware. The final simulation and measurement results are then presented in chapter 5.

Chapter 2

Theory

This chapter covers the theory in quantum optics needed to understand quantum information experiments with trapped ions. Furthermore, I explain the fundamentals of noise and its sources.

2.1 Trapped-ion qubit control

Every quantum computer needs physical qubits at heart [13]. In trapped-ion quantum computing, this physical realisation takes the form of single ions. A notable feature of such an elemental system is that an atom is the same everywhere in the world. Atoms from the alkaline earth metal group are often used since, after ionization, a relatively simple one-electron-like ion is left [14]. Although it is not a qubit in itself, since the remaining valence electron does not only have two electronic states, two states can be defined as our two qubit states $|0\rangle$ and $|1\rangle$. Other states can then assist as auxiliary states for state preparation through optical pumping or state-dependent readout. The atom is ionised such that the charged particle experiences a force in an electric field and can therefore be held in place, or trapped, by a well-crafted electromagnetic potential.

To control the electronic states, or spin, of the ion, we use coherent laser light. A laser can interact with the ion and induce transitions between spin states. In the following sections, we will derive the Hamiltonian for this interaction and subsequently see how we can use it to create single- and multi-qubit gates.

2.1.1 $^{40}\text{Ca}^+$ as a qubit

A popular choice, including the configuration used in the eQual setup at Trapped Ion Quantum Information (TIQI) for ions, is calcium, specifically its most abundant isotope $^{40}\text{Ca}^+$. Figure 2.1 shows the energy level splittings for such an ion. As there are many electronic states, selecting two for the qubit is necessary. For certain ions, like $^9\text{Be}^+$ or the odd isotope $^{43}\text{Ca}^+$, a feasible option is encoding the qubit in the hyperfine levels of the ground state $S_{1/2}$ called a *hyperfine qubit*. In the case of $^{40}\text{Ca}^+$, we opt for an *optical qubit*, choosing two states from the $S_{1/2}$ and $D_{5/2}$ manifolds. Since this transition is a quadrupole transition, this selection leads to a qubit with a long lifetime. State detection utilises $P_{1/2}$ as an auxiliary state. When a laser is tuned to the $S_{1/2} \leftrightarrow P_{1/2}$ dipole transition and the ion is in $S_{1/2}$, photons can be scattered. Conversely, if the qubit is in the $D_{5/2}$ manifold, the laser is out of tune for all transitions, so the ion is *dark* [15]. It is customary to represent the ground state and excited state as $|g\rangle = S_{1/2}$ and $|e\rangle = D_{5/2}$ respectively,

instead of $|0\rangle$ and $|1\rangle$. This representation offers a more physical description compared to the purely logical and arbitrarily assigned digits.

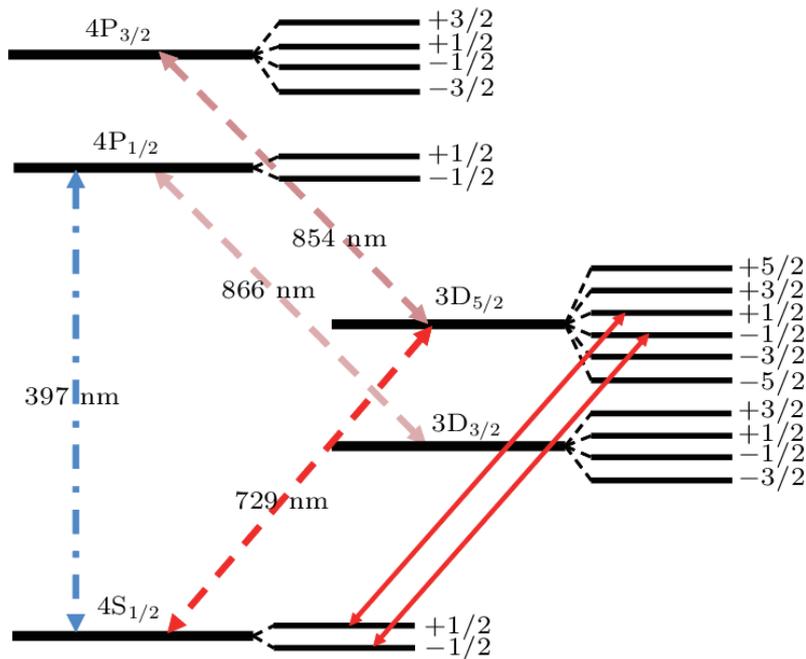


Figure 2.1: $^{40}\text{Ca}^+$ energy level splittings. Figure taken from [16].

We can express the Hamiltonian of just the two-level system as

$$\hat{H}_s = \hbar \frac{\omega_{ge}}{2} \hat{\sigma}_z, \quad (2.1)$$

where ω_{ge} is the energy difference between the two states, $\hat{\sigma}_z = |e\rangle\langle e| - |g\rangle\langle g|$ is the Pauli-Z matrix, and \hbar is the Planck constant.

The ions have to be contained somehow. Since in three dimensions, a static electric field alone is not capable of containing a particle, Wolfgang Paul invented a trap using both a static and an oscillating field [17]. While the classic Paul trap arranges electrodes in 3D, a more recent approach is surface traps, which "unfold" the electrodes onto a 2D plane for easier fabrication. Another solution utilises a combination of a static electric and magnetic field, called the *Penning trap*.

Given that the ions are well localised, the potential at the ion's position can be approximated via Taylor expansion by a quadratic potential. Therefore, the ion's motion can be described by a quantum harmonic oscillator (QHO). This results in the motional part of the Hamiltonian of

$$\hat{H}_m = \hbar\omega_m \left(\hat{a}^\dagger \hat{a} + \frac{1}{2} \right). \quad (2.2)$$

Here, ω_m denotes the trap frequency, while \hat{a}^\dagger and \hat{a} are the creation and annihilation operators of the harmonic oscillator. Note that this represents only one oscillator, while in a 3D trap, there are three modes. We will need only one mode, as the others can be cooled to the ground state and be uncoupled during operations. However, one motional mode is crucial, since multi-qubit gates use the motion as a mediator to couple different ion spins together, as discussed in section 2.1.4. For further details on trapping, refer to [18] and [19].

2.1.2 Ion-light interaction

As discussed previously, lasers are used to interact with the ion. Due to their monochromatic nature, lasers can be precisely tuned to exact transitions between the two electronic states of the qubit. Additionally, their narrow beams allow for the addressing of individual ions. If the states $|g\rangle$ and $|e\rangle$ are coupled by an electric field $\mathbf{E}_0 \cos(kz - \omega_L t + \phi)$, the Hamiltonian governing the interaction is given by

$$\hat{H}_I = \hbar\Omega\hat{\sigma}_x \cos(kz - \omega_L t + \phi), \quad (2.3)$$

where $\Omega \propto |\mathbf{E}_0|$ denotes the Rabi frequency, $\hat{\sigma}_x = |e\rangle\langle g| + |g\rangle\langle e|$ is the Pauli-X matrix, ω_L is the laser frequency and k is the wave vector of the electric field in the z -direction [20].

Now we will treat z quantum-mechanically as the position operator in the motional QHO: $\hat{z} = z_0(\hat{a} + \hat{a}^\dagger)$. Here, $z_0 = \sqrt{\hbar/(2m\omega_z)}$ is the size of the wave packet in the motional ground state with the trap frequency of ω_z and the mass of the ion m . We can now also define the Lamb-Dicke parameter $\eta = kz_0$. Combining this with the Hamiltonian of the ion itself, $\hat{H}_0 = \hat{H}_s + \hat{H}_m$, the total Hamiltonian including the light interaction becomes

$$\hat{H}_{lab} = \hat{H}_0 + \hbar\frac{\Omega}{2}(\hat{\sigma}_+ + \hat{\sigma}_-) \left(e^{i(\eta(\hat{a} + \hat{a}^\dagger) - \omega_L t + \phi)} + h.c. \right). \quad (2.4)$$

We also expanded the cosine into its exponential form and used $\hat{\sigma}_x = \hat{\sigma}_+ + \hat{\sigma}_-$. Now we change to the interaction picture with respect to \hat{H}_0 :

$$\hat{H} = \hbar\frac{\Omega}{2} \left(e^{i\omega_{ge}t}\hat{\sigma}_+ + e^{-i\omega_{ge}t}\hat{\sigma}_- \right) \left(e^{i\eta(e^{-i\omega_m t}\hat{a} + e^{i\omega_m t}\hat{a}^\dagger)} e^{-i(\omega_L t - \phi)} + h.c. \right). \quad (2.5)$$

Since the laser frequency ω_L should drive the qubit transition with frequency ω_{ge} , we want $\omega_L \approx \omega_{ge}$. Therefore, $|\omega_L - \omega_{ge}| \ll \omega_L + \omega_{ge}$, allowing us to drop the fast counter-rotating terms utilising the rotating-wave approximation. We also express the detuning as $\Delta = \omega_L - \omega_{ge}$. This results in the Hamiltonian

$$\hat{H} = \hbar\frac{\Omega}{2}\hat{\sigma}_+ e^{i\eta(e^{-i\omega_m t}\hat{a} + e^{i\omega_m t}\hat{a}^\dagger)} e^{-i(\Delta t - \phi)} + h.c.. \quad (2.6)$$

If we are in the Lamb-Dicke regime, where $\eta \ll 1$, we can expand the exponential $e^{i\eta(e^{-i\omega_m t}\hat{a} + e^{i\omega_m t}\hat{a}^\dagger)} = 1 + i\eta(e^{-i\omega_m t}\hat{a} + e^{i\omega_m t}\hat{a}^\dagger)$.

The final Hamiltonian for the ion-light interaction is now

$$\hat{H} = \hbar\frac{\Omega}{2}\hat{\sigma}_+ \left(e^{-i\Delta t} + i\eta e^{-i(\omega_m + \Delta)t}\hat{a} + i\eta e^{i(\omega_m - \Delta)t}\hat{a}^\dagger \right) e^{i\phi} + h.c.. \quad (2.7)$$

There are three resonances depending on the detuning of the laser Δ . By choosing $\Delta = 0$, the first term (carrier) is prevalent, which interacts only with the spin and not the motion; it couples the states $|g, n\rangle$ and $|e, n\rangle$, where n is the motional state. When $\Delta = -\omega_m$ however, the second term (red sideband) couples the states $|g, n\rangle$ and $|e, n-1\rangle$, thus altering the motional state. Similarly, for $\Delta = \omega_m$, the third term (blue sideband) couples $|g, n\rangle$ and $|e, n+1\rangle$. It is important to note that also the other off-resonant terms are still in play, although not as strong. This is especially relevant for this thesis, as a noisy phase $\phi(t)$ broadens the spectrum of the drive such that it can excite these terms.

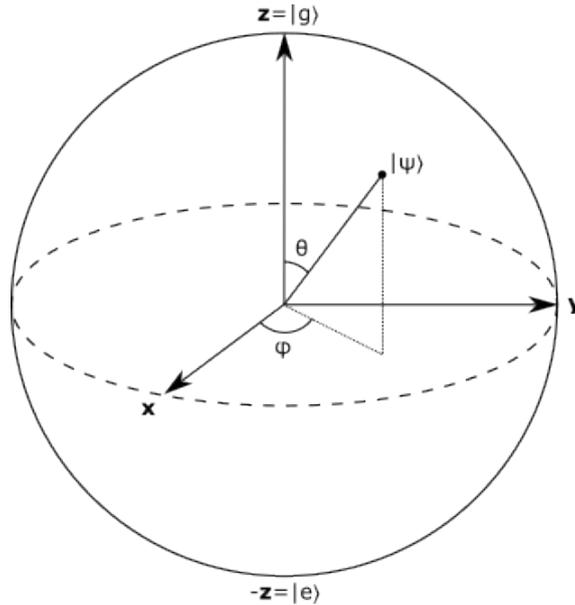


Figure 2.2: Bloch sphere representation of a qubit state $|\psi\rangle$.

2.1.3 Single-qubit rotations

The qubit state in a general form is represented as $|\psi\rangle = \alpha |g\rangle + \beta |e\rangle$, where $\{\alpha, \beta\} \in \mathbb{C}$ are probability amplitudes. Due to normalisation and the irrelevant global phase, the state can be expressed with only two real numbers:

$$|\psi\rangle = \cos(\theta/2) |g\rangle + e^{i\phi} \sin(\theta/2) |e\rangle. \quad (2.8)$$

These numbers are deliberately chosen as angles, allowing the qubit state to be visualised on the surface of a sphere, called the Bloch sphere (see figure 2.2). Here, θ and ϕ represent the polar and azimuthal angles. Because single-qubit operations rotate the state vector in this polar coordinate system, they are also called rotations.

To understand how such a rotation is performed, we revisit the previous result, equation 2.7. Since the qubit is stored in the spin of the ion, we are not interested in the motional part for the moment. Therefore, we set $\Delta = 0$ and use only the carrier excitation. Hence, the Hamiltonian of the single qubit rotation is

$$\hat{H}_{sq} = \hbar \frac{\Omega}{2} \hat{\sigma}_+ e^{i\phi} + h.c.. \quad (2.9)$$

By utilising the fact that $\hat{\sigma}_x = \hat{\sigma}_- + \hat{\sigma}_+$ and $\hat{\sigma}_y = i(\hat{\sigma}_- - \hat{\sigma}_+)$, the previous equation can be transformed into

$$\hat{H}_{sq} = \hbar \frac{\Omega}{2} [\cos(-\phi) \hat{\sigma}_x + \sin(-\phi) \hat{\sigma}_y]. \quad (2.10)$$

Here it is much easier to see the effects of the gate. Firstly, the Rabi frequency Ω determines the speed of the rotation. The Pauli matrices $\hat{\sigma}_x$ and $\hat{\sigma}_y$ act as a rotation around the x- and y-axes, respectively. The phase ϕ therefore selects a rotation axis anywhere on the xy-plane. Note the importance of a stable phase here. If the phase is noisy and changes with time, also the rotational axis changes and the implemented gate is no longer the same.

2.1.4 Multi-qubit interaction with the MS gate

Controlling a single qubit is not sufficient to create a quantum computer. Entanglement, which connects several qubits into a larger computational space, is also necessary. The first entangling gate was proposed by Ignacio Cirac and Peter Zoller [21]. However, achieving high gate fidelities with this gate is challenging, as it requires the ions to be in the motional ground state. To address this, Anders Sørensen and Klaus Mølmer proposed a phase gate, known as the MS gate, which works even in thermal motion.

The MS gate, like all trapped-ion entangling gates, leverages the coupling of motion of ions in a trap by the Coulomb force between the ions. Thus, a common motional mode can be utilised to mediate entanglement. Since the MS gate depends on the motion, we cannot ignore it, as in the single-qubit case. To derive the gate, I will initially focus on the light interaction for one ion.

The MS gate operates by modulating the laser with two frequency tones, therefore applying two sidebands to the ion simultaneously: the red sideband (RSB) and blue sideband (BSB). We again use equation 2.7 as a starting point. We choose $\Delta = -\omega_m - \delta$ to tune to the RSB and $\Delta = \omega_m + \delta$ for the BSB, where δ is a small symmetric detuning from the sidebands. As the laser is now bichromatic, the phases can also differ, therefore we relabel them as $\phi = \phi_R$ and $\phi = \phi_B$ for the RSB and BSB respectively. We also assume that the laser intensities and therefore the Rabi frequency Ω are the same for both sidebands. The Hamiltonian for the ion j is then the sum of the RSB and BSB Hamiltonian:

$$\begin{aligned} \hat{H}_{MS,j} = & \hat{H}_{RSB,j} + \hat{H}_{BSB,j} = \\ & \hbar \frac{\Omega}{2} \hat{\sigma}_+^{(j)} \left[\left(e^{i(\omega_m + \delta)t} + i\eta e^{i\delta t} \hat{a} + i\eta e^{i(2\omega_m + \delta)t} \hat{a}^\dagger \right) e^{i\phi_{R,j}} \right. \\ & \left. + \left(e^{-i(\omega_m + \delta)t} + i\eta e^{-i(2\omega_m + \delta)t} \hat{a} + i\eta e^{-i\delta t} \hat{a}^\dagger \right) e^{i\phi_{B,j}} \right] + h.c.. \end{aligned} \quad (2.11)$$

Since we want to entangle ions, we have to apply this laser scheme to several ions. In the minimal case of two ions, this results in

$$\hat{H}_{MS} = \sum_{j=1,2} \hat{H}_{MS,j}. \quad (2.12)$$

This is the final MS Hamiltonian I will later use in MS gate simulations. We can rewrite it differently to enhance understanding. Firstly, we rewrite the phases into the phase of the spin $\phi_s = \frac{\phi_B + \phi_R}{2}$ and the phase of the motion $\phi_m = \frac{\phi_B - \phi_R}{2}$. Secondly, we can use the transformation $\hat{\sigma}_\phi = \hat{\sigma}_x \cos \phi + \hat{\sigma}_y \sin \phi$ [22]. This allows us to decompose equation 2.11 into the off-resonant carrier, near-resonant and off-resonant sideband:

$$\hat{H}_{MS,j} = \hat{H}_{car} + \hat{H}_{res} + \hat{H}_{offres}, \quad (2.13)$$

$$\begin{aligned} \hat{H}_{car} &= \hbar \Omega \hat{\sigma}_{-\phi_s} \cos((\omega_m + \delta)t - \phi_m) \\ \hat{H}_{res} &= -\hbar \frac{\Omega}{2} \eta \hat{\sigma}_{\frac{\pi}{2} - \phi_s} \left(\hat{a} e^{i(\delta t - \phi_m)} + \hat{a}^\dagger e^{-i(\delta t - \phi_m)} \right) \\ \hat{H}_{offres} &= -\hbar \frac{\Omega}{2} \eta \hat{\sigma}_{\frac{\pi}{2} - \phi_s} \left(\hat{a} e^{-i((2\omega_m + \delta)t - \phi_m)} + \hat{a}^\dagger e^{i((2\omega_m + \delta)t - \phi_m)} \right). \end{aligned} \quad (2.14)$$

In the ideal scenario, we would only have the resonant \hat{H}_{res} , which would drive a perfect MS gate. However, there is still a carrier approximately ω_m off-resonantly as well as another sideband excitation detuned by $2\omega_m$. Although these two other terms oscillate fast compared to the resonant term and therefore approximately average out over time, they still affect the qubit state, lowering the final gate fidelity, especially if they are excited by noise. However, to understand how this Hamiltonian implements the gate, we will do a second rotating wave approximation and drop them for now. Also, without loss of generality, we set the phases $\phi_s = \phi_m = 0$. Therefore, the simplified MS gate Hamiltonian for one ion is given by

$$\hat{H}_{MS} = -\hbar \frac{\Omega}{2} \eta \hat{\sigma}_y \left(\hat{a} e^{i\delta t} + \hat{a}^\dagger e^{-i\delta t} \right). \quad (2.15)$$

We want to derive the gate propagator from this. Since equation 2.15 does not commute with itself at different times, we use the Magnus expansion:

$$\hat{U}_{MS}(t) = e^{-\frac{i}{\hbar} \int_0^t dt' \hat{H}_{MS}(t') - \frac{1}{2\hbar^2} \int_0^t dt' \int_0^{t'} dt'' [\hat{H}_{MS}(t'), \hat{H}_{MS}(t'')] + \dots} \quad (2.16)$$

To calculate the commutator, we will use $[\hat{a}, \hat{a}^\dagger] = 1$:

$$\left[\hat{H}_{MS}(t'), \hat{H}_{MS}(t'') \right] = \left(\hbar \frac{\Omega}{2} \eta \right)^2 2i \sin(\delta(t' - t'')) \hat{\sigma}_y^2. \quad (2.17)$$

Since $\hat{\sigma}_y^2$ is equal to the identity, all further commutators in the expansion are zero and the Magnus expansion terminates. Performing the integration of the exponents therefore leads to the exact propagator for the given Hamiltonian:

$$\hat{U}_{MS}(t) = e^{(\alpha(t)\hat{a}^\dagger - \alpha^*(t)\hat{a})\hat{\sigma}_y} e^{-i\beta(t)\hat{\sigma}_y^2}, \quad (2.18)$$

where we defined $\alpha(t) = -\frac{\Omega}{2}\eta \sin(\delta t/2)e^{-i\delta t/2}$ and $\beta(t) = \left(\frac{\Omega\eta}{2\delta}\right)^2 (\delta t - \sin(\delta t))$. The first exponential here is the displacement operator. It moves the quantum state in the motional phase space in circles. The second exponential is the geometric phase, which gets picked up during this displacement and is proportional to the area of the circle. For a single ion, a global phase does not matter, however. Hence, to see the entangling effect, we introduce a second ion by replacing $\hat{\sigma}_y$ by the operator on two ions, $\hat{J}_y = \hat{\sigma}_y \otimes \mathbb{1} + \mathbb{1} \otimes \hat{\sigma}_y$. The derivation is still valid, since although \hat{J}_y^2 is not the identity anymore, it only contains terms with $\hat{\sigma}_y$, which commute with all operators in \hat{H}_{MS} and therefore the Magnus expansion still terminates. The two-ion propagator is then:

$$\hat{U}_{MS}(t) = e^{(\alpha(t)\hat{a}^\dagger - \alpha^*(t)\hat{a})\hat{J}_y} e^{-i2\beta(t)(\mathbb{1} \otimes \mathbb{1} + \hat{\sigma}_y \otimes \hat{\sigma}_y)}. \quad (2.19)$$

Now the geometric phase depends on the quantum state, which was the goal. Since we do not want the ions entangled with the motion at the end of the gate, we set $\alpha(t_g) = 0$, which leads to a gate time of $t_g = \frac{2\pi}{\delta}$. Selecting the detuning $\delta = 2\eta\Omega$ to achieve the maximally entangled state, the propagator is

$$\hat{U}_{MS}(t_g) = e^{-i\frac{\pi}{4}(\mathbb{1} \otimes \mathbb{1} + \hat{\sigma}_y \otimes \hat{\sigma}_y)}. \quad (2.20)$$

This implements a controlled phase gate in the Y-basis. We can express the actions of this gate in the computational basis:

$$\begin{aligned}
 \hat{U}_{MS}(t_g) |gg\rangle &= \frac{1}{\sqrt{2}} (|gg\rangle + i |ee\rangle) \\
 \hat{U}_{MS}(t_g) |ge\rangle &= \frac{1}{\sqrt{2}} (|ge\rangle - i |eg\rangle) \\
 \hat{U}_{MS}(t_g) |eg\rangle &= \frac{1}{\sqrt{2}} (-i |ge\rangle + |eg\rangle) \\
 \hat{U}_{MS}(t_g) |ee\rangle &= \frac{1}{\sqrt{2}} (i |gg\rangle + |ee\rangle).
 \end{aligned} \tag{2.21}$$

We note that the gate transforms the ground state $|gg\rangle$ into a bell state and is therefore indeed a maximally entangling gate.

The final propagator derived in this section, equation 2.19, is consistent with the original paper by Sørensen and Mølmer [23], only the sign of the exponent differs. This is expected due to the different definition of the detuning Δ from the carrier. In their paper, they assume a detuning $\Delta = \omega_m - \delta$, whereas in my derivation, it is the sum $\Delta = \omega_m + \delta$. Note that my Δ is δ in their notation. This different definition results in a different sign for the geometric phase. However, my derivation is different to [23] and is more similar to [24, p: 102-103], [22, p. 82-85] and [25, p. 35-45]. Nevertheless, they are not exactly the same as mine. For instance, [24] introduces a common phase ϕ_c , possibly equivalent to my spin phase ϕ_s and later treated as a spin phase, yet the signs for ϕ_c are opposite for RSB and BSB. The spin phase is the average of RSB and BSB phases, leading to $\phi_s = 0$. Additionally, the signs for the detuning δ_m should be opposite in equation 6.33 according to my derivation. The final propagator has a sign switch accordingly. In [25], there is a missing factor of i (and $-i$ for the hermitian conjugate) in all Hamiltonians, leading to the MS gate acting as a natural gate in X-basis instead of Y-basis when all phases are set to zero. However, the sign of the phase aligns with mine. My derivation is the closest to [22], with minor discrepancies such as a missing factor of i in equation 5.15, although it does not affect subsequent equations. Also, the calculation of the propagator is a little bit different, nevertheless leading to the same propagator. However, the state mapping retrieved from applying the propagator in equation 5.25 introduces a sign mismatch compared to my state mapping in equation 2.21.

Only [24] includes the off-resonant carrier and off-resonant sideband in their MS Hamiltonian. These terms are very important in my case, since driving the carrier term with noise is the biggest factor in gate error due to phase noise.

