

**Developing tools for cold atom
research : an ultracold Na-K
experiment, proposals for unitary
 k -designs, and a web interface**

Rohit Prasad Bhatt

2022

Dissertation
submitted to the
Combined Faculties of the Natural Sciences and Mathematics
of the Ruperto-Carola-University of Heidelberg, Germany
for the degree of
Doctor of Natural Sciences

Put forward by
Rohit Prasad Bhatt
born in: Rudraprayag, India
Oral examination: 18 July 2022

**Developing tools for cold atom
research : an ultracold Na-K
experiment, proposals for unitary
 k -designs, and a web interface**

Referees : Prof. Dr. Markus K. Oberthaler
Prof. Dr. Lauriane Chomaz

Dedicated to the universe for letting life sprout.

Abstract

Synthetic quantum systems present a highly controllable platform for understanding fundamental physics and development of associated technological applications. This thesis reports on building tools for research on synthetic quantum systems, and is divided into three parts. The first part describes the development and set-up of a new experimental platform based on cold atomic mixtures of sodium and potassium. During this thesis, a dual-species magneto-optical trap was realized, followed by transferring sodium atoms into magnetic and optical traps. The second part presents theoretical proposals for implementing random unitary operations (unitary k -designs) in cold atom systems. Unitary k -designs offer a versatile tool for characterizing quantum systems without the need for tomographic protocols. The third and final part describes the development of *qlue*, which is a web framework enabling standardized internet access to various platforms of quantum research in a user-friendly manner. This can facilitate access to these platforms for a larger group of researchers.

Zusammenfassung

Synthetische Quantensysteme stellen eine hochgradig kontrollierbare Plattform für das Verständnis der fundamentalen Physik und die Entwicklung der damit verbundenen technologischen Anwendungen dar. Diese Arbeit berichtet über die Entwicklung von Methoden für die Forschung an synthetischen Quantensystemen und ist in drei Teile gegliedert. Der erste Teil beschreibt die Entwicklung und den Aufbau einer neuen experimentellen Plattform, die auf kalten atomaren Mischungen von Natrium und Kalium basiert. Im Rahmen dieser Arbeit wurde eine magneto-optische Doppel-Spezies-Falle realisiert, gefolgt vom Transfer von Natriumatomen in magnetische und optische Fallen. Im zweiten Teil werden theoretische Konzepte für die Implementierung zufälliger unitärer Operationen (unitäre k -Designs) in kalten Atomsystemen vorgestellt. Unitäre k -Designs bieten ein vielseitiges Instrument zur Charakterisierung von Quantensystemen ohne die Notwendigkeit von tomographischen Protokollen. Der dritte und letzte Teil beschreibt die Entwicklung von *qlue*, einem Web-Framework, das einen standardisierten Internetzugang zu verschiedenen Plattformen der Quantenforschung auf benutzerfreundliche Weise ermöglicht. Dies kann den Zugang zu diesen Plattformen für eine größere Gruppe von WissenschaftlerInnen erleichtern.

Acknowledgements

The journey of my PhD has been very exciting. Words cannot express the importance of people who have made this possible. My apologies in advance for not being able to do justice.

I would like to thank my supervisors Prof. Dr. Markus K. Oberthaler and (retired) Dr. Fred Jendrzejewski. Dr. Jendrzejewski was kind enough to offer me a PhD position trusting in me to start a new experiment. He has been of immense support throughout the journey and offered me a great selection of projects which I thoroughly enjoyed to be involved in. Prof. Oberthaler has been a strong pillar of support in testing times and was readily available to resolve any concerns in tough situations. Thanks to both of you.

I would like to thank my PhD referees : Prof. Dr. Markus K. Oberthaler , Prof. Dr. Lauriane Chomaz, Dr. Martin Gärttner and Prof. Norbert Frank for agreeing to be a part of my examining committee. A huge thanks to Prof. Oberthaler and Prof. Chomaz for understanding the situation and giving all their support.