2.2 Impact of noise on trapped-ion qubits

We make use of sinusoidal signals to control the ions, be it in the form of lasers or electrical signals controlling the latter. However, every real signal is subject to noise. Such a noisy sinusoidal $U(t)$ can be described by the following model [26, 27]:

$$U(t) = U_0 (1 + \alpha(t)) \cos(\omega_0 t + \phi(t)), \tag{2.22}$$

where U_0 is the amplitude and ω_0 is the frequency. The parameters $\alpha(t)$ and $\phi(t)$ modify the amplitude and phase of the signal and are therefore referred to as amplitude and phase noise, respectively. For an ideal signal, they are $\alpha(t) = 0$ and $\phi(t) = \phi_R$, where ϕ_R is the ideal phase. The Fourier spectrum of the signal is depicted in figure 2.3a, where the ideal signal is a Dirac delta function at ω_0 . When $\alpha(t)$ and $\phi(t)$ are not constants, they modulate

the carrier. Thus, some of the power in the carrier goes into sidebands. This results in a broadening of the peak, and the noise manifests as a pedestal around the carrier. The shape of the pedestal depends on the type of noise. A zoom-in into one sideband is illustrated in figure 2.3b. Here, three main types of noise are described, each differing in their frequency dependence [28, 29]. The noise floor, therefore the minimum amount of noise, is determined by Johnson-Nyquist noise or thermal noise [30]. It arises from statistical fluctuations of charges in a conductor due to temperature. Going closer to the carrier, flicker noise will start to take over. Flicker noise comes mostly from semiconductor devices. It is generally not well understood but believed to come from trapped charges at oxide-semiconductor interfaces or mobility fluctuations [29]. It follows a $\frac{1}{f}$ dependence on frequency. Even closer to the carrier is white frequency noise. It often comes from passive-resonator frequency standards like quartz oscillators and follows a $\frac{1}{f^2}$ dependence on frequency [28].

In general, stationary noise sources contribute both to amplitude noise and phase noise to equal parts. So why are we only concerned with phase noise in this thesis? Amplitude noise does not play as significant a role as phase noise in electronics, as phase noise dominates in power in most circumstances. Amplifying a noisy signal with a non-linear, limiting amplifier rejects amplitude noise, while phase noise stays the same. An example is the noise coming from an oscillator. Oscillators are used as timing references in nearly all electronic circuits. For RF generation, the noise from the oscillator directly limits the achievable noise level at the output. An oscillator allows for one amplitude solution, while the phase can be chosen freely. Therefore, in case of an amplitude noise impulse, the oscillator steers itself back to the steady-state amplitude; the noise gets rejected. In the case of a phase shift due to noise, no restoring force brings it back [31]. Therefore, up to frequencies where the restoring time is low enough, phase noise dominates. This can be seen in a simulation of an oscillator in figure 2.4. Also, to reach higher clock frequencies, the frequency of an oscillator often gets multiplied by a frequency multiplier. Such a multiplier increases phase noise linearly with frequency, while the amplitude noise keeps the same [32]. In conclusion, in most cases, amplitude noise is negligible in comparison to phase noise. In certain cases, however, such as far from the carrier, it might become relevant. In the following, I will focus on phase noise and its impact on quantum gates, following the argument above.

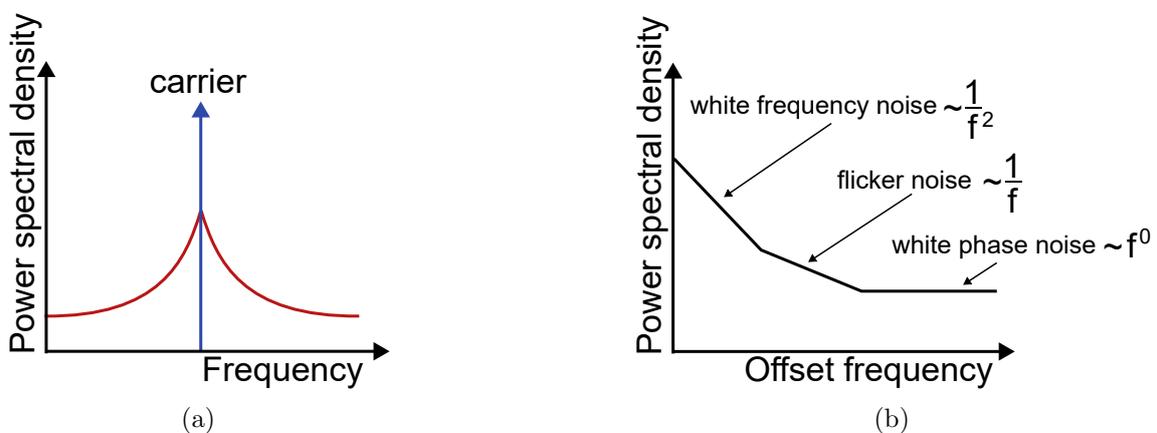


Figure 2.3: (a) Depiction of a noise-widened carrier. The frequency axis is linear. (b) A zoom-in on the noise near the carrier. The frequency axis is logarithmic.

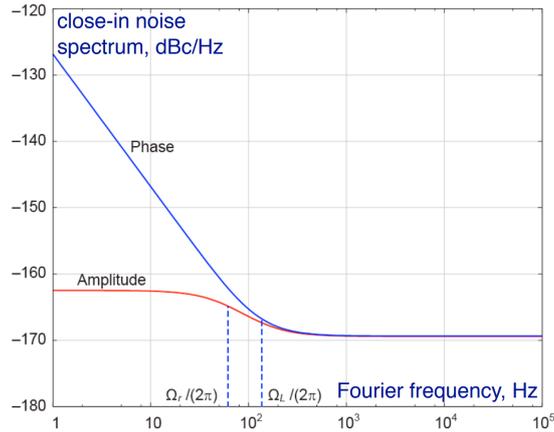


Figure 2.4: Simulation of a Van der Pol Oscillator. Figure taken from [33].

2.2.1 Phase noise

As described in equation 2.22, phase noise is the variation of the phase of a signal by a random, time-dependent variable $\phi(t)$. This is a form of phase modulation, with noise instead of information. Since any signal can be written as a sum of sinusoids via the Fourier transform, we will initially assume $\phi(t) = \delta \cos(\omega_m t)$. The modulation is described by just the amplitude δ and the frequency ω_m . This simplification to a single tone makes it clearer to see the effect of the modulation on the modulated spectrum in the carrier-frequency domain. We also drop the amplitude noise $\alpha(t)$ now since we are looking at phase noise. The resulting modulated signal can then be described by [26]

$$\begin{aligned}
 U(t) &= U_0 \cos(\omega_0 t + \delta \cos(\omega_m t)) \\
 &= U_0 \Re \left\{ e^{i\omega_0 t} e^{i\delta \cos(\omega_m t)} \right\} \\
 &= U_0 \sum_{n=-\infty}^{\infty} \Re \left\{ (i)^n J_n(\delta) e^{i(\omega_0 + n\omega_m)t} \right\}.
 \end{aligned} \tag{2.23}$$

Here, J_n is the Bessel function of the n -th order. In the last line, $U(t)$ is written as an infinite sum of tones. They are located at multiples of ω_m around the carrier with the amplitude J_n . Therefore, in contrast to amplitude modulation, the spectrum of phase modulation in the carrier-frequency domain extends further than the frequency of the modulation itself. Figure 2.5 shows the spectrum for a high modulation amplitude δ (black) compared to the carrier without modulation (blue). Note that, due to Parseval's theorem, the total power of all tones is equal to the power of the unmodulated carrier. In most cases, however, it is reasonable to assume that $\delta \ll 1$, where only the first sideband is relevant. This assumption holds true for most electronic devices, as efforts are typically made to minimise noise levels.

Let us extend this to arbitrary noise. In time domain, the random variable $\phi(t)$ can be described by a probability density function. In nature, this is often a Gaussian distribution, since it is the limit of the sum of several sources, following the central limit theorem [32, 26]. Using the Fourier transformation, the power spectral density $S_\phi(f)$ of $\phi(t)$ describes the signal in frequency domain. The two descriptions are linked by the fact that the total noise power, therefore the integral of $S_\phi(f)$, is equal to the variance σ^2 of $\phi(t)$ (see figure 2.6a). Since $\phi(t)$ is an angle with units rad , $S_\phi(f)$ is in units rad^2/Hz .

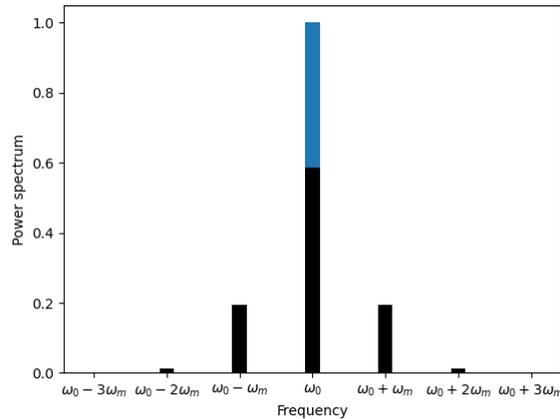


Figure 2.5: Spectrum of a phase modulated signal with modulation amplitude $\delta = 1$ in black. The spectrum of the unmodulated carrier can be seen in blue as a single tone at ω_0 .

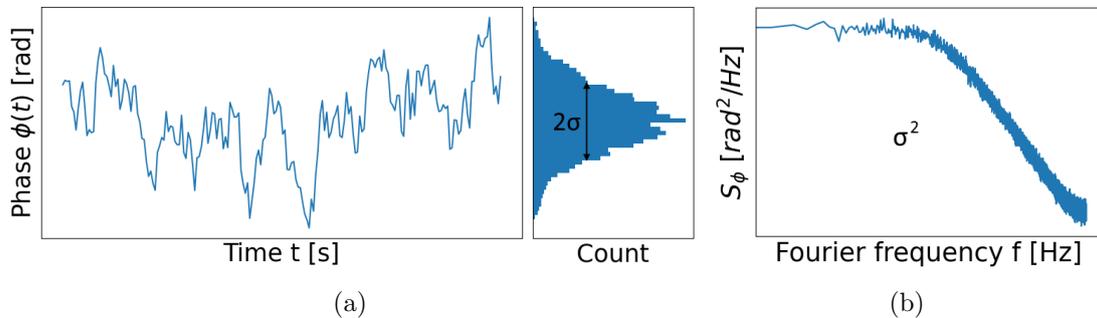


Figure 2.6: (a) The signal $\phi(t)$ in time domain and its histogram. The signal is approximately Gaussian distributed with a standard deviation of σ . (b) The Fourier transformed power density spectrum $S_\phi(f)$ (log-log plot). The area under the curve is equal to the variance of the signal in time domain.

Analogous to the phase modulation with a single sinusoid, the carrier is now modulated by the whole spectrum $S_\phi(f)$. Since the signal does not contain just a single Dirac impulse anymore, convolution has to be used to describe the modulation correctly. The power density spectrum $S_E(f)$ in the carrier-frequency domain is now [34]

$$S_E(\nu - \nu_0) = e^{-\sigma^2} \delta(f) + e^{-\sigma^2} S_\phi(f) + e^{-\sigma^2} \sum_{n=2}^{\infty} \frac{1}{n!} S_\phi(f) *^{n-1} S_\phi(f), \quad (2.24)$$

where $f = \nu - \nu_0$ is the frequency offset from the carrier frequency ν_0 . The first term here is the Dirac function of the carrier itself. The phase noise $S_\phi(f)$ is placed around the carrier unchanged in the second term, similar to the first sideband in the single-tone case. The remaining terms are repeated convolutions of $S_\phi(f)$ with itself. Note also the scaling factor $e^{-\sigma^2}$ before both the carrier term and the noise terms, which becomes smaller, the higher the noise variance is. This comes from the fact that the higher the noise power is, the more power goes from the carrier into higher and higher-order terms. An example of modulated noise is given in figure 2.7. Here, $S_\phi(f)$ in red is assumed to be white noise with a bandwidth of 15.6MHz and a variance of $\sigma^2 = 1$. The cut-off frequency is assumed to be ideal to emphasise the effect of the modulation. The modulated spectrum S_E is shown in blue. Since the variance and therefore noise power is relatively high, one can see quite

high noise density above the cut-off frequency, coming from the higher order terms.

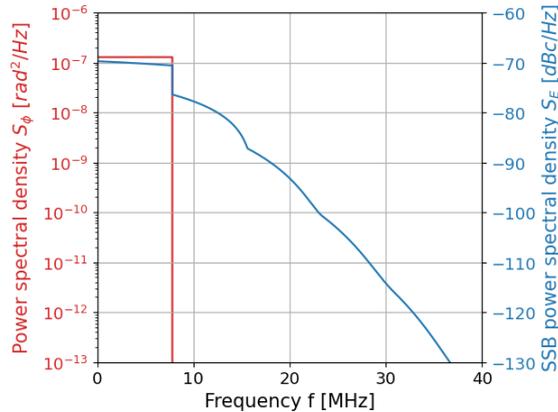


Figure 2.7: The power spectral density of white noise, filtered with an ideal low-pass filter, in the Fourier domain (S_ϕ , red) and in the carrier domain (SSB S_E , blue).

The power spectral density S_E is given in units of dBc/Hz. It is the ratio of the noise power in a 1Hz bandwidth to the carrier power (the "c" denotes "relative to the carrier"). This definition comes from phase noise measurements with spectrum analysers in earlier days, where the power at the carrier frequency is compared to the power at a particular frequency offset. This also includes amplitude noise however, and is also distorted by the higher order terms, as it can be seen in figure 2.7. Nowadays, dedicated measurement instruments like a signal source analyser (SSA) can measure $S_\phi(f)$ directly. Therefore, in 1999, the Institute of Electrical and Electronics Engineers (IEEE) defined phase noise as [35]

$$\mathcal{L}(f) = \frac{S_\phi(f)}{2}. \quad (2.25)$$

The factor of two is due to two different conventions. $S_\phi(f)$ contains the power of both negative and positive frequencies (double-sideband (DSB)), whereas $S_E(f)$ disregards noise power in the lower sideband and only contains the upper sideband (single-sideband (SSB)). Note the potential for confusion, as in some literature the SSB noise is called the two-sided noise, since the noise is distributed onto two sides. Conversely, the one-sided noise is equivalent to the DSB noise and higher by the factor of two, or equivalently 3dB. Since the historical phase noise measurement with the spectrum analyser is a SSB measurement, it is established in data sheets to report the SSB noise. Also IEEE recommends the use of $\mathcal{L}(f)$ and therefore SSB phase noise. For small phase fluctuations ($\sigma^2 \ll 1$), the new definition of $\mathcal{L}(f)$ does not differ much from the spectrum $S_E(f)$, since only the first order term in equation 2.24 is non-negligible. In this case, also the units rad^2/Hz and dBc/Hz are interchangeable except for the conversion to decibel, since the small-angle approximation holds.

The total phase noise spectrum consisting of $\frac{1}{f}$ -type contributions as described in the introduction of this section can be written as the sum:

$$\mathcal{L}(f) = \sum_k \frac{\mathcal{L}_k}{f^k}, \quad (2.26)$$

where the coefficient \mathcal{L}_k is a measure of the power of the respective noise spectrum [9]. We

only consider the terms $k \in \{0, 1, 2\}$, corresponding to white phase noise, flicker noise and white frequency noise.

2.2.2 Phase noise in single-qubit rotations

Single-qubit rotations can be fully described by the classical description of rotations around the Bloch sphere. Therefore, it is possible to derive a mathematical formalism for the influence of phase noise on these rotations, which we will do in the following section. The derivation follows the work of Zilong Chen et al. [9]. Note that their calculations were carried out within the rotation group $SO(3)$. Although usually quantum operations are expressed in the special unitary group $SU(2)$, I will adhere to the former for consistency.

As seen in equation 2.10, each single-qubit rotation can be described by a rotation about a rotational axis in the xy -plane with the azimuthal angle ϕ relative to the x -axis. In the ideal case, where the phase is fixed to $\phi(t) = \phi_R$, the final Bloch vector \mathbf{J}_f° can be obtained from the initial vector \mathbf{J}_i by

$$\mathbf{J}_f^\circ = R(\phi_R, \psi)\mathbf{J}_i. \quad (2.27)$$

Here, $R(\phi_R, \psi)$ describes a rotation around the rotational axis given by ϕ_R and the rotation angle ψ . To study the effect of a variable phase, we modulate the phase $\phi(t) = \phi_R + \beta \cos(\omega_m t + \alpha_m)$ similar to equation 2.23 with a coherent tone. The rotational axis now oscillates around the mean in the xy -plane. Due to this changing axis, the final Bloch vector \mathbf{J}_f differs now from the ideal \mathbf{J}_f° by the deflection vector $\mathbf{j}_f = \mathbf{J}_f - \mathbf{J}_f^\circ$. This deflection can also be seen as a small rotation r :

$$\mathbf{J}_f = r\mathbf{J}_f^\circ. \quad (2.28)$$

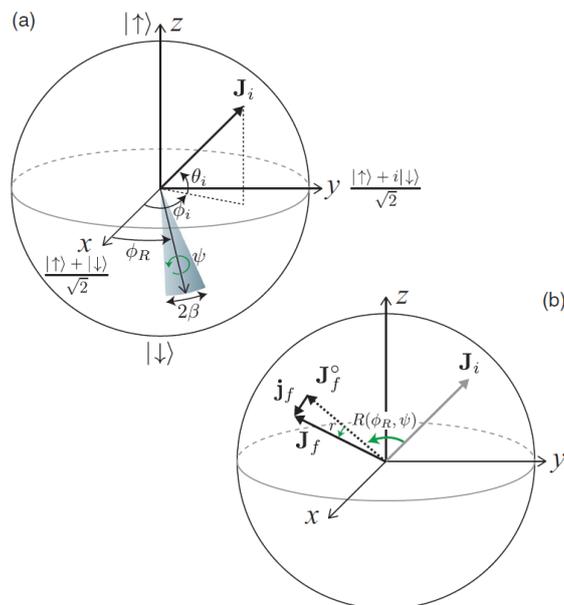


Figure 2.8: (a) The phase ϕ_R of the gate is modulated by a sinusoidal with amplitude β . Hence, the rotational axis oscillates in the xy -plane around the mean ϕ_R . (b) The ideal final Bloch vector \mathbf{J}_f° is deflected to \mathbf{J}_f by the modulation. The deflection vector is \mathbf{j}_f . Figure taken from [9].

A depiction of the modulated rotational axis and the resulting deflection vector \mathbf{j}_f can be seen in figure 2.8. The rotation r does not depend on the initial Bloch vector \mathbf{J}_i , since the modulation of the phase changes the rotational axis of the ideal rotation R and therefore the rotation itself, not the vector it is applied to. This can be thought of as splitting the ideal rotation R into infinitesimal rotations dR , which are interleaved by infinitesimal rotations changing the phase. These interleaved rotations can be expressed as one rotation rR , which does not involve \mathbf{J}_i . Instead, the rotation r depends on the parameters of the modulation, β and ω_m , and the parameters of the ideal rotation R , namely the rotation angle ψ and ϕ_R . The dependence on the phase ϕ_R can be overcome by changing the reference frame before the rotation and changing it back afterwards:

$$r(\phi_r) = R_z(\phi_R)r(0)R_z(-\phi_R). \quad (2.29)$$

Hence, it is enough to determine r for the case $\phi_R = 0$ and $\mathbf{J}_i = \mathbf{x}$, since in this case $\mathbf{J}_f^o = \mathbf{J}_i$. A tilde (\sim) denotes this special case. Under the assumption that $\beta \ll 1$, we can make use of the small-angle approximation. Therefore, the components of the deflection vector $\tilde{\mathbf{j}}_f = (0, \tilde{j}_y, \tilde{j}_z)$ can be approximated as angles and r can be split into rotations R_y and R_z with angles \tilde{j}_z and \tilde{j}_y . The rotation r is then

$$r(0) = R_y(-\tilde{j}_z)R_z(\tilde{j}_y) = \begin{pmatrix} 1 & -\tilde{j}_y & -\tilde{j}_z \\ \tilde{j}_y & 1 & 0 \\ \tilde{j}_z & 0 & 1 \end{pmatrix}. \quad (2.30)$$

Note that we dropped higher-order terms. We want to solve for \tilde{j}_y and \tilde{j}_z now. First, we write the time-dependent state in the interaction picture $\mathbf{J}_f = r\mathbf{J}_i$ as a density matrix $\hat{\rho}$ in SU(2):

$$\mathbf{J}_f = \begin{pmatrix} 1 \\ \tilde{j}_y \\ \tilde{j}_z \end{pmatrix} \equiv \hat{\rho} = \frac{1}{2} (\mathbb{1} + \hat{\sigma}_x + \tilde{j}_y \hat{\sigma}_y + \tilde{j}_z \hat{\sigma}_z). \quad (2.31)$$

I use the density matrix instead of the state vector because it simplifies the upcoming derivatives. Using $\hat{\rho}$, we can write down the von Neumann equation with the Hamiltonian 2.10, where we define $-\phi(t) = \beta \sin(\omega_m t + \alpha_m)$. The negative sign comes from different definitions of the Hamiltonian, it does not matter in the end since it introduces just a phase shift absorbed by α_m . Therefore, I will go with the definition of [9] and the von Neumann equation results in:

$$i\hbar \frac{d\hat{\rho}}{dt} = [\hat{H}_{sq}, \hat{\rho}], \quad (2.32)$$

$$\frac{d\tilde{j}_y}{dt} \hat{\sigma}_y + \frac{d\tilde{j}_z}{dt} \hat{\sigma}_z = \omega_m t [\tilde{j}_z \beta \sin(\omega_m t + \alpha_m) \hat{\sigma}_x - \tilde{j}_z \hat{\sigma}_y + (\tilde{j}_y - \beta \sin(\omega_m t + \alpha_m)) \hat{\sigma}_z]. \quad (2.33)$$

We can drop the term proportional to $\hat{\sigma}_x$, since the product $\tilde{j}_z \beta$ is small compared to the \tilde{j}_y, \tilde{j}_z and β alone. We can equate the coefficients of $\hat{\sigma}_y$ and $\hat{\sigma}_z$ individually to arrive at the differential equations:

$$\frac{d\tilde{j}_y}{d\psi} + \tilde{j}_z = 0, \quad (2.34)$$

$$\frac{d\tilde{j}_z}{d\psi} - \tilde{j}_y = -\beta \sin(\nu\psi + \alpha_m), \quad (2.35)$$

where we introduced $\nu = \omega_m/\Omega$ and replaced $\psi = \omega_m t$. Note that these coupled differential equations are the equations of motion of an undamped harmonic oscillator with resonance frequency Ω driven by an external force with frequency ω_m , with \tilde{j}_y and $-\tilde{j}_z$ as the displacement and velocity. You could arrive at the standard uncoupled differential equation of second order by elimination of \tilde{j}_z in equation 2.35 by substituting equation 2.34. We can now solve for \tilde{j}_y and \tilde{j}_z , assuming both are initially zero:

$$\tilde{j}_y = \frac{\beta}{1 - \nu^2} [-\nu \cos \alpha_m \sin \psi - \sin \alpha_m \cos \psi + \sin(\nu\psi + \alpha_m)], \quad (2.36)$$

$$\tilde{j}_z = \frac{\beta}{1 - \nu^2} [-\nu \cos \alpha_m \cos \psi - \sin \alpha_m \sin \psi + \cos(\nu\psi + \alpha_m)]. \quad (2.37)$$

We now have a formalism to express the deflection vector $\mathbf{j}_f = (j_x, j_y, j_z)$ dependent on a coherent phase modulation. For now, we adhere to a constant modulation frequency ω_m . However, we assume β and α_m now to be random. According to the assumption that many noise processes are Gaussian (see section 2.2.1), we sample the amplitude β from a Gaussian distribution with a zero mean and a variance of $\langle \beta^2 \rangle$. The phase α_m is assumed to be uniformly distributed between 0 and 2π . The SSB phase noise is now modeled as $\mathcal{L}(f) = \langle \beta^2 \rangle \delta(f - f_0)/4$. The denominator comes from equation 2.25 and the fact that $\langle \sin(t)^2 \rangle = 1/2$. We can now define the covariance transfer matrix as $T(\phi_R, \mathbf{J}_f^\circ, \psi, \omega_m) = 4\langle \mathbf{j}_f \mathbf{j}_f^\top \rangle / \langle \beta^2 \rangle$. This matrix tells us the variance of deflection along an arbitrary measurement axis \mathbf{n} . It is normalised such that $\int \mathbf{n}^\top \cdot T \cdot \mathbf{n} \mathcal{L}(f) df = \langle (\mathbf{j}_f \cdot \mathbf{n})^2 \rangle$, therefore, T is a direct map from noise spectrum to deflection vector. We can now use our previous results 2.36 and 2.37 to calculate the transfer matrix \tilde{T} depending on noise frequency ω_m and gate length ψ , again for the special case $\phi_R = 0$ and $\mathbf{J}_i = \mathbf{x}$:

$$\tilde{T}(\psi, \omega_m) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \tilde{T}_{yy} & \tilde{T}_{yz} \\ 0 & \tilde{T}_{zy} & \tilde{T}_{zz} \end{pmatrix}, \quad (2.38)$$

$$\begin{aligned} \tilde{T}_{yy}(\psi, \omega_m) &= \frac{2}{(1 - \nu^2)^2} \left\{ [\cos \psi - \cos(\nu\psi)]^2 + [\nu \sin \psi - \sin(\nu\psi)]^2 \right\}, \\ \tilde{T}_{zz}(\psi, \omega_m) &= \frac{2}{(1 - \nu^2)^2} \left\{ \nu^2 [\cos \psi - \cos(\nu\psi)]^2 + [\sin \psi - \nu \sin(\nu\psi)]^2 \right\}, \\ \tilde{T}_{yz}(\psi, \omega_m) &= \tilde{T}_{zy}(\psi, \omega_m) = \frac{2}{1 - \nu^2} [\cos \psi - \cos(\nu\psi)] \sin \psi. \end{aligned} \quad (2.39)$$

As an example, the transfer functions \tilde{T}_{yy} and \tilde{T}_{zz} are plotted for four different gate lengths ψ in figure 2.9. As expected, with increasing ψ , the magnitude of the transfer function increases. However, the plot also changes qualitatively. While the peak is always around $\nu = 1$, the transfer function experiences poles at different normalised frequencies. The undamped harmonic driven oscillator (equation 2.34) has a transient response of a beating with the frequency of the difference between the resonance at Rabi frequency Ω and the driving force ω_m . Sampling exactly a node of the beating with the gate length ψ translates to a pole in the transfer function. With increasing ψ , the beating becomes faster compared to the gate time and therefore the poles are spaced tighter. We can also observe that the longer the gate is applied, the sharper the peak is around the resonance $\Omega = \omega_m$. In the limit of $|\psi| \rightarrow \infty$, the transfer function becomes a Dirac delta function at resonance.

We can calculate the transfer matrix T for arbitrary ϕ_R and \mathbf{J}_i from \tilde{T} with the transformation

$$T = D(\phi_R, \mathbf{J}_f^\circ) \tilde{T}(\psi, f) D(\phi_R, \mathbf{J}_f^\circ)^\top \quad (2.40)$$

using the map

$$D(\phi_R, \mathbf{J}_f^\circ) = \begin{pmatrix} 0 & -J_y^\circ & -J_z^\circ \cos \phi_R \\ 0 & J_x^\circ & -J_z^\circ \sin \phi_R \\ 0 & 0 & -J_x^\circ \cos \phi_R + -J_y^\circ \sin \phi_R \end{pmatrix}. \quad (2.41)$$

Until now we only considered a single modulation frequency. The total noise in the Bloch vector, the covariance noise matrix, is simply the integral of the transfer matrix over the phase noise spectrum \mathcal{L} :

$$\tilde{V}(\psi) = \int_0^\infty \tilde{T}(\psi, f) \mathcal{L}(f) df. \quad (2.42)$$

Note that this still assumes $\beta \ll 1$ and consequently $\mathcal{L}(f)$ to be small. The arbitrary noise matrix can be found similarly to the transfer matrix:

$$V = D(\phi_R, \mathbf{J}_f^\circ) \tilde{V}(\psi) D(\phi_R, \mathbf{J}_f^\circ)^\top. \quad (2.43)$$

We calculate an arbitrary noise matrix of a single π pulse ($\psi = \pi$) with assumed white noise ($\mathcal{L}(f) = \mathcal{L}_\circ$) for demonstration. Note that for a π pulse, the initial and final ideal Bloch vector are related by $\mathbf{J}_i = -\mathbf{J}_f$. We also write the initial Bloch vector in polar coordinates as $\mathbf{J}_i = (\cos \theta_i \cos \phi_i, \cos \theta_i \sin \phi_i, \sin \theta_i)$ with the polar angle θ_i and the azimuthal angle ϕ_i like in figure 2.2.

$$\begin{aligned} V(\psi = \pi) &= D(\phi_R, -\mathbf{J}_i) \mathcal{L}_\circ \int_0^\infty \tilde{T}(\psi, f) df D(\phi_R, -\mathbf{J}_i)^\top \\ &= \frac{\mathcal{L}_\circ \Omega}{2} D(\phi_R, -\mathbf{J}_i) \begin{pmatrix} 0 & 0 & 0 \\ 0 & \psi - \frac{1}{2} \sin 2\psi & -\sin^2 \psi \\ 0 & -\sin^2 \psi & \psi + \frac{1}{2} \sin 2\psi \end{pmatrix} D(\phi_R, -\mathbf{J}_i)^\top \\ &= \frac{\pi \mathcal{L}_\circ \Omega}{2} \begin{pmatrix} 1 - \cos^2 \theta_i \cos^2 \phi_i & \frac{1}{2} \cos^2 \theta_i \sin 2\phi_i & \frac{1}{2} \sin 2\theta_i \cos \phi_i \\ \frac{1}{2} \cos^2 \theta_i \sin 2\phi_i & \cos^2 \theta_i \cos^2 \phi_i & 0 \\ \frac{1}{2} \sin 2\theta_i \cos \phi_i & 0 & \cos^2 \theta_i \cos^2 \phi_i \end{pmatrix} \end{aligned} \quad (2.44)$$

To translate this result into a gate fidelity, we can evaluate the state fidelity [36] after the gate. It describes the difference between an ideal gate and an imperfect implementation. Since the noise matrix V is already the deviation from the ideal Bloch vector, we can obtain the gate infidelity with

$$1 - F = \text{tr}(V)/4. \quad (2.45)$$

For the example of the π pulse in equation 2.44 above, this results in the infidelity

$$1 - F = \frac{1}{8} \pi \Omega \mathcal{L}_\circ (1 + \cos^2 \theta_i \cos^2 \phi_i). \quad (2.46)$$

We can see that the infidelity depends on the initial state. To get a more general measure for the influence of phase noise, we can average the infidelity over all possible initial states.

We can also calculate it for an arbitrary gate length ψ , since no calculations of \mathbf{J}_i are needed anymore. By doing so we obtain

$$\langle 1 - F \rangle = \frac{1}{6} \psi \Omega \mathcal{L}_\circ. \quad (2.47)$$

As expected, the gate infidelity scales linearly with the gate length ψ . However, note that it also scales with the noise density \mathcal{L}_\circ , but not with the bandwidth of the noise and consequently the total noise power, since the bandwidth is assumed to be infinite in the example above. Furthermore, it is linear in the Rabi frequency Ω . This behaviour can be explained by the transfer matrix which peaks at the resonance at Ω with a peak width scaling with Ω . Hence, the gate infidelity can be characterised by the product of Rabi frequency and the phase noise density at said frequency for long gates.

In contrast, it is not as easy for short gates. As it can be seen in figure 2.9, shorter gates lose the zero at DC. $\tilde{T}_{yy}(\psi = \pi)$ already does so for the π pulse, while for even shorter pulses, also \tilde{T}_{zz} includes slow noise. The inclusion of frequencies near DC might be a particular problem, since in this range the phase noise density is the highest due to flicker and white frequency noise. Let us look at the magnitude of the problem. All short gates that sample DC frequencies sample quite a broad frequency range $> \Omega$. Thus, if the corner frequencies f_{c1} and f_{c2} of the $\frac{1}{f}$ and $\frac{1}{f^2}$ noise are below Ω , all contribution from $\frac{1}{f}$ and $\frac{1}{f^2}$ noise is included in the bandwidth. Therefore, we cannot think about noise density anymore, but total noise power. We therefore integrate the phase noise density. As the upper limit, we can choose infinity for $\frac{1}{f^2}$. Flicker noise needs an upper limit since the integral does not converge. We can set it to $10f_{c1}$. The exact value does not really matter, as the contribution of $\frac{1}{f^2}$ is much higher anyway. For the lower limit, we cannot choose 0. The limits $\lim_{f \rightarrow 0} \frac{1}{f}$ and $\lim_{f \rightarrow 0} \frac{1}{f^2}$ are infinity, which is not physical. The lowest realistic frequency is determined by the acquisition time of the system. In our case, this is the maximum time of an experiment from state preparation to readout. Slower phase variations do not have an impact, since phase is determined by the first pulse applied to a qubit. Let us assume a maximum experiment time of 1s. At an MS gate time of $100\mu\text{s}$ (see section 5.2.2), this equates to ten thousand gates. Let us choose a Rabi frequency of $\frac{\Omega}{2\pi} = 100\text{kHz}$ for simplicity. Thus, 1s also equates to two million single-qubit gates. Frequencies lower than $f_{min} = \frac{1}{1\text{s}} = 1\text{Hz}$ are therefore not observed. As corner frequencies, we choose $f_{c1} = 1\text{kHz}$ and $f_{c2} = 300\text{Hz}$, both with respect to white noise of $\mathcal{L}_\circ = -100\text{dBc/Hz}$.

First, we compare the noise power contribution of the white noise with an effective bandwidth of $\frac{\Omega}{2\pi} = 100\text{kHz}$ given by $\tilde{T}_{yy}(\psi = \pi, \omega_m)$ to the contribution of $\frac{1}{f}$ and $\frac{1}{f^2}$ noise with parameters above, where P_k is the noise contribution of noise with frequency dependence $\frac{1}{f^k}$ (see equation 2.26):

$$\begin{aligned} P_0 &= \int_0^{\frac{\Omega}{2\pi}} \mathcal{L}_\circ df = 1 \cdot 10^{-5} = -50\text{dBc} \\ P_1 &= \int_{f_{min}}^{10f_{c1}} \frac{\mathcal{L}_1}{f} df = \int_{f_{min}}^{10f_{c1}} \mathcal{L}_\circ \frac{f_{c1}}{f} df = 6.9 \cdot 10^{-8} = -71.6\text{dBc} \\ P_2 &= \int_{f_{min}}^{\infty} \frac{\mathcal{L}_1}{f^2} df = \int_{f_{min}}^{\infty} \mathcal{L}_\circ \frac{f_{c1}^2}{f^2} df = 9 \cdot 10^{-8} = -50.46\text{dBc}. \end{aligned} \quad (2.48)$$

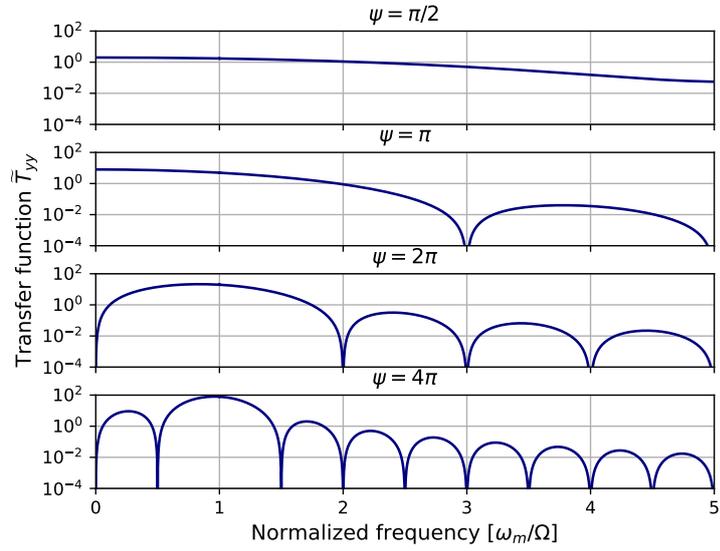
We can see, that with these parameters, the $\frac{1}{f^2}$ noise contributes about as much noise power as the white noise, while the flicker noise is negligible. We can also compare the

the \tilde{V}_{yy} element of the noise matrix by integrating the transfer function \tilde{T}_{yy} over the noise spectrum \mathcal{L}_k :

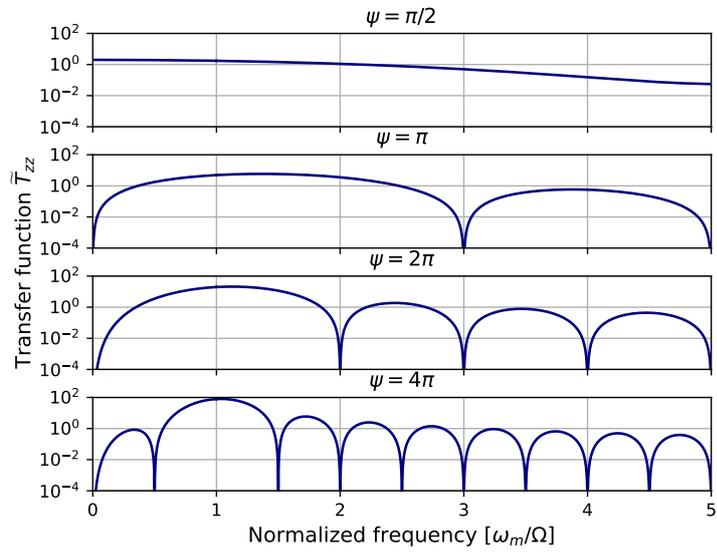
$$\begin{aligned}
 \tilde{V}_{yy,0}(\psi = \pi) &= \int_0^\infty \mathcal{L}_\circ \tilde{T}_{yy}(\psi = \pi, f) df = 9.87 \cdot 10^{-5} \\
 \tilde{V}_{yy,1}(\psi = \pi) &= \int_{f_{min}}^\infty \frac{\mathcal{L}_1}{f} \tilde{T}_{yy}(\psi = \pi, f) df = \int_{f_{min}}^\infty \mathcal{L}_\circ \frac{f_{c1}}{f} \tilde{T}_{yy}(\psi = \pi, f) df = 2.15 \cdot 10^{-8} \\
 \tilde{V}_{yy,2}(\psi = \pi) &= \int_{f_{min}}^\infty \frac{\mathcal{L}_2}{f^2} \tilde{T}_{yy}(\psi = \pi, f) df = \int_{f_{min}}^\infty \mathcal{L}_\circ \frac{f_{c1}^2}{f^2} \tilde{T}_{yy}(\psi = \pi, f) df = 7.20 \cdot 10^{-5}.
 \end{aligned}
 \tag{2.49}$$

Here I integrated also flicker noise up to infinity, as it is limited by the transfer function. White frequency noise contributes about as much to \tilde{V}_{yy} and thus to the gate infidelity for the worst case (certain final Bloch vectors \mathbf{J}_f°) as the white phase noise does. \tilde{V}_{zz} is not an issue since it has a zero at DC. Flicker noise is negligible.

Hence, white frequency noise could matter in certain circumstances. However, this example is likely the worst case. My assumptions are on the high side. Since the optical qubit lifetime of $^{40}\text{Ca}^+$ is about 2s, the fidelity would be limited by qubit relaxation rather than the phase noise after a second. Rabi frequencies will likely get rather higher than lower to speed up gates, where the relative effect of $\frac{1}{f^2}$ is decreasing, as it does not scale with Rabi frequency. Also, in terms of gate length, \tilde{V}_{yy} for the π pulse is the peak of the problem. For lower gate lengths, DC is always sampled by the pulse, but the transfer function decreases in magnitude. For longer gates, there is a zero at DC, eliminating the influence of slow noise. However, this analysis shows, that for short gates, the full integral of the noise profile with the transfer function has to be considered.



(a)



(b)

Figure 2.9: Transfer functions for four different gate lengths ψ . (a) $\tilde{T}_{yy}(\psi, \omega_m)$ (b) $\tilde{T}_{zz}(\psi, \omega_m)$

Chapter 3

Quantum Experiment Control at TIQI

In this chapter, I describe the current setup for quantum experiments at TIQI. In this thesis, this is the eQual setup. As it is of special interest, I additionally focus on the experiment control of the setup, the Quantum Experiment Next-Generation Control Hub (QuENCH) system. I will also briefly review the technical background needed to understand the setup and conducted measurements.

3.1 Programmable hardware

Qubit control requires the generation of several RF pulses in parallel, which have to be timed precisely relative to each other and with the correct pulse length. This is a challenging task for traditional computers. While a CPU supports a high throughput of arbitrary operations, they are not designed for meeting strict real-time requirements. Furthermore, the flexibility makes a CPU less efficient for highly specific tasks. In contrast, programmable hardware provides a solution for such high-throughput, real-time applications.

A widely used architecture of programmable hardware today is an field-programmable gate array (FPGA). It comprises configurable logic blocks (CLBs), containing combinatorial logic and flip-flops, interconnected through programmable connections. The FPGA is connected to the outside world with input output banks (IOBs). Certain hardware which is often used is also available as dedicated blocks, for example, digital signal processing (DSP) slices and block RAM (BRAM). The DSP slices contain multipliers and wide adders, while the BRAM allows storage of big blocks of data, which can be accessed quickly [37].

The advantage of FPGAs is that the hardware can be tailored to specific product requirements. This enables parallel processing of a lot of data since highly specific, individual logic blocks can work in parallel. However, programming an FPGA requires proficiency in digital design and differs from traditional programming paradigms due to its inherent parallelism. Moreover, since it is much closer to hardware, a lot of the higher abstraction and libraries provided by traditional programming fall away. Also, the synthesis and implementation of hardware, which is comparable to the compilation of code, is more complicated, as is debugging. Compared to application specific integrated circuits (ASICs), they are reprogrammable and do not require high upfront development efforts. However, FPGAs consume more power, operate at slower maximum clock speeds and have higher per-piece costs. Therefore, programmable hardware is ideally suited for situations where CPUs are not fast enough, necessitating custom hardware, yet demand flexibility in configuration

and involve small product volumes. Since we have eight experimental setups in the group and all of them require constant feature updates for new experiments, this is the case for qubit control at TIQI. Therefore, we use FPGAs for RF pulse generation.

3.2 Hardware description languages (HDLs)

To program an FPGA, an HDL is used. As the name implies, it does not define an algorithm like programming languages, but describes the logic and its interconnections within the FPGA. It is more a descriptive form of a schematic than a program.

There are two widely used HDLs, Very High Speed Integrated Circuit Hardware Description Language (VHDL) and Verilog. VHDL syntax resembles a more natural language. It tends to be very verbose, which makes it a little bit more tedious to write, but in turn tends to be self-documenting. It is a strongly typed language. Verilog on the other hand is closer to the C language, making it easier to learn when coming from a software background. It requires less code for the same functionality, also due to the weak typing, but needs more documentation to be able to understand the code [38, 39]. VHDL tends to be more popular in Europe, while Verilog is prevalent in the USA. Nevertheless, TIQI uses only Verilog for its projects.

In 2005, Verilog was extended with SystemVerilog. It provides a higher level of abstraction, making it easier to model systems of increasing complexity. These include object-oriented programming and interface abstraction. Also, it makes verification easier by supporting assertions, constraint random verification and code coverage [39]. SystemVerilog is a strict superset of Verilog, thus any Verilog file can also be used in a SystemVerilog project. However, since it is a lot of effort to adhere to the full SystemVerilog syntax, some tools do not support it or only parts of it. This includes some simulation software used by TIQI like Icarus Verilog [40]. Thus, TIQI has not yet fully switched to SystemVerilog.

3.3 Standardised interface for data converters (JESD204)

JESD204 is a standard defining an interface between data converters, such as ADCs and DACs, and the hardware processing them, such as FPGAs or ASICs. It accommodates the increasing data rate of converters resulting from increased resolution and sampling rate. Additionally, it reduces the number of pins required due to its serial design, leading to simpler and less expensive designs. The electrical interface utilises current mode logic (CML), which is more power-efficient at high frequencies than other interfaces like complementary metal–oxide–semiconductor (CMOS) or low voltage differential signaling (LVDS) [41]. Due to these advantages, JESD204 has become very popular.

The standard developed over the years, expanding features and speed. JESD204A brought support for multiple parallel lanes, JESD204B added deterministic latency and speed, while JESD204C tackles efficiency and further enhances throughput. Since we use JESD204B for the control system, I will focus on it.

In the JESD204B protocol, one logic device is connected to one or more converter devices. The converter devices can comprise several converters, given by the number M . Each converter device is connected to the logic device by one link, consisting of L differential signal pairs, called *lanes*. Each device is clocked by its own device clock, which in turn is derived from a common source clock. This configuration is depicted in figure 3.1. Other clocks are derived from the device clock, like the frame clock and the local multiframe clock (LMFC). These are used to arrange samples in frames and multiframe for higher

layers, where the frames and multiframes are aligned to the edges of these clocks.

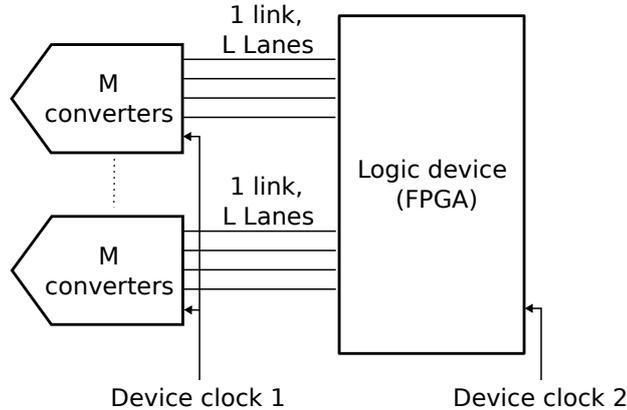


Figure 3.1: Illustration of devices within the JESD204B protocol.

JESD204B is split into three different subclasses: Subclass 0 to 2. They differ in the way they achieve synchronisation and deterministic latency. Subclass 0 does not support deterministic latency, but is mainly included in JESD204B to be backwards-compatible to JESD204A. It profits however from the higher line rates up to 12.5Gbps. In terms of synchronisation, the L lanes within one link are all aligned to each other, like in all other subclasses too. Inter-device synchronisation is not trivial, however. Several subclass 0 devices may be synchronised using a SYNC signal, which is a return path from receivers to transmitters. However, a separate interface is needed for that.

Subclass 1 devices are synchronised utilising a timing signal called SYSREF. It is source synchronous, thus it is distributed together with the common clock to all devices. The goal of the synchronisation is to align the edges of the LMFCs of all devices, such that all multiframes start at the same time. The LMFC period is an integer multiple of the device clock and the LMFC edge is aligned to the device clock. Initially, the individual LMFCs can be aligned to a different clock edge on each device, thus having a different phase. The SYSREF signal has a period of an integer multiple of the LMFC, thus also of the device clock. By sampling the rising edge of SYSREF, the phase of the LMFC can be determined and adjusted, such that all LMFCs are aligned. The SYSREF signal can be periodic, gapped periodic or one-shot. Since a periodic SYSREF signal is a sub-harmonic of the device clock, the other two modes might be preferable to reduce possible spurious effects on the converter.

For subclass 2 devices, the procedure is very similar, however, they use the SYNC signal instead of the SYSREF signal as a timing reference. For ADC devices, the LMFC clock is reset to the same device clock edge upon receiving a SYNC edge from the logic device. For DACs, the SYNC is sent from the converter to the logic device, where it determines the phase difference between the LMFCs utilising the sampled SYNC. In case of a difference, the logic device sends phase adjustment information to the DAC, which in turn adjusts the LMFC accordingly. In comparison to the ADC case, this requires several iterations. The advantage of subclass 2 devices is the simpler clock distribution network, since no SYSREF is required. However, the SYNC signal is system-synchronous. It is generated by the receiver and not the common clock source, like SYSREF. In the latter case, the distributed clock and SYSREF are skewed by roughly the same amount, making it easier to adhere to setup and hold constraints. For subclass 2 devices, the skew of the SYNC signal to the device clock sampling is influenced by the skew from source clock to both transmitter

and receiver, as well as the skew from transmitter and receiver. Thus, it becomes hard to adhere to timing constraints. Thus, subclass 2 is recommended for slower device clocks up to 500MHz.