Very special thanks are due to my team members : Lilo and Jan. Thanks for all the fun we had over the past four years and all your support till the last moment of writing my thesis. I would also like to thank all members of the NaLi-SoPa-NaKa family : Dr. Lisa Ringena, Dr. Apoorva Anant Hegde, Andy, Ingrid, Yannick, Noah and Gopal. I also express my gratitude to the past members of this family.

Sincere thanks are also due to each and every member of the SynQS family for their help and support. The rich exchanges within this awesome group has been a constant source of learning for me.

I would like to thank all my collaborators with whom I had the fortune to work. Thank you Dr. Valentin Kasper, Dr. Daniel Egger, Laurin Fischer and Dr. Christian Gogolin.

I am also deeply indebted to our secretaries : Mrs. Dagmar Hufnagel, Mrs. Christiane Jäger, and Mrs. Elke Schmetzer. Thanks for all your help in resolving confusing administrative situations.

I would like to acknowledge the special role of the University of Heidelberg, Kirchoff institute for Physics and HGSFP in supporting me throughout my PhD. I owe a humble gratitude to Mr. Lamade, Mr. Spiegel and all the Diplomandenwerkstatt staff, for helping us with various projects for the lab.

Acknowledgements

Special thanks are due to Studentenwerk Heidelberg for providing me an accommodation for four years. I am thankful to all my flatmates as well for making living in Heidelberg very enjoyable.

I am grateful to my family members for their immense support without which this work would not have materialized. I owe the accomplishments of my life to all my teachers from Kindergarten to Graduation and the benign grace of Almighty God.

With the blessings and love of all well-wishers, to this thesis I put my name.

Rohit Prasad Bhatt

Contents

Abstract	i
Zusammenfassung	i
Acknowledgements	iii
Contents	v
List of Figures	ix
List of Tables	xi
1 Introduction	1
1.1 Quantum simulation and ultracold quantum gases	1
1.1.1 Why Na-K?	4
1.2 Random unitaries	4
1.3 Web interface for cold atom backends	5
1.4 Outline of the thesis	5
I Na-K ultracold mixture experiment	7
2 Experimental setup	9
2.1 SoPa : at a glance	9
2.2 Vacuum system	10
2.2.1 2D-MOT chamber	12
2.2.2 Science chamber	14
2.2.3 Achieving vacuum	15
2.3 Laser setup and frequency stabilization	16
2.3.1 Sodium	18
2.3.2 Potassium	19
2.4 Laser cooling and trapping	20
2.4.1 Laser cooling	20
2.4.2 Doppler limit and sub-Doppler cooling	21
2.4.3 Magneto-Optical Trap	22
2.4.4 2D-MOT	23
2.4.5 3D-MOT	24
2.5 Fluorescence imaging	25

2.5.1	Imaging system	26
2.5.2	Extracting atom number and temperature	27
2.6	Experiment control system	28
2.6.1	Is labsript enough?	33
3	Atom number dynamics in a MOT	35
3.1	Atom counting in a MOT	35
3.1.1	Why do atom counting in a MOT?	35
3.1.2	Controlling atom number in a MOT	37
3.1.3	Imaging noise	37
3.2	Atom number dynamics and statistical analysis	38
3.2.1	Sodium	39
3.2.2	Potassium	40
3.2.3	Unified statistical analysis	40
3.3	Simulation of the atom dynamics	42
3.4	Immersing sodium in potassium	44
3.4.1	Full counting statistics of sodium	45
3.4.2	Effect of inter-species interaction	46
4	Towards ultracold quantum mixture	49
4.1	Bose-Einstein condensation	49
4.2	Electromagnet coil upgrade	51
4.3	Optimizing sodium 2D and 3D-MOT	53
4.4	Compressed MOT and optical molasses	55
4.5	Magnetic trapping and microwave evaporation	56
4.5.1	Experimental results	57
4.6	Optical dipole trap	60
4.6.1	Experimental results	61
4.7	Optical tweezers	64
4.7.1	Array of tweezers and defect-free sorting	64
4.7.2	Problems with potassium vacuum	64
4.7.3	Experimental details	65
4.8	Summary and future plans	67
II	Random unitaries with ultracold atoms	69
5	Random Unitaries : an overview	71
5.1	Haar measure	71
5.2	Unitary k -designs	72
5.3	Frame potentials	73
5.4	Applications of random unitaries	73
5.4.1	Benchmarking and verifying quantum simulators	74
5.4.2	Characterizing quantum states	74

5.4.3	Quantum phases and phase transitions	75
5.4.4	Quantum chaos and information scrambling	76
6	Generating Random Unitaries	77
6.1	Approximate unitary k -design in collective spins	77
6.2	Approximate unitary k -design for 1D-fermion system	80
6.3	Summary	82
III	Qlue: a web interface for cold atom backends	85
7	Web technology for cold atoms	87
7.1	Qlue architecture	88
7.1.1	The JSON schemas	89
7.2	The client	90
7.2.1	QisKit and PennyLane plugins	90
7.3	The server	92
7.3.1	Django web app with URLs	92
7.3.2	The view functions	93
7.3.3	The Dropbox database	94
7.3.4	An example workflow	95
7.4	The backends	97
7.4.1	The simulator backend	97
7.4.2	The experiment backend	99
7.5	Usage examples	101
7.5.1	MOT loading on a experiment	101
7.5.2	Spin-squeezing in a BEC	102
7.5.3	Fermi Hubbard model dynamics	102
7.6	Summary	103
8	Conclusion and outlook	105
	Appendices	107
A	Technical details	109
A.1	Electronics	109
A.2	Analysis code structure	111
B	Tweezers sorting algorithm	115
B.1	The scheme	115
B.2	Calculation methods	117
B.3	Performance	118
B.3.1	Setting 1:	118
B.3.2	Setting 2:	118
B.4	Code	119

Bibliography	127
Erklärung	149

List of Figures

1.1	Quantum Gases as an interdisciplinary research field	3
2.1	SoPa beginning	10
2.2	SoPa now	10
2.3	Vacuum system	11
2.4	2D-MOT chamber design	13
2.5	Science chamber design	14
2.6	Baking curve	16
2.7	Laser table setup for sodium and potassium.	17
2.8	Spectroscopy setup.	18
2.9	Sodium spectroscopy	19
2.10	Potassium spectroscopy	20
2.11	MOT working principle	22
2.12	Sodium 2D MOT magnetic field.	24
2.13	First picture of our dual-species MOT.	25
2.14	Imaging system.	27
3.1	Experimental sequence for atom counting.	39
3.2	Time traces of sodium and potassium.	39
3.3	Accurate atom counting of sodium.	40
3.4	Characterization of atom number fluctuations.	41
3.5	Simulation of the imaging statistics.	43
3.6	Counting statistics of sodium.	46
4.1	MOT coil manufacturing	52
4.2	MOT coil magnetic field measurement	53
4.3	MOT optimization result	54
4.4	Dark-SPOT MOT temperature measurement	55
4.5	Zeeman shift of sodium ground states	58
4.6	Magnetic trap lifetime measurement	58
4.7	Setup of the crossed dipole trap	62
4.8	Image of atoms captured in the crossed dipole trap	63
4.9	Schematic of the tweezer setup	66
4.10	Indication of tweezer trapping	67
6.1	Unitary k -designs for a collective spin qudit.	79

6.2	Unitary k -designs for fermions	82
6.3	Unitary k -designs for fermions in various sectors	83
7.1	Qlue architecture.	89
7.2	Dropbox folder structure	95
7.3	Simulator backend files	98
7.4	MOT loading in SoPa experiment via qlue	101
7.5	Simulating spin-squeezing in a BEC via qlue	102
7.6	Simulating Fermi Hubbard model dynamics via qlue	103
A.1	Photo-diode circuit schematic and PCB layout	110
A.2	Shutter controller circuit schematic	110
A.3	IGBT switching circuit schematic	111
B.1	Sorting algorithm	116