After the LMFC clocks of all devices are aligned either with SYSREF or SYNC, the deterministic latency feature is implemented in the same way for subclass 1 and 2. The transmitter starts the initial line alignment at the same LMFC edge across all lanes. This initial edge is determined by SYNC for both subclass 1 and 2, but it is not timing critical in this case since the LMFC is slower than the device clock by a factor of 256 or more. After the initial line alignment, the receivers buffer the incoming data and release it on each lane after a well-defined amount of time known as the Rx buffer delay. Since it is fixed and referenced to an LMFC edge which is the same on all devices, it is deterministic. The Rx buffer delay must be longer than any possible delay in this implementation. The minimum Rx buffer delay can be determined by the maximum time it takes to receive valid data in the initial lane alignment phase over all lanes [42, 43, 41]. Some implementations even shorten the Rx buffer delay to the link delay variation by splitting it into a fixed and variable delay. The fixed part is covered by offsetting the edge of the LMFC of each link by the fixed part, so the buffer has to cover only the variance over power cycles [44].

Since deterministic latency is important for quantum control for precisely timed pulses, we need subclass 1 or 2. Subclass 1 supports higher device clocks and is, therefore, more capable of adjustment for future development, while the downsides are minimal. Thus, JESD204B subclass 1 is used as a data converter interface at TIQI.

3.4 The experimental setup

All my experiments were conducted on the eQual setup, where eQual stands for "encoded Qubit alive". It was mainly built by Robin Oswald [45] and Roland Matt [46]. As part of the IARPA "LogiQ" (Logical Qubit) program together with the group of Rainer Blatt from the University of Innsbruck (UIBK), the goal was to demonstrate a logical qubit. A logical qubit groups several physical qubits together to form a fault-tolerant qubit. Thus, the setup is suitable for experiments with many ions.

Ion traps are operated at an ultra-high vacuum (UHV) to avoid collisions of the ions with residual gas. Some traps achieve this at room temperature, but this requires UHV-compatible components and time-consuming vacuum chamber bake-outs. The eQual setup however is a cryogenic setup. If the trap is cooled down to about 5K within a cryostat, residual gas tends to stick to the cold surfaces. With the help of this "cryopumping", it is much easier to reach the required UHV. Additionally, many materials become UHV-compatible at cryogenic temperatures.

The lab holding the experiment is split into three areas: The control room, the ion trap room and the laser room. The lasers needed to manipulate the ions are located in the laser room, from where they are guided by optical fibres to the ion trap room, containing two cryostats with traps inside. The whole setup can be controlled from computers in the control room via Ethernet.

In the laser room are two enclosures for optics and two optical tables. Most lasers and their reference cavities, that is 732nm, 423nm, 397nm, 854nm and 866nm, are contained in one of the enclosures, a custom-built rack based on a commercial 19-inch rack. It is water-cooled to minimise temperature fluctuations, which could affect both cavity stability as well as optics alignment. The qubit laser however comes from a 729nm TiSaph laser on one of the optical tables. The second enclosure is dedicated to a high-finesse cavity and

optics to *lock* the 729nm laser, which means stabilising its frequency.

The ion trap room contains two cryogenic setups surrounded by an optical table each. The table is mostly used to deliver the beams into the cryogenic chamber, for power monitoring the laser beams and pulsing the laser used for coherent operations. The room also holds two racks, one for electronics and one for optics. The electronics rack contains the control system for the setups, in detail explained in section 3.5. It creates RF pulses, which are then sent to the optics rack. It contains mostly acousto-optic modulators (AOMs). The AOMs are placed in the path of the lasers before they go to the trap. They modulate the RF frequency coming from the control system onto the laser. The RF voltage generates an acoustic wave applied to a piezoelectric crystal. A laser sent through this crystal is diffracted and shifted in frequency by the frequency of the acoustic wave and thus the RF signal.

The cryogenic setup itself consists of an ion trap with a vacuum chamber surrounding it. A commercial closed-cycle cryostat reaching into the chamber cools an inner chamber containing the trap down to 5K. Several feed-troughs provide access for electrical signals and fibres. Electrical connections are needed for the DC electrodes and RF drive of the trap, as well as for piezo-actuators, heaters and the oven. Eleven optical fibres leading into the chamber allow individual addressing of ions. Other freespace lasers reach the trap through viewports, which also serve as an optical path for readout by the camera and the photomultiplier tube (PMT). An effusive oven in the chamber is used to load ions. It creates a flux of neutral atoms, which are then ionised by the 423nm laser and an additional 375nm laser.

The chamber can hold various ion traps, which have changed over time. My experiments were conducted with the latest trap called the Weizmann trap. It gets its name from the collaborative design and fabrication efforts with the group of Roee Ozeri at the Weizmann Institute. It is a linear Paul trap made of fused silica glass wafers. The electrode-ion distance is 150 μ m. The gold-coated electrodes are wire-bonded to the printed circuit board (PCB) it is placed on. A picture of the trap can be seen in figure 3.2.



Figure 3.2: The Weizmann trap, the ion trap used for experiments in this thesis. Figure adapted from [45].

3.5 The control system

The latest iteration of the control system in use at TIQI is called QuENCH, developed by Martin Stadler, Cagri Önal, Utku Altunkaya, Vlad Negnevitsky, Ilia Sergachev and others [47]. It is the successor of the previous Modular Advanced Control of Trapped IONS (M-ACTION) system by Vlad Negnevitsky [24] and is backwards compatible, since many experiments still use old hardware. At the beginning of this thesis, the eQual setup

was in the upgrade process for QuENCH. Since QuENCH RF generation hardware is indispensable for noise generation and the TI card, together with Martin Stadler we put significant effort into finalising the integration.

The core of the system is the main controller*, comprising a system on a chip (SoC) with two ARM CPUs and an FPGA. Experiments can be written in C++ and are compiled and linked into the application that is uploaded to the processor. A PC can change parameters and start experiments by connecting to the processor via Ethernet. The processor generates the instructions for a sequence of pulses according to the experiment and writes it to the FPGA. The FPGA is then responsible for time-accurate execution of the experiment. It does so by orchestrating the rest of the coherent electronics, most notably the RF generation cards.

The RF generation cards provide functionality to control the laser pulses, generating wave forms using direct digital synthesis (DDS) (see section 4.3 about DDS). They can be programmed with instructions for the pulse sequence containing phase, frequency and amplitude information before the experiment. During the experiments, the instructions are then triggered by the main controller. The QuENCH RF generation cards improve on some of the shortcomings of the previous generation. They drastically decrease the minimum pulse time, improve time resolution and double the channel density. Furthermore, they support several sine tones on the same physical output channel, removing the need for power combiners to add the output of two single-tone channels. This is useful in the MS gate experiments for the RSB and BSB. The cards consist of a carrier card with an FPGA[†], connected to two FPGA mezzanine card (FMC) cards with four channels each. One of these cards was replaced with the TI card for this thesis. The noise generator was implemented on the FPGA of the carrier board.

Both the main controller and the RF generation cards slot into a custom backplane based on the MicroTCA standard, which itself is in an enclosure which can be mounted into a 19-inch rack. Communication between the cards is handled by dedicated differential lanes each from the main controller to the RF cards and a backplane clock. The FPGA of the main controller also handles other coherent electronics, like the camera and PMT for readout, as well as arbitrary waveform generators (AWGs) to set the trap electrode voltages.

The experiment is controlled from the control PC with a GUI called Ionizer2. It connects to a server on the main controller over Ethernet. It queries the available experiments from the application running on the main controller and creates a separate page for each experiment, where experiment parameters can be set. Each experiment generates pulse sequence instructions that are loaded into memory blocks of the FPGA when the experiment is started. The results of the measurements are plotted in one or several windows and the underlying data is saved for further processing. Ionizer2 also has a Python API to automate longer experiment sequences with Python scripts.

Ionizer2 controls other, non-time-critical devices in the setup via plugins. These include fixed-frequency RF sources, shim electrodes to compensate stray electric fields in the trap, piezo-actuators to adjust the lengths of optical cavities and control loops for frequency and intensity stabilisation of the lasers. For the asynchronous control, the plugin communicates with single-board computers (Raspberry Pi's) over the Ethernet network. The Raspberry Pi's in turn control the asynchronous devices. A full overview of all parts of the control system is shown in figure 3.3.

*Avnet ZedBoard.

†AMC FMC carrier Kintex (AFCK).

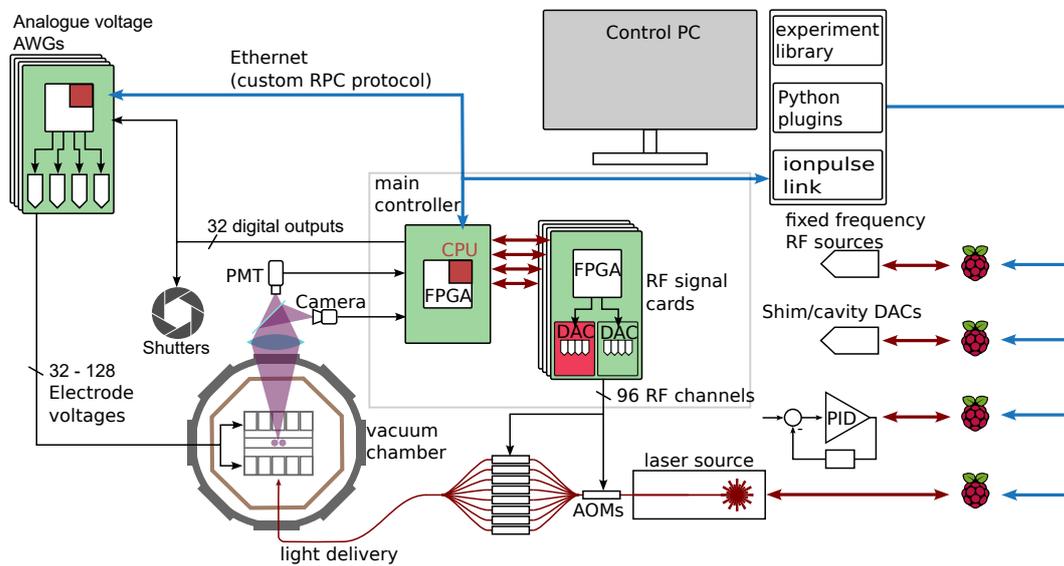


Figure 3.3: An overview of the QuENCH control system. The TI DAC card is highlighted in red. Image by Martin Stadler.

Chapter 4

Implementation

Utilising the experimental setup and control system, we want to quantify the influence of phase noise on the gate fidelity of quantum gates. That requires the possibility of injecting excess noise into the RF signals. Precise and flexible control of the power and profile of the injected noise is important. Precisely knowing the input parameters allows us to derive an exact dependency on the noise. The flexibility of generated noise parameters allows us to test different hypotheses based on theory and simulation. Also, the AFE8000EVM, the high-performance transceiver card, which Texas Instruments provided us with, had to be integrated into the current control system. Thus, several changes to the hardware, firmware and software of the QuENCH control system were needed for this thesis.

4.1 DAC upgrade - AFE8000

The AFE8000EVM is an evaluation board for an RF sampling transceiver chip, the AFE8000 from Texas Instruments [48]. The acronym AFE stands for "analogue front end". The AFE8000 is a high-performance, wide bandwidth multi-channel transceiver, primarily used for high-frequency RF applications like radar systems, software-defined radio or wireless communication testing. It features eight 14-bit DACs and ten 14-bit ADCs, of which we only use the DACs. Its DACs sample with 9GSPS in straight mode, which could go up to 12GSPS in interleaved mode (a mode where the output of two parallel DACs is combined to increase the sampling rate). The output frequency is 5-7125MHz, with a maximum signal bandwidth of 800MHz (for 8 channels). Each transmitter chain can be attenuated by a digital step attenuator with a range of 40dB. It also features a digital upconverter, which means that samples can be provided at a lower sampling rate and upconverted to a higher frequency. This is a useful feature for us, as it allows us to generate samples at a lower frequency, saving us resources on the FPGA. The AFE8000 also features a high-speed data interface supporting the JESD204B and JESD204C protocol. These protocols feature deterministic latency, which is important for real-time control of pulses.

The AFE8000EVM also includes the LMK04828, a low-noise clock jitter cleaner to generate a clock for both the transceivers as well as the JESD204 protocol. It features a dual phase locked loop (PLL) designed to multiply a clock signal sourced from an internal quartz oscillator while simultaneously minimising excess noise. This removes the need for a high-frequency reference clock, just a 10MHz reference signal is needed to synchronise the LMK04828 with the main reference. The interface to a carrier card is realised by a high pin count FMC connector.

As mentioned, the AFE8000 is designed for high-frequency applications in the GHz range,

although our requirements only necessitate RF frequencies up to about 400 MHz. Thus, the evaluation board sent to us was modified by Texas Instruments to adjust the reconstruction filters for four of the channels for these lower frequencies. We also had to make some minor adjustments ourselves. First, we had to desolder the four edge-mounted SMA connectors to fit the card into the MicroTCA chassis. Furthermore, the AFE8000EVM, being an evaluation board, is notably long. Hence, the FMC connector alone cannot be expected to bear the mechanical load of the card being mounted vertically to the AFCK slotted into the chassis, similar to the regular front-end cards. To address this, I designed a small mechanical fixture consisting of a metal plate which can be screwed to both the AFE8000EVM and the AFCK, providing additional support. However, the card still protrudes several centimetres beyond the front panel plane, which is not ideal. If the card would be used more extensively in the future, it would have to be integrated more densely to shorten the card. Front-facing edge-mounted SMA connectors would also be useful, since in the current setup, the adjacent slot has to be empty to be able to access the SMA connectors. It is probably a challenging task to comply with these requirements, while still keeping the very high channel number. A picture of the AFE8000EVM is shown in figure 4.1.

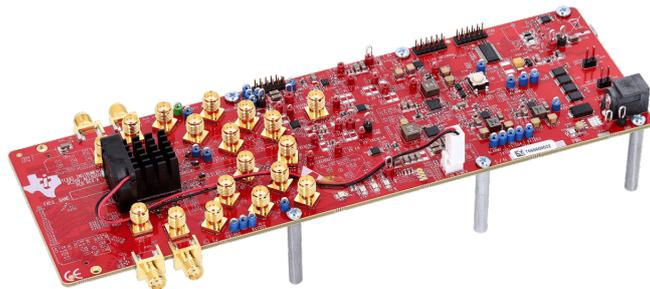


Figure 4.1: The AFE8000EVM evaluation board. Picture taken from [49].

4.2 Noise generation

An important part of the implementation was the ability to inject well-controlled noise into the RF signals. Since I wanted flexible noise profiles and power levels, the easiest way would be to synthesise it on board. One option could have been to utilise an arbitrary waveform generator and inject the synthesised noise either by modulating it onto the RF signal or directly onto the laser using an AOM [8]. However, this would necessitate additional hardware and more modifications to the eQual setup. Since we generate the RF signals anyway on custom hardware (see section 4.3), the FPGA of the RF signal cards, the most straightforward solution is to generate the noise directly on the FPGA and add it to the phase of the sine-wave synthesis. Since the upconverter feature of the TI card frees some FPGA resources, resource constraints should not be a problem.

Indeed, generating noise on an FPGA is not a trivial task. High-level programming languages usually come with a large library of algorithms, including pseudo-random number generation. This is not the case on an FPGA. Usually, the FPGA developer implements functional blocks on a hardware level to utilise resources as efficiently as possible. There are implementations of sophisticated noise generation algorithms [50], but the complexity of these algorithms often comes with a high demand for FPGA resources.

Thus, it is helpful to think about the requirements on our noise to find a reasonable balance, since we want to add a noise generator to each of the eight channels supported by one RF generation card. Our goal is to simulate real phase noise as close as possible. This translates to noise which is

- (pseudo-) random,
- Gaussian-distributed in the time domain and
- ideally encompasses the three noise profiles described in section 2.2.1 in frequency domain.

It is hard to create true randomness on an FPGA, since logic is deterministic [50]. Luckily, we do not need a truly random signal, a pseudo-random signal is good enough. A pseudo-random sequence exhibits statistical randomness, despite being generated by a deterministic algorithm. While this might be insufficient for applications in cryptography, in our case we only care about its statistical properties.

A simple method to create pseudo-random numbers on an FPGA is a linear feedback shift register (LFSR) [51]. It is a shift register where the input bit is a linear function of the previous bits. By appropriately selecting the feedback function, the LFSR can cycle through all possible states (excluding zero), with the last bit of the current state serving as the output. This results in a pseudo-random bitstream. The architecture of an LFSR can be represented by its characteristic polynomial over the integers modulo 2 in finite-field arithmetic [52]. In our case, the polynomial is $P(x) = x^{31} + x^{28} + 1$. The feedback function takes values from the register cells whose corresponding monomial in the characteristic function is non-zero, known as *taps*. In our case, these taps are 28 and 31, and they are fed back to the input using an XOR function. A graphical representation of the LFSR implementation can be seen in figure 4.2.

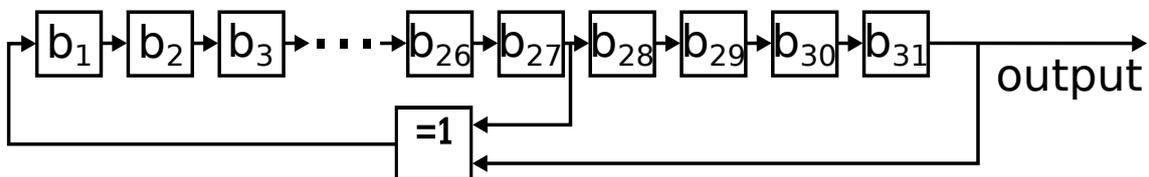


Figure 4.2: The architecture of an LFSR. The output is the last bit of a shift register, where the input is a function of the previous state.

The length of the register determines the period of the pseudo-random number sequence. With a register length of $n = 31$ bits, and the LFSR cycling through all states except the the all-zero state before repeating itself, the period is $T = 2^n - 1$. However, we require the LFSR to generate a random number of length $L = 14$ bit each clock cycle, whereas the output of the LFSR is only one bit. One approach is to take a longer substring of the status register as an output. This would still result in a uniform distribution, but the subsequent numbers are highly correlated, since each one is the previous one shifted by one plus an additional bit. Thus, a solution is to advance the LFSR by L every clock cycle, which can be done with replication. Note that $n = 31$ is conveniently chosen such that the period T is a Mersenne prime number. Thus, even though the LFSR advances faster by a factor of $L = 14$, the periodicity T does not change, since T and L are co-prime and therefore the LFSR still cycles through all states before repeating itself [53]. Interestingly, this means that a bigger LFSR is not always better. Increasing the register length to $n = 33$, for example, would decrease the total periodicity, as $\gcd(T, L) = 7$ while T is only increased by a factor of four. In total, the chosen register length of $n = 31$, using an FPGA

clock period of $T_c = 4\text{ns}$, equates to a periodicity of $T = T \cdot T_c = 8.6\text{s}$, which is significantly longer than every individual experiment.

The generated number sequence using an LFSR is uniformly distributed in the frequency domain, which is desirable, but the amplitude is also uniformly distributed in time domain. This is an unnatural distribution, as most noise exhibits a Gaussian distribution (see section 2.2.1). Therefore, we aim to convert the uniform noise to Gaussian noise. One approach for this conversion is the Box-Muller transform [54]. However, this transform uses non-arithmetic functions, which have a higher resource demand. Thus, I made use of the central limit theorem instead. It states that the mean of independently distributed variables converges to a Gaussian distribution, regardless of the distribution of the individual variables. Therefore, I add the result of $n = 4$ parallel 14-bit LFSRs together, resulting in 16-bit Gaussian-distributed white noise (see figure 4.6). Although the theorem claims Gaussian distribution only in the limit of $n \rightarrow \infty$, $n = 4$ already provides a reasonably good approximation. The resulting distribution and a comparison to the ideal Gaussian distribution are depicted in figure 4.3a. To better see the difference between the two distributions, especially tiny differences in the tails of the Gaussian distribution, I use a quantile-quantile (Q-Q) plot (see figure 4.3b). It is a graphical method for observing the similarity of two distributions. The samples of the distribution in question are split into quantiles and plotted against the quantiles of a reference, usually an ideal model, on a scatter plot. The closer the two distributions are, the closer they are to the identity line. Here, it can be observed that for values within about $\pm 23\text{k}$ ($\pm 2.5\sigma$), the synthesised distribution closely aligns with the ideal one. However, in the tails, the approximation breaks down due to the limitation imposed by the total sum of four 14-bit numbers being limited to a 16-bit number (ranging from -32768 to 32767). Despite this limitation, the distribution is assumed to be sufficiently good since both theory (section 2.2.2) and simulations do not indicate a dependence of the gate fidelity on the distribution in the time domain.

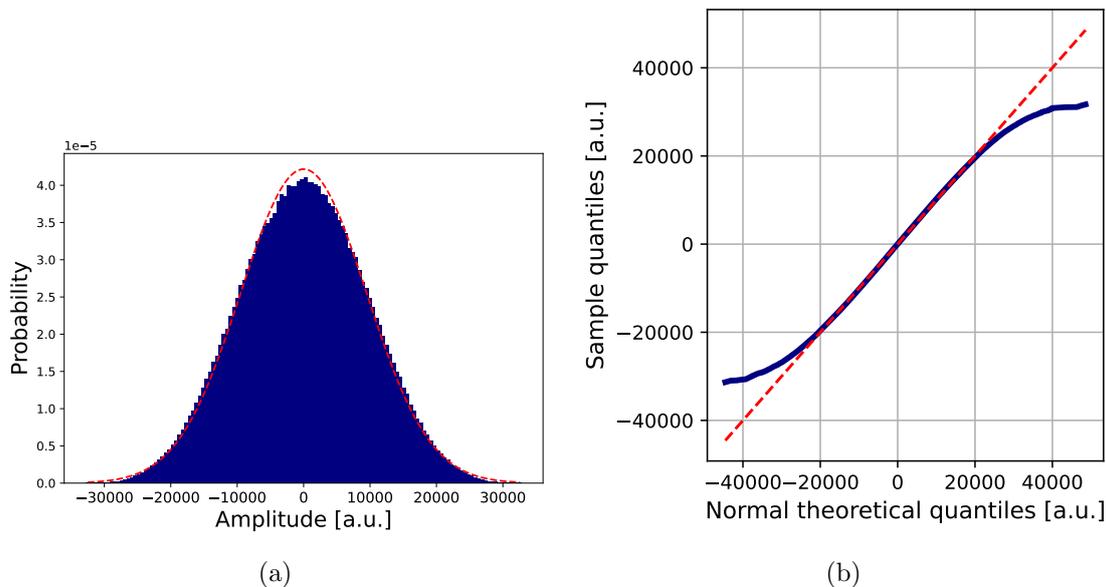


Figure 4.3: (a) Histogram of the sum of four uniformly distributed numbers generated by an LFSR, with the ideal Gaussian distribution as a comparison (red dashed line). The bin count is 128. (b) The quantile-quantile plot of the generated distribution versus the ideal Gaussian distribution.

Since we want to be able to generate different noise profiles, the white noise has to be modified. To do this, I pass the white noise through a filter with the appropriate frequency response. Such a filter can be constructed using a PI (proportional-integral) controller. It is typically used to steer a system to a desired outcome by comparing the fed-back output of the system to a setpoint. The setpoint is set to the generated white noise in this case. The controller consists of a proportional component and an integral component. The proportional part multiplies the error signal (the difference between the setpoint and the feedback) with a constant K_P , thereby generating white phase noise at the output, adjusted in the power level by K_P . The integral part integrates the error and multiplies it with K_I for scaling. Since the phase is the derivative of frequency, this creates random phase walk or white frequency noise ($\frac{1}{f^2}$ dependence). The feedback is the time-delayed, scaled-down output. Without it, the output would eventually wander out of the numerical limits of the output of the controller. It exhibits a high-pass behaviour with a corner frequency of about 100Hz, countering the integrator up to those frequencies. The controller with feedback is depicted in figure 4.4.

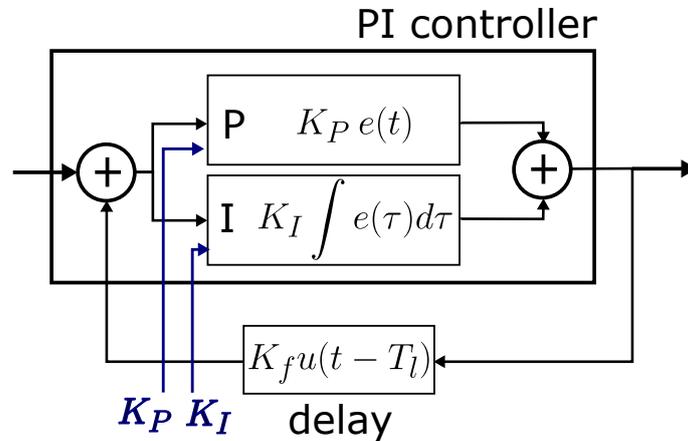


Figure 4.4: The implemented filter. It uses a PI controller with a feedback structure to prevent the output from exceeding bounds.