List of Tables

2.1	Table of laser parameters	25
3.1	Comparison of mean load and loss probabilities	47
4.1	Atom number and temperature at different stages	59
4.2	Comparison of magnetic trap evaporation settings	59
7.1	Qlue code organization.	90
7.2	Example JSON for backend capabilities	91
7.3	JSON schema for user submitted jobs.	91
7.4	JSON schema for backend results	92
A.1	A layout of the analysis file	111
B.1	Sorting performance for 1D array of 10 tweezers	118
B.2	Sorting performance for 1D array of 20 tweezers	118

Introduction

Quantum mechanics is one of the most rigorously verified theories in experiments to a high degree of precision. The field of atomic, molecular and optical (AMO) physics has both contributed to and benefited from the development of quantum mechanics. Amalgamation of ideas from these fields has led to the current state-of-the-art experiments where one can prepare clean quantum systems and probe them for a better understanding of quantum mechanics and other areas of physics. This requires substantial efforts on building platforms for performing experiments, developing theoretical tools to understand those systems and using technology to make the platforms available to a wider group of researchers. The importance and usefulness of these aspects is highlighted in the following sections.

1.1 Quantum simulation and ultracold quantum gases

The concept of simulating a physical system refers to building a dummy system which reproduces the behavioural aspect of the original physical system. This idea dates back centuries to systems built for tracking planetary motion [1]. Building a model for a particular system is usually referred to as simulation. However, if the model is capable of simulating any system with arbitrary precision, the process is called computation [2]. In modern times, an abstract machine capable of solving wide range of problems was conceptualised as the so-called Turing machine [3].

At a microscopic level, the systems occurring in nature are governed by the laws of quantum mechanics. Physicists have developed a theoretical description of quantum mechanics, which can be used to simulate the microscopic dynamics of any quantum system on a classical computer. However, the description suffers from an inherent flaw in that the amount of computing resources required for the simulation grow exponentially in the size of the system being simulated. The power of exponential growth is so daunting that simulating the microscopic dynamics of systems with more than few tens of particles would need more computing power than available to all of humanity. Theoretical efforts have proposed approximate methods like quantum Monte-Carlo simulation [4, 5], tensor network methods [6], artificial neural networks

[7, 8] etc. However, these methods have a limited range of applicability and suffer from their own drawbacks.

At the beginning of the 1980s, Manin [9] and Feynman [10] independently proposed the idea of using quantum systems themselves to build “machines” which could model other quantum mechanical systems. More than a decade later first theoretical [11] and experimental [12] results validated the original idea. This led to the genesis of the field of quantum simulation where one uses a controllable quantum system to study the physics of another less controllable quantum system. Apart from lifting the curse of dimensionality, quantum simulators can also be used for verifying the predictions of a theoretical model, e.g. if it predicts a phase transition under a given set of parameters etc. An added advantage of quantum simulators is that they can be engineered to exhibit slower time scales than the physical systems they model. This allows a real-time measurement of particle and spin dynamics.

In the paradigm of quantum simulation, there are two major categories : Digital quantum simulation (DQS) and analog quantum simulation (AQS). In DQS, the simulator offers a set of universal gates and the unitary evolution of any quantum system can be approximated by a sequence of gates from the universal set. Such a simulator is also called a universal quantum computer or a quantum Turing machine. The precision of approximation can be improved at the cost of longer circuits and by using quantum error correction. In AQS, the simulator directly implements the dynamics of the system under study, so it is not universal like DQS. However, at the current state due to unavailability of a scalable fault-tolerant DQS, an AQS still provides the advantage that it can be used for studying systems with larger sizes for longer time evolution. For a comprehensive review on quantum simulation, the reader is referred to [2, 13–15].

Accompanied by seminal progress in experimental techniques, the engineering of controlled quantum systems has been possible on several platforms (see references in [2, 13–15]) like ultracold quantum gases (atoms and molecules), trapped ions, nuclear spins (NMR and NV centers), photonic systems, quantum dots and superconducting circuits. The young field of ultracold quantum gases has seen a rapid progress following the realization of Bose-Einstein condensation [16–18] and Fermi degeneracy [19] in dilute atomic vapor. These systems are realized in clean vacuum environments at temperatures close to absolute zero where quantum mechanical effects become dominant. The development of techniques for precise coherent control of individual particles and their detection has made quantum gases a powerful test bed for exploring a wide range of fundamental physics. Figure 1.1 tries to capture the intersection of this relatively new field with other areas in physics.

Ultracold atoms are one of the most extensively used systems for quantum simulation [21–25]. For this purpose, the atoms are usually trapped in external potentials like optical tweezers [26, 27] or optical lattices [28, 29]. The tunability of these potentials

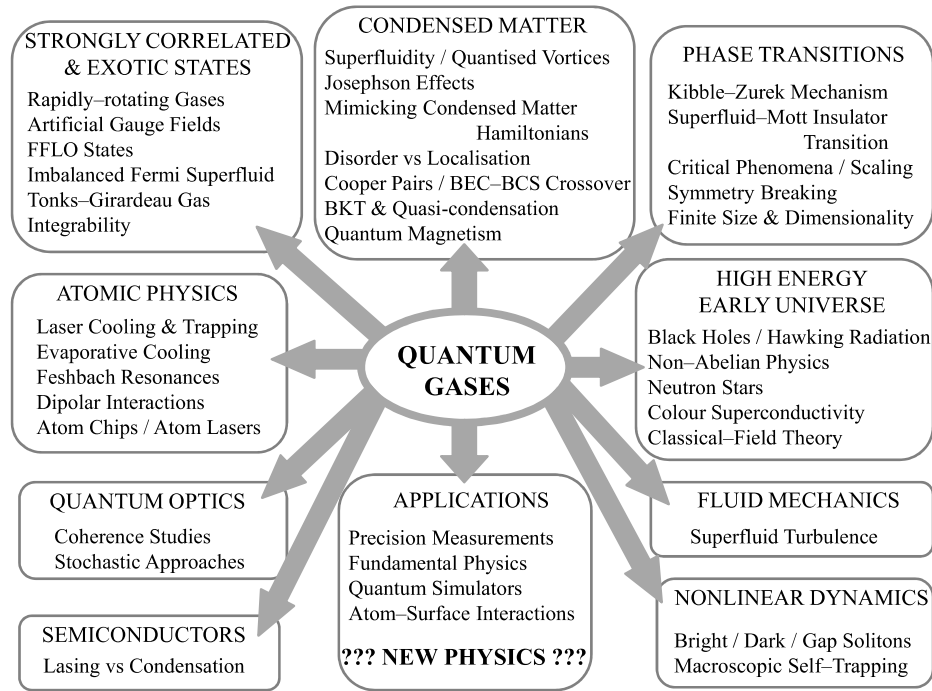


Figure 1.1: Quantum Gases as an interdisciplinary research field. Figure taken from [20] with permission.

allows changing the trap depth, geometry, dimensionality etc. in a controlled fashion. The inter-atomic interactions can also be tuned using Feshbach resonance [30]. Furthermore, the imaging methods have achieved single atom and single site resolution in quantum gas microscope experiments [31–34]. These developments have created profound experimental results, like observation of superfluid-Mott insulator transition [35], BEC-BCS crossover [36, 37], and quantum simulation of hallmark models like the Fermi-Hubbard model [38, 39] or spin models [40–43] etc.

Ultracold atomic mixtures extend the horizon by encompassing a wider range of physical phenomenon in compound many-body quantum systems. Such systems usually have two atomic species as quantum degenerate gases and offer extra tuning knobs like inter-species Feshbach resonance, relative population, mass imbalance etc. Coupling of the two atomic species can result in interesting properties, which makes them suitable to quantum simulate a wide range of physics like quantum thermal cycles [44, 45], impurity [46, 47] and quasi-particle physics [48, 49], high energy physics [50, 51] etc. These systems have also been proposed as viable platforms for error-corrected quantum computation [52].

Our new sodium-potassium (Na-K) mixture experiment is set up for using it as a platform to study some of the above mentioned ideas. In our experiment, we have demonstrated laser cooling of both species in a dual species magneto-optical trap (MOT). We have used the magneto-optical trap to perform single atom counting of sodium and studied the effect of potassium on the stochastic atom number dynamics

in a few atom sodium MOT. Furthermore, we have demonstrated capture of sodium atoms in magnetic and optical traps which is required for cooling to quantum degeneracy.