This filter design implements white phase and frequency noise, but not flicker noise. Flicker noise is more complex to generate than the other two. One possible solution is an infinite impulse response (IIR) filter. This approach approximates a slope of $\frac{1}{f}$ by alternating poles and zeros. Increasing the number of poles and zeros allows for fitting to arbitrary accuracy [55]. Another approach is the Voss-McCartney algorithm [56], which involves adding several random white noise sources. Each source is updated at progressively lower frequencies, resulting in a mixture of white noise (where the output is replaced by a new value every clock cycle) and $\frac{1}{f^2}$ noise (where the output is a sum of *all* previous inputs). Both approaches are more resource-intensive than generating white noise or white frequency noise, but they are not unreasonably so. However, I decided to not implement flicker noise because of the time needed to implement it. Its priority did not seem to be as high in comparison to other tasks, as simulation indicates that noise profile, especially flicker noise, does not matter that much (see chapter 5).

I added an additional low-pass filter with a corner frequency of 7.85MHz. The purpose of this filter is to limit noise power at frequencies well above the relevant frequencies (Rabi and trap frequency), since high noise power creates other problems with measurement (see chapter 5). Since we want to cover a big range of noise levels, it is not a good idea to adjust the power level with only the filter constants K_P and K_I . Very small values for low noise levels lead to quantisation errors. Therefore, I added a global scaler. It does not

multiply the noise with a constant using a DSP slice; rather, it shifts the value, thus scaling it by powers of two. This allows it to cover a large range without increasing the bit width of K_P and K_I . It shifts the total frequency response, including the I and P components and the high-pass behaviour of the feedback structure. In the end, the synthesised noise is added to the ideal phase to create a noisy phase used in sine-wave synthesis (section 4.3). An example of generated noise is depicted in figure 4.5. The complete noise generator is illustrated in figure 4.6.

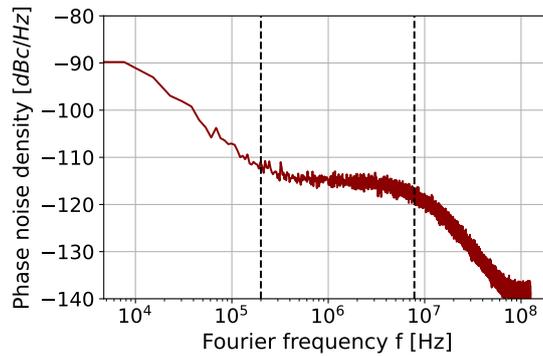


Figure 4.5: Example of simulated noise output. The first dashed line is at the corner frequency of the PI filter (set by filter parameters), while the second dashed line is at the corner frequency of the low-pass filter (fixed, 7.85MHz).

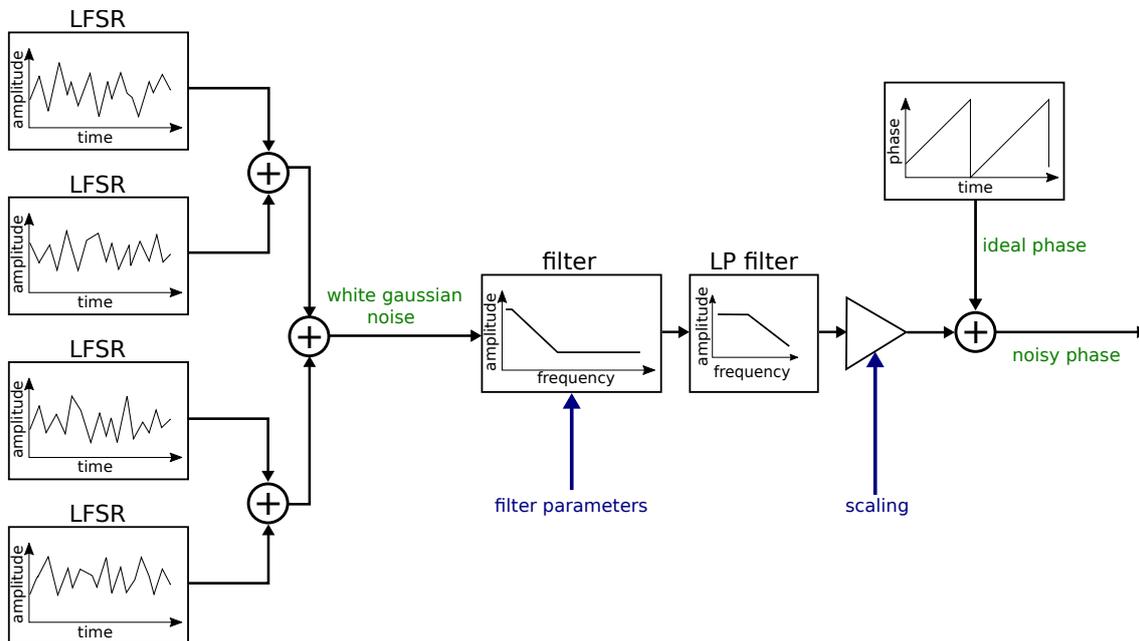


Figure 4.6: The noise generation block implemented in firmware.

4.3 On-board complex sine-wave synthesis

The QuENCH system incorporates RF generation through direct digital synthesis (DDS). This entails digitally creating a continuous stream of sine-wave samples using the FPGA of the RF generation card, which is then sent to the DAC for conversion into an analogue signal. I will now provide a brief overview of such an implementation in firmware.

The basis is a register representing the instantaneous phase of the signal. A *phase* parameter sets the initial phase offset. Since the instantaneous phase is the integral of the frequency, it is advanced at each clock cycle by the clock period multiplied by the current frequency, specified by the *frequency* parameter. This block is therefore known as a *phase accumulator*. The register only needs to hold the phase ranging from 0 to 2π , since sinusoids are 2π periodic, and any overflow acts as a modulo- 2π function.

Subsequently, the instantaneous phase is converted to a sinusoid using a mapping of phase to amplitude in a memory block called a lookup table (LUT). It is unnecessary to store a whole period of a sine though, as a sinusoid exhibits a four-fold symmetry. Since the saved sinusoid in the LUT is always full scale to minimise quantisation error, the amplitude is scaled down to the desired value by a constant *amplitude* parameter.

These samples generated at every clock cycle are then sent to a DAC for conversion to an analogue signal. In our setup, where the samples are generated on the FPGA on a carrier board while the DAC is on a dedicated mezzanine board, the samples are transferred over the FMC connector connecting the carrier board with the DAC card with a high-speed data interface called JESD204B. An overview of the sine-wave synthesis is provided in figure 4.7.

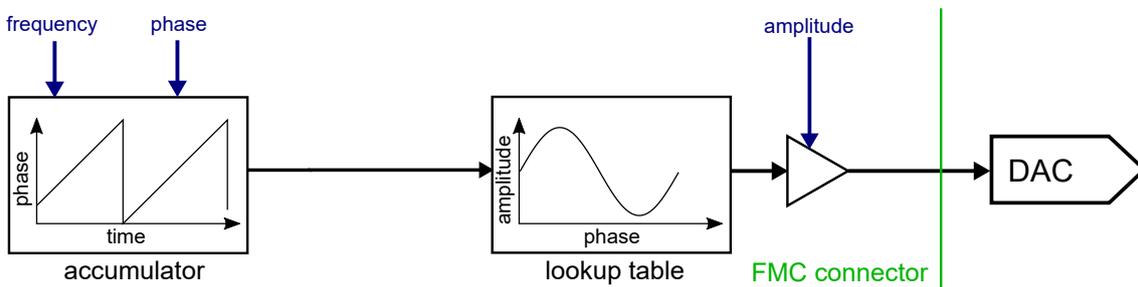


Figure 4.7: Architecture of sine-wave synthesis for the previous QuENCH DAC card. Figure adapted from Martin Stadler.

An advantage of this on-board synthesis compared to a fixed-function RF signal generator, where just the frequency, phase and amplitude are set, is the great flexibility it offers. The QuENCH system enables the generation of not just one tone, but up to four tones on the same channel by digitally combining the outputs of several synthesisers. In contrast, combining tones using fixed-function signal generators requires multiple channels, which then need to be combined using analogue power combiners. Alternatively, AWGs can be used, which would also allow for any pattern including many tones. However, the tight integration of the QuENCH system avoids long delays during experiments due to sample upload time. For this thesis, the flexibility of the QuENCH RF generation allows the value of the phase accumulator to not only be influenced by the phase and frequency parameters but also by the generated instantaneous phase noise. Thus, the injected phase noise is already included in the sine-wave present at the DAC output and does not have to be added at a later point, requiring additional hardware.

For this thesis, Texas Instruments has provided a DAC card to replace the currently used

QuENCH DAC card (see section 4.1). It features a built-in upconverter. Thus, it can accept samples at a lower frequency (baseband), which are then upconverted to a higher frequency (RF frequency). However, to preserve phase information during the upconversion, the samples must be provided as complex samples. In practice, this is achieved by providing both an *in-phase* (I) and a *quadrature* (Q) component. They represent the real and imaginary parts of a complex value, rather than the amplitude and phase information (see 4.8). During upconversion, the upconverter multiplies these components with the sine wave coming from a local oscillator (LO), where the phase of the LO for the Q component is shifted by 90° . The LO is a digitally generated sine wave exactly like described before, called the numerically controlled oscillator (NCO).

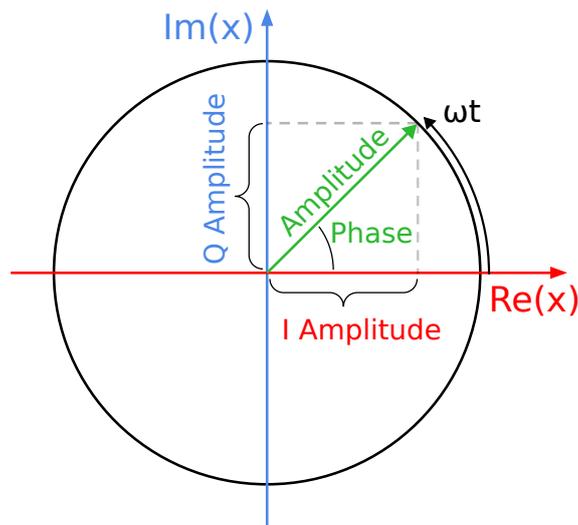


Figure 4.8: A complex phasor represented in the complex plane with I and Q components. The phasor advances in time in the anticlockwise direction.

Since the TI DAC card now requires I/Q samples, the sine-wave synthesis has to be adjusted accordingly. The phase accumulator stays the same, as the phase is used for both the I and Q samples. Subsequently, the path is divided into separate I and Q paths, where the phase of the Q path is shifted by -90° . To motivate this, see the rotation of the phasor in figure 4.8 through the complex plane, with the I component as the real part and the Q component as the imaginary part. The I component is the cosine $\cos(\omega t)$, while the Q component follows a sine $\sin(\omega t)$. As time and therefore angle moves anticlockwise, the Q component is lagging 90° behind. We can write the upconversion also as a complex multiplication of the LO phasor $e^{i\omega_{LO}t}$ and the I/Q phasor $e^{i\omega t}$:

$$\begin{aligned} x(t) &= \Re\{Ae^{i\omega_{LO}t}e^{i\omega t}\} \\ &= A \cos(\omega_{LO}t) \cos(\omega t) - A \sin(\omega_{LO}t) \sin(\omega t), \end{aligned} \quad (4.1)$$

where ω_{LO} is the LO frequency and ω is the frequency of the I/Q signal. We can observe that the unshifted LO ($\cos(\omega_{LO}t)$) is multiplied with the I sample, which is a cosine ($\cos(\omega t)$), while the 90° -shifted LO ($-\sin(\omega_{LO}t)$) is multiplied with the Q sample, which is a sine ($\sin(\omega t)$).

For each of the signal paths, the phase has to be converted to an amplitude. Naively, with a fixed number of ports per LUT, this would require double the amount of LUTs, each

saving the sine or cosine. However, we can take advantage of the displacement symmetry of the sine and cosine. Previously, we needed to save a quadrant of the sine function to reconstruct the full sine over a period. However, it is also sufficient to save an octant of both the sine and cosine functions in the same memory entry. This approach doubles the memory width but halves the depth, thereby keeping the memory size the same. Since both the sine and cosine values are retrieved with one memory call, one memory block of the same size can accommodate both the I and Q paths. For phases not in the first octant, the sine and cosine amplitudes in the first octant can be switched and mirrored across the x- and y-axis to retrieve the desired amplitude in that octant [57].

The amplitudes of both the I and Q paths are then scaled down by the amplitude parameter, similar to before. However, this multiplication needs to be performed individually, requiring an additional multiplier. The generated I and Q samples are then again sent to the TI DAC card over the FMC connector using JESD204B. A depiction of the total signal path can be seen in figure 4.9.

As observed, although there are now two signal paths instead of one, most of the resources can be shared between them. The effective increase in resources is only one additional multiplier requiring a DSP slice for the final amplitude adjustment, plus some minor logic. However, the samples can be provided at a lower rate — in our case the FPGA clock frequency of 250MHz — compared to the previous rate of 1GHz. Sampling for a 1GSPS DAC at 250MHz, as previously done, requires hardware replication to generate four samples per clock cycle, which can now be omitted. Consequently, we reduce the resources used for sine-wave synthesis, down to almost a quarter, thanks to the upconverter of the TI card.

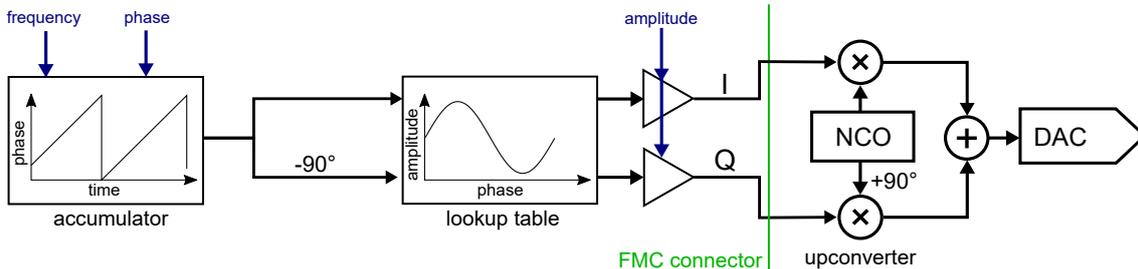


Figure 4.9: Architecture of sine-wave synthesis modified for this thesis, accommodating the upconverting feature of the TI DAC card.

4.4 Clock generation and distribution

The programmable logic within the FPGA of the RF generation card is clocked at various speeds. Most logic operates with the management clock at 125MHz to relax timing constraints. However, certain parts, namely the sample generation and the GTX transceivers used for the JESD204B protocol, run at higher speeds. Some other frequencies are also used in other modules of firmware, but since these were not altered in the scope of this project, I will not cover them here.

To test the TI card, Texas Instruments also provided an example design using a Virtex 7 FPGA VC707 Evaluation Kit along with accompanying example firmware, aiming to simplify the integration of the card. At the core of this design is a custom JESD IP core from Texas Instruments, which implements the JESD204 protocol*. Since the TI card already has an internal reference†, the firmware of the example design is entirely clocked by

*The TI 204c IP core [43].

†The Texas Instruments LMK04828, a dual PLL [58].

references provided by the TI card over the FMC connector. This includes both the system clock and the clock for the GTX transceivers. I ported the JESD core to the QuENCH system. However, this process was not seamless, as the schematics of the hardware of the example design and the QuENCH RF generation cards differ.

To understand the problem, we have to look at the architecture and physical placement of the GTX transceivers [59]. The GTX transceivers used by the used FPGAs combine four lanes to form a quad. Each quad also physically contains two high-speed reference clock inputs to clock each of them. The clock can alternatively be sourced from one of the two adjacent quads. The example design uses eight lanes ($L=8$). Thus, the JESD core uses two quads. In the example design on the Virtex 7, the used quads are positioned next to each other, allowing them to be clocked by the same reference. This is not the case for the QuENCH system, however. The lanes, hardwired on both the TI card and the RF generation card, are now connected to two quads that are not adjacent to each other. The transceiver reference clock from the TI card is directly wired to one of these quads and therefore cannot clock the other quad. This situation applies to both FMC connectors available on the RF generation card, so switching to the other connector does not resolve the issue. Figure 4.10 illustrates the problem. The TI card on the left is connected to the RF generation card via the FMC connector 2. Within the RF generation card lies the FPGA with its four quads and the two clock input buffers for their assigned clocks, positioned according to the physical layout. Note that all clocks within the TI card and the RF generation card are differential up to these clock buffers, although this is not depicted for simplicity. The hardwired transceiver reference clocks are depicted in yellow. Not shown here are the transceiver lanes, where quad 0 and 2 are mapped to the FMC connector 2, while quad 1 and 3 are mapped to the FMC connector 1. Since I connected the TI card to FMC connector 2, the JESD core uses therefore quad 0 and 2.

As also shown in figure 4.10, the RF generation card includes a crossbar switch. It can map different input clocks to different output clocks. However, the transceiver reference clock is not connected to the crossbar switch and therefore cannot be rerouted to clock the used quads. Additionally, no other reference from the TI card is routed over the crossbar switch, making it impossible to use the TI card to clock the GTX transceivers. As a result, an external reference is needed. For this, we use a clock generation board[‡]. It contains a dual-loop PLL, providing versatile clock generation. One output is then used to feed a 250MHz clock over the U.FL connector into the RF generation card. Another clock output is connected to the 10MHz reference input of the TI card to synchronise its internal reference to the external oscillator. A third 250MHz output is connected to the main controller of QuENCH. This clock distribution concept is illustrated at the bottom of figure 4.10.

The reference connected to the RF generation card is internally connected to the crossbar switch, where it is forwarded to both quads individually. Since two separate clocks are now needed, the JESD core, in particular the Xilinx GTX IP core[§] within it, had to be slightly modified to accommodate this. Furthermore, the main system clock was previously also derived from the transceiver clocks. However, since the JESD core instantiates the differential clock input buffers within the core itself and does not output the single-ended clock for use outside of the core, this is not possible. This may be solved with another configuration of the GTX IP core, but I rather connected the external reference over the crossbar switch to yet another input buffer, from where it is forwarded to the clock manager to generate the main 125MHz management clock. Note that the instantiation

[‡]The HMC7044 evaluation board [60].

[§]The LogiCORE IP 7 Series FPGAs Transceivers Wizard.

of the differential input clock buffers by the JESD core has another interesting side effect. Previously, in the QuENCH firmware, the input/output (I/O) of each of the FMC connectors, including the reference clocks, was routed through an I/O module. However, Vivado is unable to place a differential input buffer when the differential lines are routed through a module, even if the wires from the input are directly assigned to an output. Bypassing the I/O module solves this problem.

Not all clocks are derived from the external reference. The sample clock for the JESD core and for the RF generation are still derived from the TX system clock coming from the TI card. However, I increased the sampling frequency of the JESD core from 125MHz in the example design to 250MHz by re-configuring the GTX IP core embedded in the JESD core. This leads to slight timing violations of about 0.02ns total negative slack. I could not investigate them further, since they appear within the encrypted part of the JESD core by Texas Instruments. However, the timing violations do not seem to be a problem in practice.

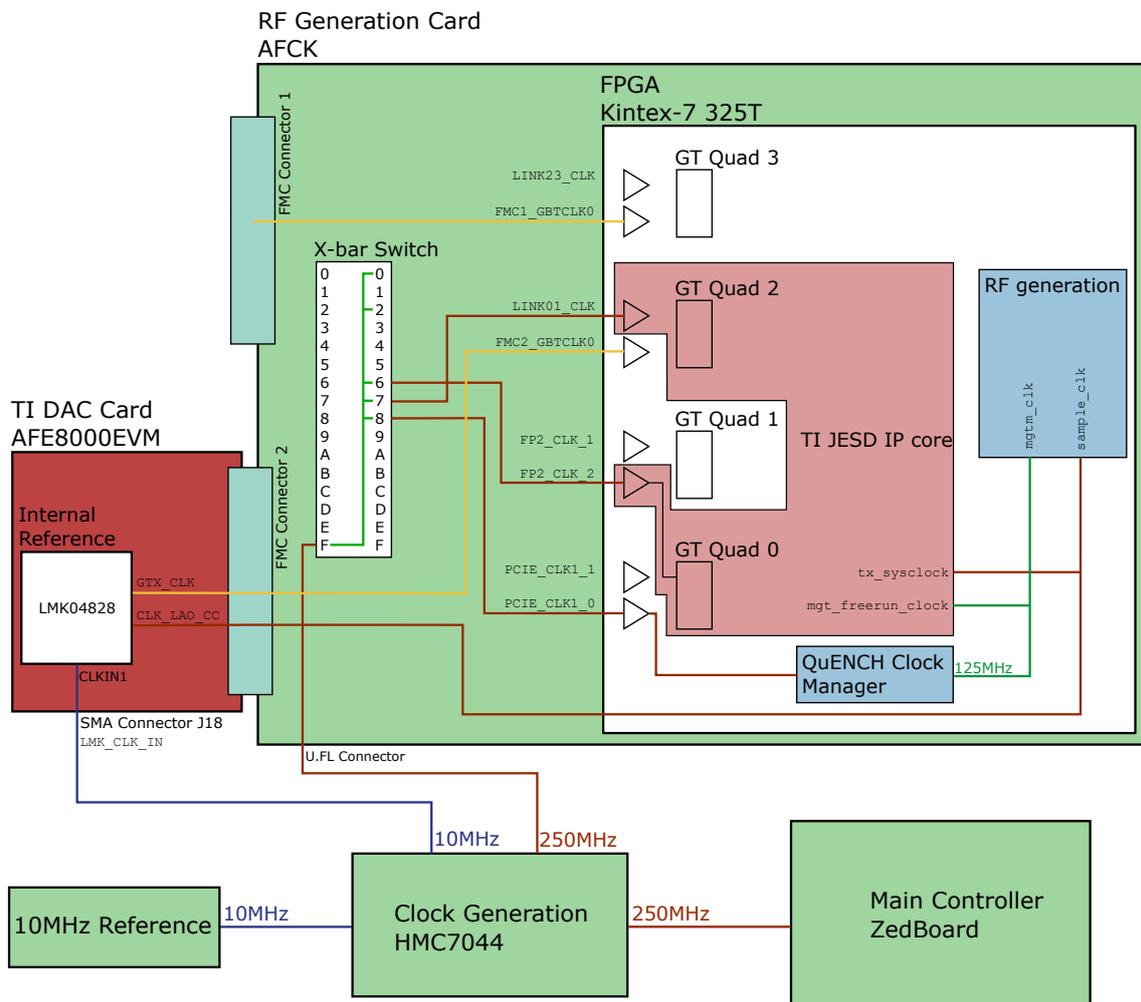


Figure 4.10: The clock distribution concept for the QuENCH system with the TI card. The figure also includes signal names from the QuENCH firmware and net names of the schematics of the TI card and the RF generation card in monospace font. While these are not important to understand this section, they may be helpful for developers continuing on the project in understanding the QuENCH firmware.

Chapter 5

Results

In this chapter, I present the main findings of this thesis. Firstly, I measure the inherent phase noise of the system and confirm the correctness of my implementation of phase noise injection. Furthermore, I present the results of simulations investigating the influence of phase noise on gate fidelity for both single- and multi-qubit gates. In the single-qubit case, the simulations are then compared to measured fidelities from the eQual setup and to the theoretical values from [9] (see also section 2.2.2).

5.1 Phase noise measurements

We confirmed the correct operation of the noise generation by measuring the phase noise with a signal source analyser. Hereby, I thank Sandro Camenzind from the Ultrafast Laser Physics group, who allowed us to use their Agilent (Keysight) E5052B and also helped us conduct the measurements. All measurements were conducted at a carrier frequency of 150MHz at -6dBFS. Initially, we measured the SSB phase noise of the reference clock* and the inherent phase noise of a sinusoidal at the output of the TI DAC card[†] (see figure 5.1a). The noise spectral density of the DAC itself is specified as $NSD = -157\text{dBFS/Hz}$ and is therefore much lower than the noise density of the reference clock up to the MHz range. Consequently, the noise performance of the TI card is likely limited by the reference clock. This also suggests that the noise measured at the output mostly reflects the noise of the internal reference clock[‡] employed in the TI card. Given that the external reference clock is a dual PLL very similar to the internal one, the traces look very similar. However, a peak at about 200Hz is evident on the TI card, followed by lower phase noise than the input from the reference clock. This discrepancy can be attributed to a lower PLL1 loop filter bandwidth of the internal reference compared to the external one. In the PLL2 loop bandwidth, the internal reference seems to add some noise overall. We note that the total phase noise characteristics of the TI card are well below what would impact our measurements, as demonstrated in the following sections.

We further tested the noise generation. Note that the following measurements exclude the low-pass filter mentioned in 4.2. In figure 5.1b, three traces each of white phase noise ($K_P = 1$) and white frequency noise ($K_I = 1$) are illustrated. These traces differ from each other by the global gain G , which shifts them up or down. For the white noise traces, it can

*HMC7044 evaluation board EVAL-HMC7044 [60, 61].

[†]The AFE8000EVM provided by Texas Instruments. [48, 62]

[‡]LMK04828 [58].

be observed that they are limited by the base noise from the TI card for frequencies below about 400Hz. However, this is not a problem, since relevant frequencies are well above that limit. A mixture of the two noise profiles can be seen in figure 5.1c. This synthetic noise profile most closely resembles the noise profile seen on actual devices (compare with 5.1a). By adjusting the proportional ($K_P = 1$) and integral gain ($K_I = 1$), the quantitative parameters of the profile can be changed. A comparison of the white phase noise, white frequency noise and a mixture of both with the ideal calculated transfer function can be seen in figure 5.1d. They agree very well up to about 10MHz, where a drop of about 0.5dB can be seen, probably the roll-off of the Nyquist filter.

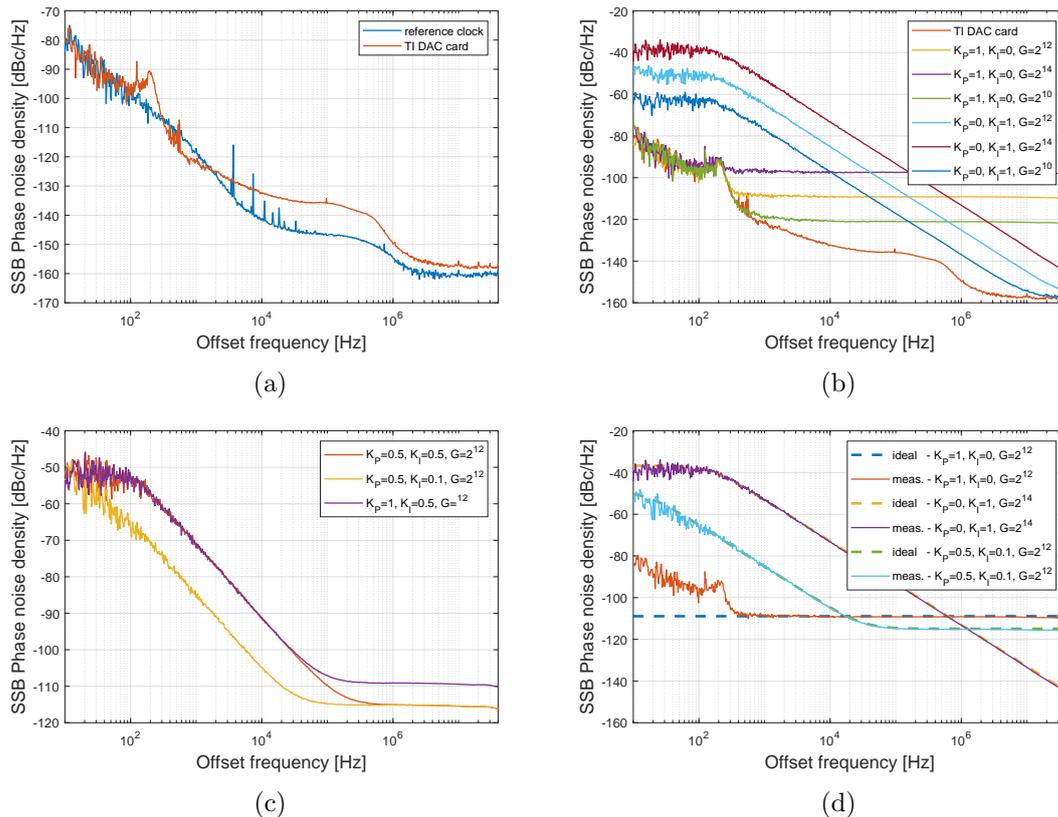


Figure 5.1: (a) Phase noise measurements of the reference clock and the output of the TI DAC card.(b) Measured phase noise with different levels of injected white phase noise and white frequency noise. The base noise from the TI card is shown as a reference. (c) Measured phase noise of a mixed white phase and frequency noise profile with different filter parameters. (d) A comparison between the calculated filter transfer function (dashed line) and the measured phase noise (solid line) for white phase and frequency noise and a mixture of both. All measurements were conducted at a carrier frequency of 150MHz.

5.2 Simulation

I simulated the impact of phase noise on the gate fidelity for both single- and multi-qubit gates. All simulations were conducted using the quantum simulation package *QuantumOptics.jl* [12] in the programming language Julia [11]. For both types of gates, I initially generated Gaussian white noise, which I filtered to achieve the desired noise profile in the Fourier domain. Each simulation run involved numerically solving the time-independent Schrödinger equation, where the Hamiltonian of the respective gate was

updated at each time step with the instantaneous phase of the laser sampled from the distribution as described earlier. The average gate fidelity was obtained by averaging the results of multiple simulation runs.

5.2.1 Single-qubit gate

For single-qubit rotations, I used the Hamiltonian 2.10. The phase $\phi = \phi_0 + \phi(t)$ is the sum of the desired laser phase ϕ_0 and the random variable $\phi(t)$, where $\phi_0 = 0$ w.l.o.g. for all subsequent simulations, as it does not change Rabi flops starting in $|g\rangle$. The Rabi frequency was set to $\frac{\Omega_0}{2\pi} = 189\text{kHz}$ to match the Rabi frequency achieved at the eQual setup. The simulation length was set to 20 Rabi oscillations or $T = 20\frac{2\pi}{\Omega_0}$. The noise was sampled from a white distribution, then low-pass filtered with a bandwidth of 7.8MHz (single-sided). Single instances of simulations experience contrast loss due to the noise-induced deflection \mathbf{j}_f from the ideal final state \mathbf{J}_f^o (see section 2.2.2). This contrast can be "revived" if \mathbf{j}_f becomes zero due to random processes, but this is not the case for the average over several instances. Thus, on average, the Rabi flops are a damped sinusoid with exponential decay, following the function

$$P_e(t) = \frac{1}{2} \left(1 + e^{-\frac{t}{\tau}} \cos(\Omega t + \pi) \right), \quad (5.1)$$

where Ω is the corrected Rabi frequency (see equation 5.5) and τ the exponential decay constant. Equation 2.47 from single-qubit theory suggests that the fidelity loss should be linear in gate time, not exponential. However, this cannot be true for long gate times, as the fidelity would eventually become negative. The derivation relied on the assumption that the deflection is small, allowing the use of the small-angle approximation. However, for long gates with relatively high phase noise, this approximation no longer holds. Instead, we can treat long gates as a series of infinitesimal gates, where each infinitesimal gate reduces the relative contrast by an infinitesimal amount:

$$dC = -C \frac{1}{\tau} dt, \quad (5.2)$$

where C is the contrast and τ the decay constant as before. Solving this differential equation for C yields the observed exponential decay $C(t) = \exp(-t/\tau)$. The fidelity is then $F = \frac{1}{2}(1 + C)$, since a truly random measurement in the limit $t \rightarrow \infty$ has a fidelity of $\frac{1}{2}$. We can also intuitively reason why this should be the case. Although the instantaneous phase of the rotational axis is a Markovian process in the case of white noise, the state vector to which the rotation is applied remembers all previous deflections. Thus, the total deflection vector, and consequently the infidelity, is a sum of all infinitesimal, random contributions, exhibiting random walk. Random walk deviation increases with the square root of the number of steps [63], or, in our case, with time. One individual Rabi flop trace can then be written as:

$$P_g(t) = 1 - P_e(t) = \cos \left(\frac{\Omega t}{2} + \alpha \sqrt{t} \right)^2, \quad (5.3)$$

where α is a Gaussian distributed random variable with standard deviation σ_α and mean zero. I also switched from the $|e\rangle$ population to the inverse $|g\rangle$ population to eliminate the phase of π . Averaging over all values of α results in:

$$\begin{aligned}
 P_g(t, \tau) &= \frac{1}{\sigma_\alpha \sqrt{2\pi}} \int_{-\infty}^{\infty} \cos(\Omega t + \alpha \sqrt{t})^2 e^{-\frac{\alpha^2}{2\sigma_\alpha^2}} d\alpha \\
 &= \frac{1}{2} (1 + e^{-2\sigma_\alpha t} \cos(\Omega t)).
 \end{aligned} \tag{5.4}$$

Choosing $\sigma_\alpha = \frac{1}{\sqrt{2\tau}}$ leads to the same exponential form as in equation 5.1.

A depiction of the average of several simulated traces can be seen in figure 5.2. The standard deviation is shown as a blue band around the average, all well an exponential fit to the simulated data. The Rabi frequency is not the same as in the noiseless case, as with increasing noise power, the carrier power must decrease due to Parseval's theorem (see section 2.2.1). This results in a reduced Rabi frequency given by:

$$\Omega = \Omega_0 e^{-\frac{\sigma^2}{2}}, \tag{5.5}$$

where σ^2 is the variance of $\phi(t)$.

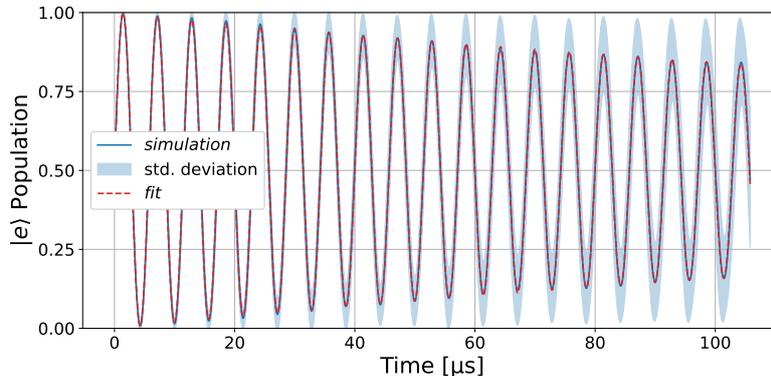


Figure 5.2: Averaging of many Rabi flops with induced phase noise at $\mathcal{L}_o = -80\text{dBc/Hz}$.

The fit parameters Ω and τ can be extracted from the averaged simulation. Running the simulation for several different noise levels results in a relation between phase noise density \mathcal{L} and the decay constant τ (see figure 5.3).

The average fidelity of a π rotation corresponds to the average $|e\rangle$ population after the gate time $t_\pi = \frac{\pi}{\Omega}$. Therefore, we can calculate the π gate fidelity as

$$F_\pi = \frac{1}{2} \left(1 + e^{-\frac{t_\pi}{\tau}} \cos(\Omega t_\pi + \pi) \right) = \frac{1}{2} \left(1 + e^{-\frac{t_\pi}{\tau}} \right). \tag{5.6}$$

By mapping the decay constants at each noise level to gate fidelities, we arrive at the relationship between phase noise density and gate infidelity depicted in figure 5.4. Since the fidelity of single-qubit rotations can be analytically calculated, we can compare the simulation results to the values obtained from equation 2.46, which depends on the initial polar angle θ_i and the azimuthal angle ϕ . Since the simulation always starts in the ground state $|g\rangle$, the initial polar angle is $\theta_i = -\frac{\pi}{2}$. The azimuthal angle ϕ is not unique at $|g\rangle$. However, since $|g\rangle$ is orthogonal to the rotational axis \mathbf{x} , all following states of the rotation are on the yz -plane and therefore have the azimuthal angle $\phi_i = \frac{\pi}{2}$ (or $\phi_i = \frac{3\pi}{2}$; however, this is the same for the even, periodic cosine function). Hence, we use $\phi_i = \frac{\pi}{2}$ as the initial angle as it is the limit for $t \rightarrow 0$. The calculated values fit very well to the simulated ones,

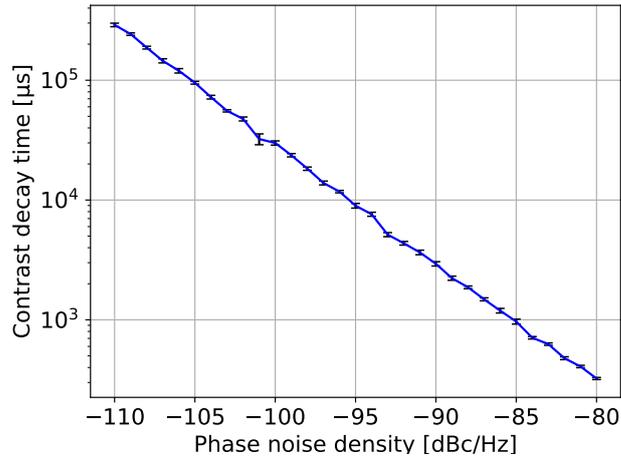


Figure 5.3: Contrast decay time versus various levels of phase noise density.

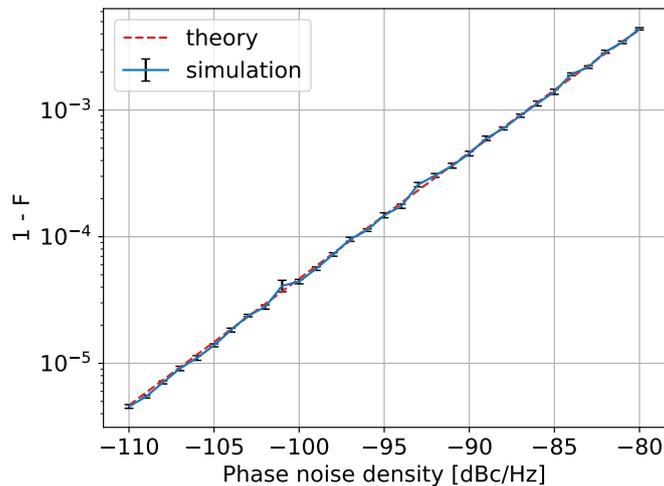


Figure 5.4: Gate fidelity of a π pulse versus various levels of phase noise density for a Rabi frequency $\frac{\Omega_0}{2\pi} = 189\text{kHz}$.

as can be seen in 5.4, confirming the simulation method. Note that the Rabi frequencies Ω are adjusted at each data point to the fitted frequencies from each simulation.

All previous simulations assumed low-pass-filtered white noise, and I found a dependence on the phase noise density. To confirm that this is the appropriate metric to compare, I conducted simulations to assess the dependence on the noise profile. For this purpose, I filtered the injected phase noise to a narrow band of 20kHz. The phase noise density within this band was set to $\mathcal{L}_\circ = -80\text{dBc/Hz}$ and the centre frequency of the band was swept. The resulting infidelity is plotted in figure 5.5. It can be observed that the infidelity is the highest when the centre frequency is at the Rabi frequency, where the 20kHz band of noise is responsible for almost all caused infidelity. This validates that the bandwidth of the injected white noise does not really matter, as long as the Rabi frequency is well within the noise bandwidth. Therefore, the total noise power does not influence the gate fidelity; only the noise power sampled by a small band around Ω matters. Therefore, I conclude that the phase noise density at the Rabi frequency is a good metric to describe gate fidelity. However, the narrow peak in figure 5.5 might be deceiving, as this fidelity was

extracted from Rabi flops, and therefore long gates. The theory of single-qubit rotations in section 2.2.2 however suggests that for shorter gates, the peak gets wider. Thus, for a π pulse, the sampled bandwidth is about as wide as Ω itself, even including DC for certain Bloch vectors (compare figure 2.9). For a $\pi/2$ pulse, the bandwidth is even higher at about 2Ω , always encompassing DC. I chose the simulation method however to keep it close to the experimental method used later. A detailed analysis of this problem can be found in section 2.2.2.

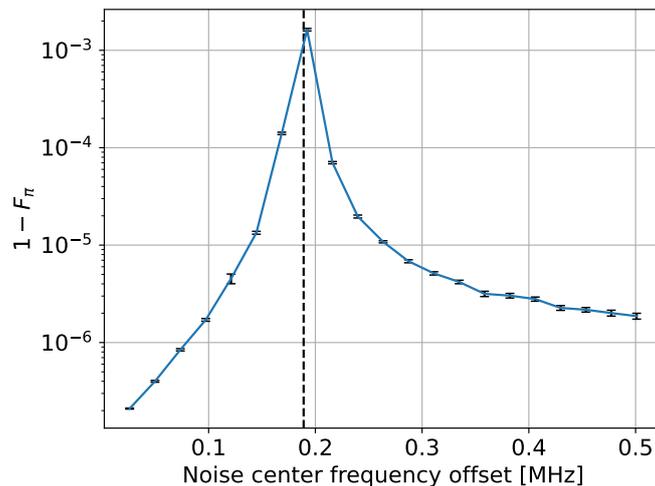


Figure 5.5: The centre frequency of a 20 kHz wide white noise band with $\mathcal{L}_{\circ} = -80\text{dBc/Hz}$ is swept and the resulting gate fidelity is measured.

5.2.2 MS gate

To simulate the MS gate for two ions, I used the result from equation 2.12. Similar to single-qubit gates, filtered Gaussian white noise adjusts the instantaneous phase of the Hamiltonian at each simulation time step. Since this Hamiltonian has two beams with a red and blue sideband, there are four phases, namely $\phi_{R,1}$, $\phi_{B,1}$, $\phi_{R,2}$ and $\phi_{B,2}$. Some of these random variables might be correlated depending on the implementation of the gate. For instance, the red and blue sideband may be generated by two digitally synthesised tones output on the same RF channel, or the same laser might be defocused to address both ions. However, it turns out that correlation between phases does not seem to matter. In the following, noise for each of the four phases is sampled uncorrelated to each other. W.l.o.g., the average of all phases is set to zero. I simulated by solving the time-independent Schrödinger equation with one sample of the noise distribution and examining the population of the target entangled state. In the ideal case, the target state is 1, and the difference to the actual state is the measured gate infidelity. The final gate fidelity is again the average over all simulation instances. An average over several instances of the population of the entangled state, the initial state $|g, g\rangle$ and $|e, e\rangle$ can be seen in figure 5.6a, with a zoom-in on the gate infidelity for the last microseconds in 5.6b. The injected phase noise is a white frequency band with $\mathcal{L}_{\circ} = -90\text{dBc/Hz}$ and a bandwidth of 8MHz. The gate parameters here and in the following are $\frac{\Omega}{2\pi} = 100\text{kHz}$, Lamb-Dicke parameter $\eta = 0.05$ and trap frequency $\omega_m = 3.75\text{MHz}$.

To retrieve a dependency on the phase noise density \mathcal{L} , I conducted a sweep of \mathcal{L} and plotted the resulting gate infidelity in figure 5.7. Similar to the single-qubit gates, the dependence is very linear. The linear fit (dashed line) indicates a factor of $a = 38.0 \pm 2.2 \frac{\text{MHz}}{\text{rad}^2}$.

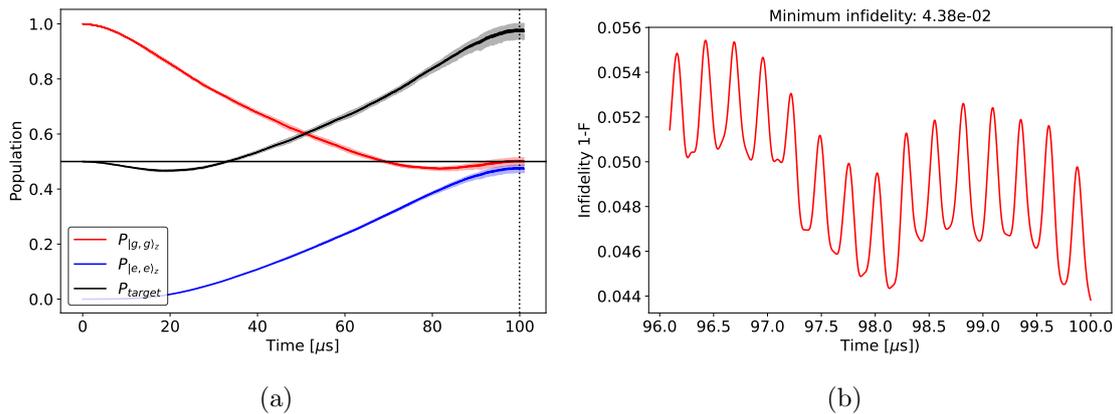


Figure 5.6: (a) The simulated populations of $|g, g\rangle$, $|e, e\rangle$ and the target state $P_{target} = |g, g\rangle + i|e, e\rangle$ averaged over several instances. (b) A zoom-in on the gate error ($1 - F = 1 - P_{target}$) for the last microseconds of the gate. The periodicity is caused by the driven off-resonant carrier term in equation 2.13.

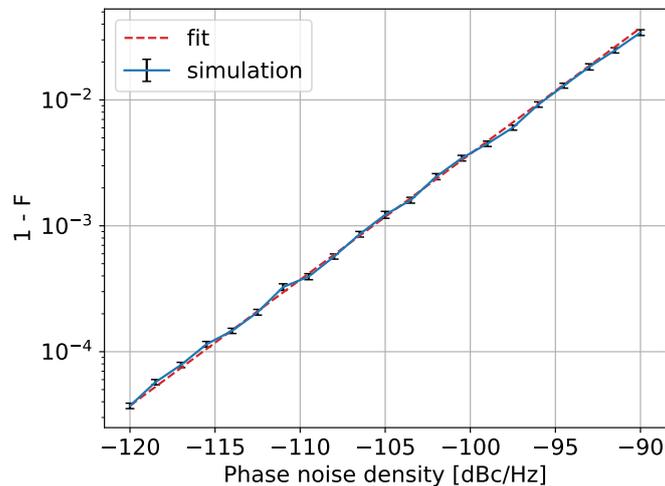


Figure 5.7: Gate fidelity of a MS gate versus various levels of phase noise density. Simulation parameters are $\frac{\Omega}{2\pi} = 100\text{kHz}$, $\eta = 0.05$ and $\frac{\omega_m}{2\pi} = 3.75\text{MHz}$.

The dependence on the profile of the noise is assessed by sweeping the centre frequency of a white noise band with a 1MHz bandwidth and a noise density of $\mathcal{L}_\circ = -90\text{dBc/Hz}$. The relevant frequency here is the trap frequency ω_m and, to a much lesser extent, the first multiple of ω_m . This can be explained by the carrier transition excited by the power spectral density at this offset, leading to incoherent spin-pumping [8]. Interestingly, the Rabi frequency does not play a significant role at all for multi-qubit gates. In figure 5.8, the first data point encompasses all frequencies from 0-1MHz and therefore also the Rabi frequency. It is even slightly negative due to numerical errors. For the MS gate, the peak in sensitivity is expected to be sharp because MS gates are comparatively long and therefore quite narrow in spectral width, sampling a narrow frequency band. Consequently, the phase noise density at the trap frequency $\mathcal{L}(\omega = \omega_m)$ proves to be an even better metric to describe noise dependency than $\mathcal{L}(\omega = \Omega)$ for single-qubit gates.

Note that these measured fidelities depend not only on the phase noise density but also on other gate parameters. To adjust the gate fidelity to a different Rabi frequency Ω or

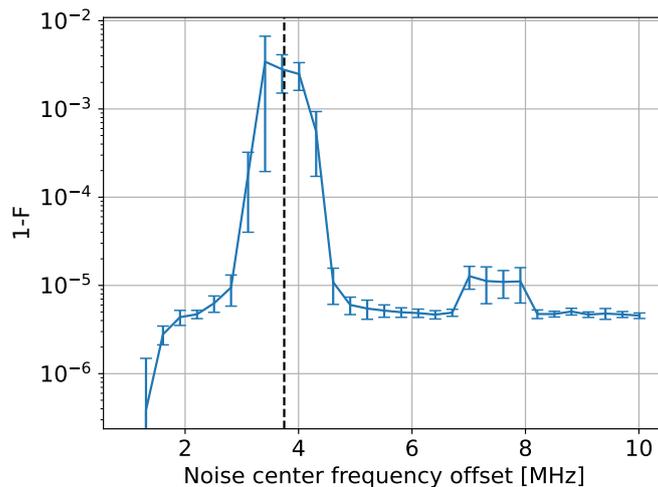


Figure 5.8: The centre frequency of a 1MHz wide white noise band with $\mathcal{L}_o=-90\text{dBc}/\text{Hz}$ is swept, and the resulting gate fidelity is measured.

Lamb-Dicke parameter η , the following relation can be used:

$$1 - F_{MS} \approx a \frac{\Omega}{\eta} \mathcal{L}(\omega = \omega_m), \quad (5.7)$$

where $a = 3.02 \pm 0.17 \frac{\text{Hz s}}{\text{rad}^2}$ is a constant extracted from the fit to the measurements before. An intuition for the relationships is given in the following: The gate infidelity is proportional to the spectral noise power of the electric field times gate time. The spectral noise power is carrier power times phase noise, thus $\propto \Omega^2 \mathcal{L}$, the gate time is $\propto \frac{1}{\delta} \propto \frac{1}{\eta \Omega}$. One Ω cancels and the total dependence is therefore as above.

5.3 Gate fidelity measurements

To validate the analytical and numerical analysis, we also measured the gate fidelity of a single-qubit π pulse. The experiments were conducted on the eQual setup (see section 3.4) on a $^{40}\text{Ca}^+$ ion, with much-appreciated support from Jeremy Flannery. In the experimental realisation, the implementation of the noise generator and the TI DAC card as the analogue front end were ultimately employed. The noise was injected into the RF signal of the single-pass AOM modulating the 729nm laser driving the Rabi flops. We did not get to the experimental verification of multi-qubit gates due to time constraints.

A typical experiment sequence is depicted in figure 5.9. At first, the ion is cooled close to the motional ground state with Doppler cooling followed by resolved sideband cooling. Subsequently, the ion is prepared in the electronic ground state through state preparation. After these basic operations, the main part of the experiment involves the implementation of the actual gates. Finally, the resulting state is read out. Note that only the main part consists of coherent pulses, while the cooling, state preparation and readout consist of incoherent pulses. This sequence is repeated several times to get a statistical average [45].

5.3.1 Single-qubit rotations

To measure the influence of phase noise density on the fidelity of single-qubit gates, we followed a methodology similar to the numerical approach. Our goal was to measure

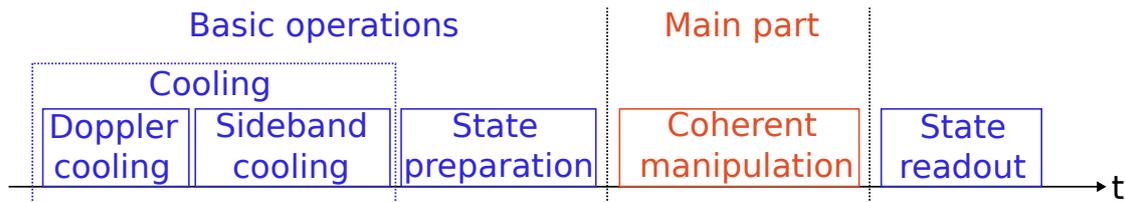


Figure 5.9: Illustration of a typical experimental sequence.

decaying Rabi flops, fit the decay constant and calculate the π pulse gate fidelity utilising equation 5.6. While single-qubit gates do not directly involve the motional state, it is still advantageous to have very cold ions. This is because, although the Rabi frequency is independent of the motional state n in the first order, considering the next-order correction, the Rabi frequency becomes [25]:

$$\Omega_n = \Omega \left(1 - \frac{\eta^2}{2}(2n + 1) \right) + O(\eta^4). \quad (5.8)$$

This would not be a problem if the ion was in a Fock state. However, after cooling, the ion is in a thermal state, which leads to a fluctuating Rabi frequency and therefore decoherence and contrast loss, even without injected noise. To mitigate this effect, the ions has to be cooled close to the motional ground state. Therefore, a lot of sideband cooling loops were necessary. Together with the requirement of enough points to fit a sine with sufficient accuracy and a quite long time window to capture the decay, a single measurement such as in figure 5.2 would take several hours. A long measurement is susceptible to interruptions, such as those caused by an unlocked laser. Also, parameters might drift over time, obscuring the results. Therefore, I selected several time windows, encompassing about two oscillations, at several time offsets. At each time window, I measured oscillations both with and without injected noise for comparison. The measurement order of the time windows was selected at random to avoid measuring the impact of parameter drift instead of just pulse length. An example of a measurement with six different time windows is depicted in figure 5.10a. The plot on top is the $|e\rangle$ state population P_e over time without injected noise, while the bottom plot depicts P_e with injected noise. The phase noise density for this particular plot was $\mathcal{L}_o = -84.8 \text{ dBc/Hz}$. The noise profile in this plot, as well as all subsequent measurements, was white phase noise with a bandwidth of 7.85MHz.

The bandwidth in this measurement, and subsequently in the simulations to mirror the experiment, was a deliberate choice. Initially, the noise profile was white noise up to the Nyquist frequency of the generated samples (125MHz). However, relatively high background noise without noise injection obscured the effects of the injected phase noise. Simply increasing the phase noise to observe an effect was not straightforward because doing so caused the Rabi frequency to decrease (see equation 5.5). Consequently, comparing the reference measurement without injected noise to the noisy measurement was more difficult due to differing Rabi frequencies. The solution was to implement an additional low-pass filter with the aforementioned corner frequency of 7.85MHz. This reduces the power in the noise by a factor of 16, while still not affecting the measurements, because both the Rabi frequency and the trap frequency for eventual MS gate experiments are well within the passband. Furthermore, the AOM has a corner frequency in the same order, such that noise above the AOM corner frequency would be filtered out anyway and would not matter. This bandwidth limitation therefore allowed us to measure points at higher phase noise densities without a significant reduction in Rabi frequency.

For every time window in 5.10a, a sine was fitted to the data points within the window for both the reference measurement and the measurement with injected noise. For the latter, a zoom-in on the six measurement windows with fits is depicted in figure 5.10b. In this approach, the length of each time window compared to the total measurement length is assumed to be small enough to neglect the effect of decay within a single time window. By fitting a sine curve to each time window, I obtained contrast and frequency values for each specific offset time of the window. Another approach would have been to fit a damped sine directly to the composite plot like figure 5.10a. However, providing good starting parameters for such a fit proved challenging, as even slight variances in frequency would result in very different fits due to the large blocks of missing points to fit to. Therefore, I chose the method of individual fits for each time slot.

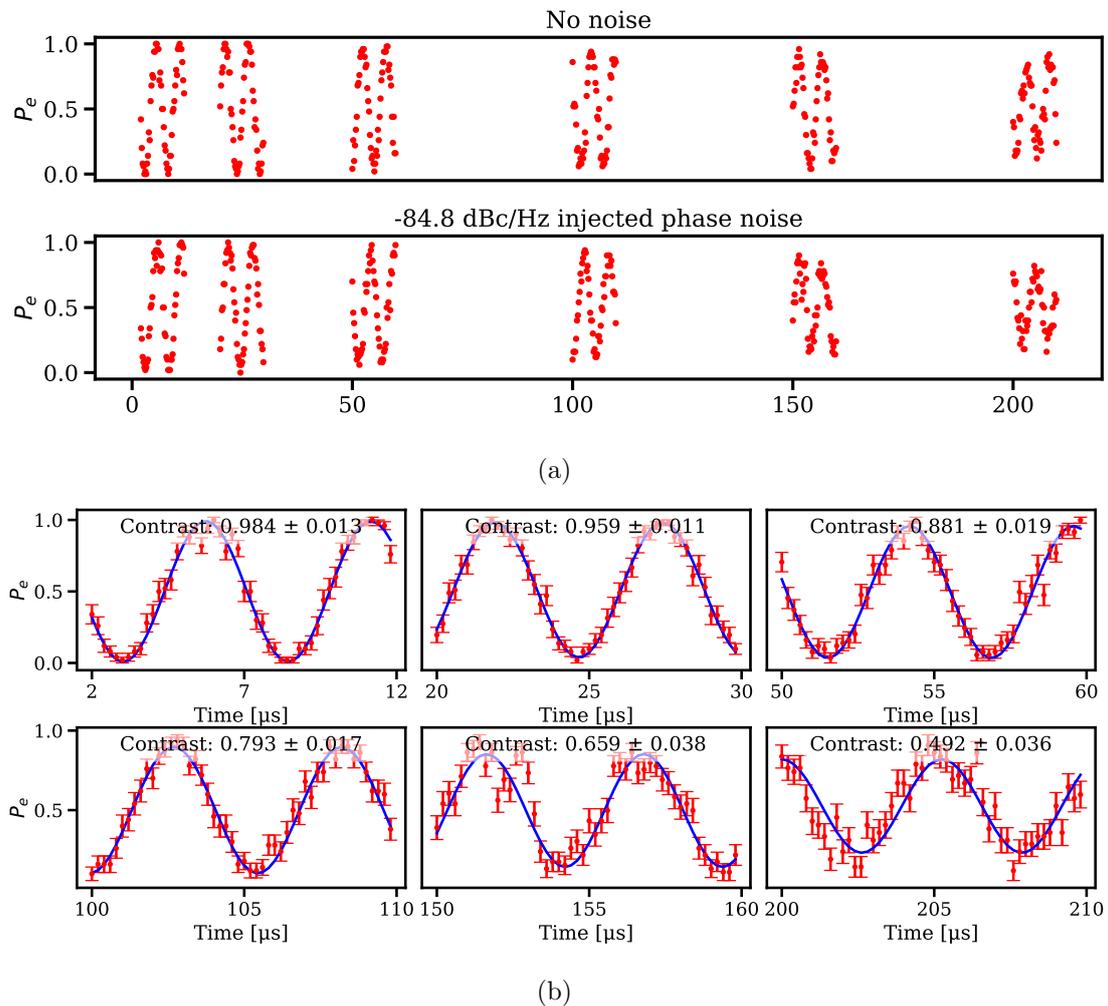


Figure 5.10: (a) Measured decaying Rabi flops. To keep the number of measured points low, only certain windows in time were measured. The top plot shows a measurement without injected noise, and the bottom plot one with injected phase noise. (b) A zoom-in on these time windows for the measurement with injected noise. For each time window, a sine was fitted, whose contrast fit serves as a contrast data point at this time offset.

The contrasts fitted to the measurement windows can then be plotted against the measurement time to observe the dependency of contrast on time. An example for $\mathcal{L}_0 = -84.8 \text{ dBc/Hz}$ can be seen in figure 5.11. The red line represents the contrast decay without additional injected noise (background noise), while the blue line represents the decay with injected

phase noise. The background noise fits well to a Gaussian decay (orange line), primarily attributed to slow Rabi frequency noise caused by laser frequency and amplitude fluctuations, as well as slow magnetic field noise [22, p. 91]. As expected, the measurement with phase noise shows a faster contrast loss than the reference, attributed to the additional exponential decay. However, in contrast to the simulations, the observed decay is now a mixture of the two noise processes. Hence, we have to separate those two to get the influence of the phase noise. The contrast decay can be modelled as a mixture of Gaussian and exponential decay:

$$C(t) = C_0 e^{-\frac{t^2}{2\tau_{gauss}^2}} e^{-\frac{t}{\tau_{exp}}}, \quad (5.9)$$

where C_0 is the initial contrast, τ_{gauss} is the decay constant for the Gaussian decay part and τ_{exp} is the decay constant for the exponential decay part. In a first attempt, I assumed the Gaussian decay to be the same for the reference and the noisy measurement. Therefore, I modelled the noisy measurement with the Gaussian decay τ_{gauss} fixed to the value retrieved from the fit of the reference. The initial contrast C_0 and τ_{exp} are free parameters. The fit of this model is represented by the black curve. However, it becomes apparent later that for low noise, assuming the same Gaussian decay leads to a fitted exponential decay constant close to zero with high confidence. This discrepancy arises because if the exponential decay is very weak, even slight variations in Gaussian decay between the reference and noisy measurement significantly influence the fitted exponential decay constant but are not accounted for. To address this issue, I also fitted a mixed fit where both the Gaussian and exponential decay constants, as well as the initial contrast, are free variables (green line). The fitted Gaussian decay constants of both the reference and noisy measurement are expected to be similar, which they are. The Gaussian decay constant of this three-parameter fit is then attributed to the slow frequency noise, while the exponential decay is assumed to be the effect of the phase noise.

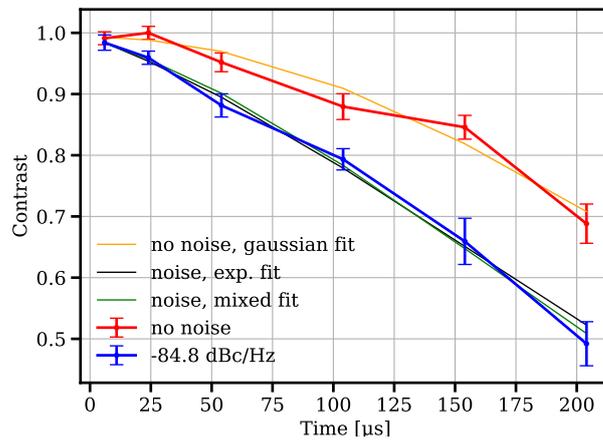


Figure 5.11: The decay of contrast for both the measurements without injected noise (red line) and with injected white phase noise at $\mathcal{L}_o = -84.8 \text{ dBc/Hz}$ (blue line). The orange line fits a Gaussian decay to the background noise. The black line is a fit of equation 5.9, where τ_{gauss} is fixed to the value retrieved from this previous fit to the background noise. The green line fits a mixed Gaussian and exponential decay to the noisy trace with both decay constants as free variables.

The previous measurement is then repeated for several phase noise densities. Each of these measurements yields a fit for the exponential decay constant, which can be converted to a gate fidelity using equation 5.6. The dependence of the gate infidelity of a π pulse

on the noise density is illustrated in figure 5.12. Additionally, the theoretical fidelity calculated with equation 2.46 is represented by the blue line, while the simulated gate fidelity is depicted by the orange dashed line. For high injected phase noise densities ($\mathcal{L}_o > -85\text{dBc/Hz}$), the measured points align well with the analytical and numerical results. Below that noise level, the Gaussian background noise dominates the measurements, leading to poor agreement. Here, the difference between the two fitting approaches becomes apparent. While both exponential and mixed fit agree well for high noise densities, it can be seen that the exponential fit has very small error bars, while the mixed fit error bars seem reasonable. For low noise densities, the exponential fit points are near zero (and are therefore omitted from the plot), but with very high confidence. The mixed fit points also disagree with the theory and simulations for low noise densities, but the error bars at least indicate their inaccuracy.

Note that there is a non-linearity, which becomes prominent above approximately $\mathcal{L}_o = -80\text{dBc/Hz}$. This is attributed to two factors. Firstly, as explained in section 5.2.1, the Rabi frequency decreases according to equation 5.5 as the injected phase noise increases. Since the gate infidelity is also linear in the Rabi frequency, this reduces the gate infidelity plotted here. Secondly, we are in a regime where it cannot be longer assumed that the noise power is small compared to the carrier. As a result, the noise power is not only in the first phase modulated sideband in the carrier domain anymore, but a non-negligible amount of noise power is also in higher-order sidebands (see section 2.2.1). Consequently, the noise power in the first sideband decreases, further reducing the measured gate infidelity. Note that the noise density on the x-axis represents the power density spectrum $S_E(f)$ in the carrier-frequency domain. This reflects the noise spectral density of the electric field at some frequency offset from the carrier relative to the actual carrier power (which is reduced by the aforementioned effects). This is in contrast to the usually reported phase noise density according to definition 2.25. I made this choice because $S_E(f)$ is the noise density that directly influences the gate fidelity. This is evident in the fact that the theoretical values fit very well to the measured points using $S_E(f)$ instead of $\mathcal{L}(f)$. However, for small, and therefore realistic, phase noise levels the difference is negligible and $S_E(f) = \mathcal{L}(f)$.

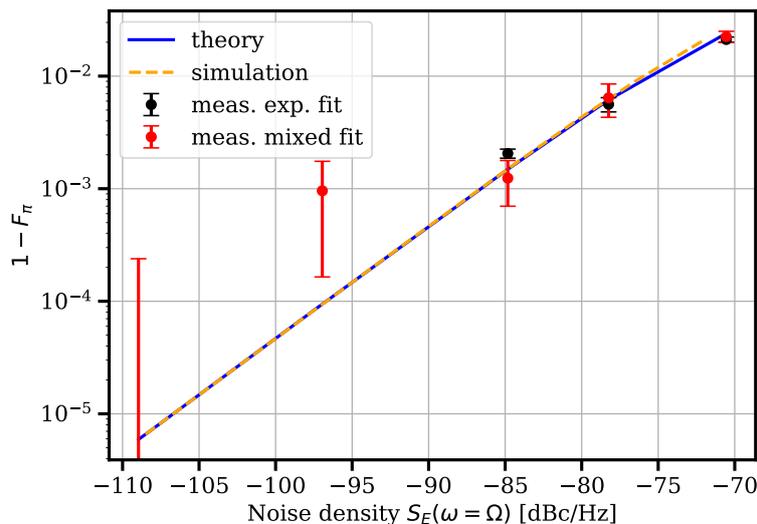


Figure 5.12: Measured gate fidelity of a π pulse versus various levels of noise spectral density for a Rabi frequency $\frac{\Omega_0}{2\pi} = 189\text{kHz}$. Note that for low noise densities (in this plot $< 80\text{dBc/Hz}$), $S_E(f)$ is virtually equivalent to $\mathcal{L}(f)$.

Chapter 6

Discussion

In the course of this thesis, I investigated the influence of phase noise from RF generation hardware on the gate fidelity of trapped-ion qubits. I analysed single- and multi-qubit gates numerically, while single-qubit gates were additionally investigated analytically. Furthermore, I implemented a noise generator on the control hardware of a quantum experiment setup to inject phase noise during experiments. This enabled measuring the effect of the noise on gate fidelity on real trapped-ion qubits. Using this hardware setup, I also confirmed the dependence of infidelity on the noise level for white noise.

For long single-qubit gates, the average gate fidelity is proportional to the gate angle ψ , the angular Rabi frequency Ω and the phase noise density $\mathcal{L}(\omega = \Omega)$ at the Rabi frequency:

$$1 - F_\psi = \frac{1}{6} \psi \Omega \mathcal{L}(\omega = \Omega). \quad (6.1)$$

Long gates are those with $\psi > \pi$. For shorter gates, it is not as straightforward as only considering the noise at the Rabi frequency, as the sampled noise bandwidth increases with shorter gates. For $\psi < \pi$, slow noise near the carrier also starts to play a role. Thus, to establish the exact relationship, an integral of the product of noise and transfer function is necessary. Also, the exact infidelity depends on the final state of the qubit after the gate.

With a Rabi frequency of $\frac{\Omega}{2\pi} = 189\text{kHz}$ used in simulation and measurement, this corresponds to $\mathcal{L}(\omega = \Omega) < -97\text{dBc/Hz}$ to achieve a gate fidelity of $1 - F_\pi = 10^{-4}$. For short gates, slow noise starts to have an influence, primarily limited by $\frac{1}{f^2}$ noise rather than $\frac{1}{f}$ noise. To ensure that the effects of slow noise remain below the impact of noise at the Rabi frequency in the worst-case scenario, the noise coefficients according to equation 2.26 should satisfy $\mathcal{L}_1 < -60\frac{\text{dBcHz}}{\text{Hz}}$ (equivalent to a corner frequency with respect to $\mathcal{L}_o = -100\text{dBc/Hz}$ of $f_{c1} < 1\text{kHz}$) and $\mathcal{L}_2 < -51\frac{\text{dBcHz}^2}{\text{Hz}}$ ($f_{c2} < 300\text{Hz}$ with $\mathcal{L}_o = -100\text{dBc/Hz}$). To account for this effect of slow noise, white noise for fast gates should be $\mathcal{L}_o = < 100\text{dBc/Hz}$. Note that these requirements have to be scaled with the Rabi frequency.

For multi-qubit gates, I found a linear dependence on the phase noise density at the trap frequency $\mathcal{L}(\omega = \omega_m)$ and the Rabi frequency Ω . It is also inversely proportional to the Lamb-Dicke parameter η :

$$1 - F_{MS} \approx a \frac{\Omega}{\eta} \mathcal{L}(\omega = \omega_m), \quad (6.2)$$

where $a = 3.02 \pm 0.17\frac{\text{Hz}\cdot\text{s}}{\text{rad}^2}$ is a fitted constant. In contrast to single-qubit gates, the noise at the trap frequency, which is typically around 1MHz, is always a good measure for noise

requirements. This is because multi-qubit gates are much longer than single-qubit gates, resulting in a narrower spectral width. With parameters of $\frac{\Omega}{2\pi} = 100\text{kHz}$ and $\eta = 0.05$, achieving a gate fidelity of $1 - F = 10^{-4}$ requires the noise at the trap frequency ω_m to be $\mathcal{L}(\omega = \omega_m) < -116\text{dBc/Hz}$. Again, an increase in Rabi frequency requires the limit to be more stringent accordingly.

Thus, the limiting factor likely lies in the requirements for multi-qubit gates. As expected, lowering the overall noise density will lead to a higher gate fidelity. However, what is more intriguing is the sensitivity of gates to specific frequencies in the noise spectrum. This could possibly enable engineering of the noise such that the noise at these sensitive frequencies is minimised. Noise shaping in the case of quantisation noise is such an example [64]. In practice, this might be difficult, since the system has to be designed to accommodate different Rabi and trap frequencies. Thus, increasing noise between those two frequencies is not really viable. Also, near-DC noise starts to matter for fast single-qubit gates, thus also slow noise cannot be ignored. Nevertheless, above the trap frequency, noise does not matter in any case.

Currently, qubit gates are far from being limited by phase noise from RF pulses. Other noise sources dominate the gate error, as is evident in the measurements conducted in this thesis. However, as efforts continue to reduce other sources of noise, phase noise from RF generation may become a relevant factor in the future. Thus, it is sensible to design systems with this consideration in mind.

6.1 Comparison to other papers

As seen in section 5, my numerical analysis and measurements fit the theoretical analysis derived in Zilong Chen et al. [9] well. This is especially true for the phase noise magnitude, as the paper's results for gate fidelity focus mostly on white noise. The impact of the profile of phase noise can also be derived from the theory and also fits my results qualitatively.

The paper of Haim Nakav and Ran Finkelstein from the Weizmann Institute [8] conducted similar simulations and measurements to mine. The authors also study the effect of phase noise on the fidelity of trapped-ion gates. However, their main focus is fast noise of the laser. It predominantly comes from the servo bump, an artefact of the feedback loop stabilising the laser frequency. I focus on the phase noise contribution from the RF generation, which has a different noise profile, as noise density increases near the carrier (see 5.1a). The main conclusions are comparable, however, especially for the MS gate, where the relevant frequency range is narrow-band around the trap frequency.

I simulated the MS gate with the parameters given in their discussion, namely with a Rabi frequency of $\frac{\Omega}{2\pi} = 100\text{kHz}$ and a Lamb-Dicke parameter of $\eta = 0.05$, which translates to a gate time of $100\mu\text{s}$. One of the differences between my and their simulation is the phase noise profile, as I used white noise, bandwidth-limited to 7.85MHz . Note that they use the concept of Rabi power spectral density (RPSD). It is the power spectral density of the carrier, normalised "such that the area under the carrier peak is Ω^2 ". It is not clear to me if this integral also includes the power of the noise away from the carrier, but it should not matter, since in this discussion, the power of the noise is negligible compared to the power of the carrier itself. It is also not obvious if the Rabi frequency in this normalisation is given in units of frequency or angular frequency. I assume that it is given in angular frequency, as numerical values for Rabi frequencies throughout the paper are written with a factor of 2π (although the unit is given as Hz instead of $\frac{1}{\text{s}}$). Thus, assuming all power is in the carrier and a Rabi frequency Ω given in angular frequency, the RPSD can be expressed as $RPSD(f) = \Omega^2 PSD(f)$, where $PSD(f)$ is given as the noise power density

relative to the carrier power. The gate infidelity resulting from my simulation is plotted against the RPSD in figure 6.1. The expected gate infidelity from the paper is plotted in green. It is a linear function with a proportionality constant of $1 \frac{sHz}{rad^2}$, as it is written by the authors. The phase noise density $\mathcal{L}(\omega = \omega_m)$, as I have plotted throughout this thesis, is given as a second axis label at the top.

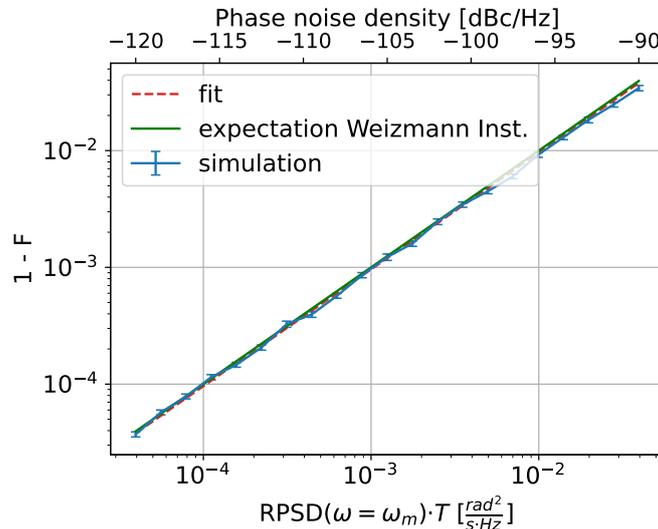


Figure 6.1: Comparison of my simulations to a linear plot with a proportionality constant of $1 \frac{sHz}{rad^2}$.

The two results seem in very good agreement. However, my conclusion differs from theirs given in their discussion. As it can be read from the plot, to reach gate infidelities of $1 - F < 10^{-4}$, the phase noise density has to be limited to $\mathcal{L} < -116\text{dBc/Hz}$. In comparison, they claim a requirement of -100dBc/Hz . Interestingly, this is close to a difference of $(2\pi)^2 = 10 \log((2\pi)^2)\text{dB} = 15.96\text{dB}$. One explanation would be that they scaled the RPSD with a Rabi frequency given in frequency, not angular frequency. This thought is also supported by the fact that they mention the limit in terms of RPSD to be $RPSD_{max} < \sim 1 \frac{Hz^2}{Hz}$ instead of $1 \frac{rad^2}{s^2 Hz}$. This would lead to the phase noise density $\mathcal{L} < 10 \log \frac{RPSD_{max}}{f_r^2} \text{dBc} = -100\text{dBc/Hz}$. Here, the Rabi frequency is written as $f_r = 100\text{kHz}$ to avoid confusion with angular Ω . In this case, there is an inconsistency with the RPSD scaling in figure 6.1.

Also, the proportionality constant for the single-qubit gates from my thesis is about 2π times greater than figure 2c of their paper suggests (assuming angular frequency RPSD scaling). I am quite confident however that the proportionality I derived is correct, as it fits both theory and my measurements.

6.2 Possible further investigation

There are multiple aspects of the influence of phase noise that were not explored fully due to time constraints. Firstly, the current method of simulation for single-qubit gates is limited in the regard that it only simulates Rabi flops. Hence, it does not generalise for short gates. Additionally, it always starts in the ground state, while real gates would be applied to different states distributed all over the Bloch sphere. The results of simulation and measurement fit well with the theory for this special case, which justifies the theory. However, this does not reflect the average gate fidelity, although they are similar (compare

equations 2.46 and 2.47). A possible method of simulation and measurement would be random benchmarking [65]. It applies a series of random gates. The final output state is then compared to the ideal one to get a fidelity for the whole gate sequence, which can be used to estimate the average infidelity per gate. Further investigation into short gates would also be interesting, since here flicker noise and especially white frequency noise start to contribute to the gate fidelity. This thesis disregarded slow noise due to the theory suggesting it may not be as relevant. Further analysis and measurement however should confirm that the assumptions for the theoretical analysis are indeed correct. Since slow noise is often the hardest to decrease in electronics, a second look at this topic might be worth it.

Multi-qubit gates were simulated with the common Lamb-Dicke approximation. However, for large Lamb-Dicke parameters, this approximation breaks down and further frequencies at multiples of the trap frequency might become relevant. There are advantages of an increased Lamb-Dicke parameter, such as decreased gate time. Thus, a simulation with more terms of the Lamb-Dicke expansion might be interesting for such experiments. I found that MS gates are sensitive to noise at the trap frequency with a certain bandwidth, but I did not investigate which parameters influence this bandwidth. Also, the dependence of MS gates on the initial state was not explored.

Finally, I did not utilise the full potential of the noise generator implemented on the RF generation hardware to measure physical quantum gates with noise injection, although I tested and integrated it fully into the control system. While I did confirm the dependency on noise density for single-qubit gates, I only did so for white noise. The functionality of $\frac{1}{f^2}$ noise generation was tested, but ultimately not used in the measurements. Thus, more measurements could highlight the effect of noise profile and acceptable white frequency noise levels. Multi-qubit measurements were not conducted at all. It would have taken changes to the experimental setup, which were not possible in time due to scheduling with other experiments. Thus, experimental validation of the effect of phase noise on multi-qubit gates is lacking.

Additionally, some improvements to the noise generation could be made. Flicker noise was not implemented due to the requirement of a more complicated generation algorithm (and because theory does not expect it to have a significant impact). It may also be beneficial to be able to generate other noise profiles like a servo bump. Although that does not reflect phase noise of the RF generation, it does reflect laser noise. This way, the project could be expanded to examine the impact of other noise sources, increasing usefulness to other scientists.

Bibliography

- [1] Paul Benioff. “The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines”. In: *Journal of Statistical Physics* 22.5 (May 1980), pp. 563–591. ISSN: 1572-9613. DOI: 10.1007/BF01011339. URL: <https://doi.org/10.1007/BF01011339>.
- [2] Yu. I. Manin. Vychislimoe i nevychislimoe [Computable and Noncomputable]. Russian. Sov.Radio, 1980, pp. 13–15.
- [3] Richard P. Feynman. “Simulating physics with computers”. In: *International Journal of Theoretical Physics* 21.6 (June 1982), pp. 467–488. ISSN: 1572-9575. DOI: 10.1007/BF02650179. URL: <https://doi.org/10.1007/BF02650179>.
- [4] David Deutsch. “Quantum theory, the Church–Turing principle and the universal quantum computer”. In: *Proc. R. Soc. Lond. A* 400 (1985), pp. 97–117. DOI: 10.1098/rspa.1985.0070. URL: <http://doi.org/10.1098/rspa.1985.0070>.
- [5] Lov K. Grover. “A fast quantum mechanical algorithm for database search”. In: (1996). DOI: 10.48550/ARXIV.QUANT-PH/9605043.
- [6] Peter W. Shor. “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer”. In: *SIAM Journal on Computing* 26.5 (Oct. 1997), pp. 1484–1509. DOI: 10.1137/s0097539795293172.
- [7] Michael Brooks. “Beyond quantum supremacy: The hunt for useful quantum computers”. In: *Nature* 574.7776 (Oct. 2019), pp. 19–21.
- [8] Haim Nakav et al. “Effect of fast noise on the fidelity of trapped-ion quantum gates”. In: *Physical Review A* 107.4 (Apr. 2023). DOI: 10.1103/physreva.107.042622. URL: <https://doi.org/10.1103/physreva.107.042622>.
- [9] Zilong Chen et al. “General formalism for evaluating the impact of phase noise on Bloch vector rotations”. In: *Phys. Rev. A* 86 (3 Sept. 2012), p. 032313. DOI: 10.1103/PhysRevA.86.032313. URL: <https://link.aps.org/doi/10.1103/PhysRevA.86.032313>.
- [10] Matthew L. Day et al. “Limits on atomic qubit control from laser noise”. In: *npj Quantum Information* 8.1 (27th June 2022), p. 72. ISSN: 2056-6387. DOI: 10.1038/s41534-022-00586-4. URL: <https://doi.org/10.1038/s41534-022-00586-4>.
- [11] The Julia Programming Language. <https://julialang.org/>. Visited on 10/11/2023.
- [12] Sebastian Krämer et al. “QuantumOptics.jl: A Julia framework for simulating open quantum systems”. In: *Computer Physics Communications* 227 (2018), pp. 109–116.
- [13] David P. DiVincenzo. “The Physical Implementation of Quantum Computation”. In: *Fortschritte der Physik* 48.9-11 (2000), pp. 771–783.
- [14] Wen-Han Png et al. “Quantum Computing With Trapped Ions: An overview”. In: *IEEE Nanotechnology Magazine* 16.4 (2022), pp. 30–36. DOI: 10.1109/MNANO.2022.3175384.
- [15] R. Blatt et al. “Ion Trap Quantum Computing with Ca⁺ Ions”. In: *Quantum Information Processing* 3.1 (Oct. 2004), pp. 61–73. DOI: 10.1007/s11128-004-3105-1. URL: <https://doi.org/10.1007/s11128-004-3105-1>.

- [16] Hong-Fang Song et al. “Relationship measurement between ac-Stark shift of 40 Ca + clock transition and laser polarization direction *”. In: *Chinese Physics B* 26 (Aug. 2017), p. 099501. DOI: 10.1088/1674-1056/26/9/099501.
- [17] Wolfgang Paul and Helmut Steinwedel. “Notizen: Ein neues Massenspektrometer ohne Magnetfeld”. In: *Zeitschrift für Naturforschung A* 8 (1953), pp. 448–450.
- [18] Masatoshi Kajita. Ion Traps. 2053-2563. IOP Publishing, 2022. ISBN: 978-0-7503-5472-1. DOI: 10.1088/978-0-7503-5472-1. URL: <https://dx.doi.org/10.1088/978-0-7503-5472-1>.
- [19] Fouad G. Major, Viorica N. Gheorghe and Günther Werth. Charged Particle Traps. Physics and Techniques of Charged Particle Field Confinement. Springer Series on Atomic, Optical, and Plasma Physics. Springer Berlin, Heidelberg, 2005, pp. XII, 356. DOI: 10.1007/b137836.
- [20] David J. Wineland et al. “Experimental Issues in Coherent Quantum-State Manipulation of Trapped Atomic Ions”. In: *J Res Natl Inst Stand Technol* 103.3 (May 1998). Epub 1998 Jun 1, pp. 259–328. ISSN: 1044-677X. DOI: 10.6028/jres.103.019.
- [21] J. I. Cirac and P. Zoller. “Quantum Computations with Cold Trapped Ions”. In: *Phys. Rev. Lett.* 74 (20 May 1995), pp. 4091–4094. DOI: 10.1103/PhysRevLett.74.4091. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.74.4091>.
- [22] Matteo Marinelli. “Quantum information processing with mixed-species ion crystals”. Doctoral dissertation. ETH Zurich, 22nd May 2020.
- [23] Anders Sørensen and Klaus Mølmer. “Entanglement and quantum computation with ions in thermal motion”. In: *Phys. Rev. A* 62 (2 July 2000), p. 022311. DOI: 10.1103/PhysRevA.62.022311. URL: <https://link.aps.org/doi/10.1103/PhysRevA.62.022311>.
- [24] Vlad Negnevitsky. “Feedback-stabilised quantum states in a mixed-species ion system”. Doctoral dissertation. Zurich, Switzerland: ETH Zürich, 2018.
- [25] Maciej Malinowski. “Unitary and Dissipative Trapped-Ion Entanglement Using Integrated Optics”. Doctoral dissertation. Zurich, Switzerland: ETH Zurich, 2021.
- [26] Frequency Standards. John Wiley & Sons, Ltd, 2003. ISBN: 9783527605996. DOI: <https://doi.org/10.1002/3527605991>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/3527605991>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/3527605991>.
- [27] Enrico Rubiola. “Phase noise and frequency stability”. In: *Phase Noise and Frequency Stability in Oscillators*. The Cambridge RF and Microwave Engineering Series. Cambridge University Press, 2008, pp. 1–34. DOI: 10.1017/CB09780511812798.005.
- [28] K.J. Button. Infrared and Millimeter Waves: Instrumentation. Infrared and millimeter waves. Elsevier Science, 2014. ISBN: 9780323153164. URL: <https://books.google.ch/books?id=PiWVSmRKVFAC>.
- [29] Erik A. McShane and Krishna Shenai. “2 - Noise in Analog and Digital Systems”. In: *The Electrical Engineering Handbook*. Ed. by WAI-KAI CHEN. Burlington: Academic Press, 2005, pp. 101–108. ISBN: 978-0-12-170960-0. DOI: <https://doi.org/10.1016/B978-012170960-0/50011-6>. URL: <https://www.sciencedirect.com/science/article/pii/B9780121709600500116>.
- [30] J. B. Johnson. “Thermal Agitation of Electricity in Conductors”. In: *Phys. Rev.* 32 (1 July 1928), pp. 97–109. DOI: 10.1103/PhysRev.32.97. URL: <https://link.aps.org/doi/10.1103/PhysRev.32.97>.
- [31] T.H. Lee and A. Hajimiri. “Oscillator phase noise: a tutorial”. In: *IEEE Journal of Solid-State Circuits* 35.3 (2000), pp. 326–336. DOI: 10.1109/4.826814.

- [32] W. P. Robins. Phase Noise in Signal Sources. Telecommunications. Institution of Engineering and Technology, 1984. URL: <https://digital-library.theiet.org/content/books/te/pbte009e>.
- [33] Enrico Rubiola and Remi Brendel. “The AM noise mechanism in oscillators”. In: *2009 IEEE International Frequency Control Symposium Joint with the 22nd European Frequency and Time forum*. 2009, pp. 33–39. DOI: 10.1109/FREQ.2009.5168137.
- [34] Pierre Brochard, Thomas Südmeyer and Stéphane Schilt. “Power Spectrum Computation for an Arbitrary Phase Noise Using Middleton’s Convolution Series: Implementation Guideline and Experimental Illustration”. In: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* 64.11 (2017), pp. 1766–1775. DOI: 10.1109/TUFFC.2017.2747620.
- [35] “IEEE Standard Definitions of Physical Quantities for Fundamental Frequency and Time Metrology-Random Instabilities”. In: *IEEE Std 1139-1999* (1999), pp. 1–40. DOI: 10.1109/IEEESTD.1999.90575.
- [36] Richard Jozsa. “Fidelity for Mixed Quantum States”. In: *Journal of Modern Optics* 41.12 (1994), pp. 2315–2323. DOI: 10.1080/09500349414552171. eprint: <https://doi.org/10.1080/09500349414552171>. URL: <https://doi.org/10.1080/09500349414552171>.
- [37] Jürgen Reichardt. Eine Einführung mit VHDL. Berlin, Boston: De Gruyter Oldenbourg, 2017. ISBN: 9783110478341. DOI: doi:10.1515/9783110478341. URL: <https://doi.org/10.1515/9783110478341>.
- [38] Cadence Design Systems. Hardware Description Languages: VHDL vs. Verilog and Their Functional Uses. 2020. URL: <https://resources.pcb.cadence.com/blog/2020-hardware-description-languages-vhdl-vs-verilog-and-their-functional-uses> (visited on 03/02/2024).
- [39] Stephen Bailey. Digital Simulation White Paper: Comparison of VHDL, Verilog and SystemVerilog. Model Technology, 2003. URL: https://doc.inmys.ru/open?path=_books%5Chdl%5CComparison%20of%20VHDL%20Verilog%20and%20SystemVerilog-201471223958883.pdf.
- [40] Stephen Williams. Icarus Verilog Extensions. 2024. URL: https://steveicarus.github.io/iverilog/usage/icarus_verilog_extensions.html (visited on 03/02/2024).
- [41] Jonathan Harris, Ian Beavers, Del Jones, et al. JESD204B Survival Guide. 2013.
- [42] JEDEC Solid State Technology Association. Serial Interface for Data Converters: JESD204B (Revision of JESD204A, April 2008). Tech. rep. JEDEC Solid State Technology Association, July 2011.
- [43] Texas Instruments. TI-JESD204-IP. URL: <https://www.ti.com/tool/TI-JESD204-IP> (visited on 03/02/2024).
- [44] Analog Devices. Digital-to-Analog Converter Data Sheet AD9154. Rev. C. Analog Devices. 2017.
- [45] Robin Oswald. “Characterization and control of a cryogenic ion trap apparatus and laser systems for quantum computing”. Doctoral dissertation. Zurich, Switzerland: ETH Zurich, 2022.
- [46] Roland Matt. “Parallel control of a dual-isotope trapped ion register”. Doctoral dissertation. Zurich, Switzerland: ETH Zürich, 2023.
- [47] Martin Stadler. “Classical control in trapped-ion quantum physics”. Note: This thesis was not yet published at time of writing. Doctoral dissertation. Zurich, Switzerland: ETH Zürich, 2024.
- [48] Texas Instruments. AFE80xx RF Sampling Transceiver. Data Sheet. Oct. 2022. URL: <https://www.ti.com/product/AFE8000>.

- [49] AFE8000EVM Evaluation Module. <https://www.ti.com/tool/AFE8000EVM>. (Visited on 25/01/2024).
- [50] Prassanna Shanmuga Sundaram. “Development of a FPGA-based True Random Number Generator for Space Applications”. Master thesis. Linköping, Sweden: Linköping Institute of Technology, 2010.
- [51] William L. Dunn and J. Kenneth Shultis. “Chapter 3 - Pseudorandom Number Generators”. In: *Exploring Monte Carlo Methods (Second Edition)*. Ed. by William L. Dunn and J. Kenneth Shultis. Second Edition. Elsevier, 2023, pp. 55–110. ISBN: 978-0-12-819739-4. DOI: <https://doi.org/10.1016/B978-0-12-819739-4.00011-1>. URL: <https://www.sciencedirect.com/science/article/pii/B9780128197394000111>.
- [52] Ulrich Jetzek. Galois Fields, Linear Feedback Shift Registers and their Applications. Carl Hanser Verlag, 2018. ISBN: 978-3-446-45140-7. DOI: <https://doi.org/10.3139/9783446456136>. URL: <https://www.sciencedirect.com/book/9783446451407/>.
- [53] Analog Devices. Pseudo Random Number Generation Using Linear Feedback Shift Registers. 2010. URL: <https://www.analog.com/en/design-notes/random-number-generation-using-lfsr.html> (visited on 25/01/2024).
- [54] G. E. P. Box and Mervin E. Muller. “A Note on the Generation of Random Normal Deviates”. In: *The Annals of Mathematical Statistics* 29.2 (1958), pp. 610–611. DOI: 10.1214/aoms/1177706645. URL: <https://doi.org/10.1214/aoms/1177706645>.
- [55] N.J. Kasdin. “Discrete simulation of colored noise and stochastic processes and $1/f^\alpha$ power law noise generation”. In: *Proceedings of the IEEE* 83.5 (1995), pp. 802–827. DOI: 10.1109/5.381848.
- [56] James McCartney Richard Voss. DSP generation of Pink (1/f) Noise. 1978. URL: <https://www.firstpr.com.au/dsp/pink-noise/> (visited on 30/01/2024).
- [57] Robert Jördens. ARTIQ Feature Request: better sin/cos generator. 2017. URL: <https://github.com/m-labs/artiq/issues/1342> (visited on 03/02/2024).
- [58] Texas Instruments. LMK0482x Ultra Low-Noise JESD204B Compliant Clock Jitter Cleaner With Dual Loop PLLs. SNAS605AS – March 2013 – Revised May 2020. Texas Instruments. May 2020. URL: <https://www.ti.com/product/LMK04828>.
- [59] Xilinx, Inc. 7 Series FPGAs GTX/GTH Transceivers User Guide. Version 1.12.1. Xilinx, Inc. Aug. 2018.
- [60] Analog Devices, Inc. High Performance, 3.2 GHz, 14-Output Jitter Attenuator with JESD204B. Rev. C. Analog Devices, Inc. One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A., Sept. 2021. URL: <https://www.analog.com/media/en/technical-documentation/data-sheets/hmc7044.pdf>.
- [61] Analog Devices. HMC7044 Evaluation Board Schematic. Analog Devices. 2016. URL: https://www.analog.com/media/en/technical-documentation/evaluation-documentation/HMC7044_EVAL_Schematic.pdf.
- [62] Analog Devices. AFE8000 Evaluation Board Schematic. Rev. A. Texas Instruments. 2022.
- [63] Eric W. Weisstein. Random Walk 1-Dimensional. URL: <https://mathworld.wolfram.com/RandomWalk1-Dimensional.html> (visited on 12/02/2024).
- [64] Sourabh Gupta and Vinod Paliakara. Using Windowing With SNRBoost3G Technology. Application Report SLAA445. Texas Instruments Incorporated, Aug. 2010.
- [65] E. Knill et al. “Randomized benchmarking of quantum gates”. In: *Physical Review A* 77.1 (Jan. 2008). ISSN: 1094-1622. DOI: 10.1103/physreva.77.012307. URL: <http://dx.doi.org/10.1103/PhysRevA.77.012307>.

Glossary

- AFCK** AMC FMC carrier Kintex. 26, 29
- AOM** acousto-optic modulator. 25, 29, 46, 47
- ASIC** application specific integrated circuit. 21, 22
- AWG** arbitrary waveform generator. 26, 34
- BRAM** block RAM. 21
- BSB** blue sideband. 7, 9, 26
- CLB** configurable logic block. 21
- CML** current mode logic. 22
- CMOS** complementary metal–oxide–semiconductor. 22
- DDS** direct digital synthesis. 26, 34
- DSB** double-sideband. 13
- DSP** digital signal processing. 21, 33, 36
- eQual** encoded Qubit alive. 3, 21, 24, 25, 29, 39, 41, 46
- FMC** FPGA mezzanine card. 26, 28, 29, 34, 36–38
- FPGA** field-programmable gate array. 21, 22, 26, 28–30, 34, 36, 37
- HDL** hardware description language. i, 22
- I/O** input/output. 38
- IEEE** Institute of Electrical and Electronics Engineers. 13
- IIR** infinite impulse response. 32
- IOB** input output bank. 21
- JESD204** standard defining an interface between data converters and processing hardware (e.g. FPGAs). i, 22, 28, 36
- JESD204B** version B of the JESD204 standard. 22–24, 28, 34, 36
- LFSR** linear feedback shift register. 30, 31
- LMFC** local multiframe clock. 22–24

- LO** local oscillator. 35
- LUT** lookup table. 34, 35
- LVDS** low voltage differential signaling. 22
- M-ACTION** Modular Advanced Control of Trapped IONs. 25
- MS** Mølmer-Sørensen. i, 7–9, 18, 26, 44, 45, 47, 52, 54
- NCO** numerically controlled oscillator. 35
- NISQ** noise intermediate-scale quantum. 1
- PCB** printed circuit board. 25
- PLL** phase locked loop. 28, 36, 37, 39
- PMT** photomultiplier tube. 25, 26
- QHO** quantum harmonic oscillator. 4, 5
- QuENCH** Quantum Experiment Next-Generation Control Hub. 21, 25–28, 34, 35, 37, 38
- RF** radio frequency. i, 2, 10, 21, 22, 25, 26, 28–30, 34–38, 44, 46, 51, 52, 54
- RSB** red sideband. 7, 9, 26
- SoC** system on a chip. 26
- SSA** signal source analyser. 13
- SSB** single-sideband. 13, 16, 39
- TIQI** Trapped Ion Quantum Information. i, 3, 21–27
- UHV** ultra-high vacuum. 24
- VHDL** Very High Speed Integrated Circuit (VHSIC) Hardware Description Language. 22
- VHSIC** Very High Speed Integrated Circuit. 60



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every written paper or thesis authored during the course of studies. In consultation with the supervisor, one of the following three options must be selected:

- I confirm that I authored the work in question independently and in my own words, i.e. that no one helped me to author it. Suggestions from the supervisor regarding language and content are excepted. I used no generative artificial intelligence technologies¹.
- I confirm that I authored the work in question independently and in my own words, i.e. that no one helped me to author it. Suggestions from the supervisor regarding language and content are excepted. I used and cited generative artificial intelligence technologies².
- I confirm that I authored the work in question independently and in my own words, i.e. that no one helped me to author it. Suggestions from the supervisor regarding language and content are excepted. I used generative artificial intelligence technologies³. In consultation with the supervisor, I did not cite them.

Title of paper or thesis:

Impact of RF noise on Trapped-ion Quantum Gate Fidelity

Authored by:

If the work was compiled in a group, the names of all authors are required.

Last name(s):

Dörrer

First name(s):

Simon

With my signature I confirm the following:

- I have adhered to the rules set out in the Citation Guide.
- I have documented all methods, data and processes truthfully and fully.
- I have mentioned all persons who were significant facilitators of the work.

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

Place, date

Zürich, 21.2.2024

Signature(s)

Simon Dörrer

If the work was compiled in a group, the names of all authors are required. Through their signatures they vouch jointly for the entire content of the written work.

¹ E.g. ChatGPT, DALL E 2, Google Bard

² E.g. ChatGPT, DALL E 2, Google Bard

³ E.g. ChatGPT, DALL E 2, Google Bard