1.1.1 Why Na-K?

An important requirement for our quantum gas mixture experiment is to engineer controlled coupling between the two species. This will allow us to explore a wide range of physics. For this reason, the choice of sodium and potassium is very important, because this mixture has favorable inter-species scattering properties in terms of availability of Feshbach resonances at moderate magnetic fields [53, 54]. From experiments on our older sodium-lithium (Na-Li) machine, we learnt that inter-species spin-exchange interaction plays a key role in determining the timescales at which experimentally relevant dynamics occur [50]. This interaction is about 50 times stronger for Na-K as compared to Na-Li. This would allow faster dynamics, thus reducing evolution time and suppressing effect of temporal drifts. Finally, since sodium and potassium have been well studied in our group, we can use the in-house knowledge and experience.

1.2 Random unitaries

Owing to the rapid progress in the research field of quantum simulation, larger quantum systems have come within the reach of experimental realization. As the size of these systems grow, the dimension of the associated Hilbert space also grows. Current state of the art quantum technology platforms have reached dimensions of 10^{30} , which have already made the demonstration of quantum supremacy possible [55]. However, tools for characterizing the quantum systems at these exponentially large scales are still in the nascent stage. A very promising approach for efficient probing of quantum systems is using statistical correlations between outcomes of measurements performed in random measurement basis [56]. To perform measurement in these random bases, one needs to apply “random unitaries” just before the usual measurement in the computational basis.

It turns out that the random unitaries are a useful resource for a lot of applications. They provide a comprehensive framework to benchmark the performance and capabilities of quantum computers [57, 58]. Several quantum information protocols [59–61] rely on the ability to produce random unitaries. Their application to random state preparation has been proposed for metrology applications [62]. For quantum many body systems, random unitaries have been proposed as a powerful tool to characterize quantum states and to extract several properties of quantum systems [63–65].

Given the importance of random unitaries, a considerable effort has been put into realizing them in physical systems. The current work reports on our proposal for

realizing random unitaries in spin systems of bosons and lattice system of fermions. On bosonic spin systems, our results complement contemporary research effort by proposing implementation of random unitaries using protocols already realized in experiments. On fermionic lattice systems, we relax the experimental requirements for realizing random unitaries by proposing a simpler protocol, which can be very useful for certain atomic species like lithium-6 (see section 6.2).

1.3 Web interface for cold atom backends

As technological progress continues, the quantum simulation and quantum computing capabilities of experiments with cold atoms and trapped ions keeps improving. In these platforms, each experimental group makes their own choice of the programming framework for controlling the experiment. The actual code that controls various devices in the lab is very specialized and does not represent the quantum simulation/-computing in an easy to understand way. For this reason, it would be beneficial to have a higher level description of the experiment.

This higher level description could also prove very useful for communicating the capabilities of the machine to a remote user who need not be concerned with the intricate details of the experiment setup. Such an approach allows a greater reach and ease of use for remote researchers who can be for example a theory collaborator. With the higher level description one can also build a simulator running on a computer which reproduces the behaviour of the experiment upto certain system size. The same higher level description can be used to execute “experiments” on both the simulator and the real hardware.

To realize these ideas, along with the higher level description, one also needs a service, which can allow an authorized remote user to submit “instructions” to the platform/simulator and fetch results of already submitted ones over the internet. Considering all the requirements, we developed a library called *qlue*, which is agnostic to the details of implementation on a particular platform. It allows researchers to provide a higher level description for their experiments and provide a controlled access to their machines for a remote user. Furthermore, it allows the research groups to setup a simulator on a computer, which mimics the experiment and can also be provided to the remote user for example to try out ideas of a possible research study.

1.4 Outline of the thesis

This thesis is organised into three parts:

Part I comprises [Chapter 2](#), [Chapter 3](#) and [Chapter 4](#). These chapters describe respectively the experimental setup, the recent results on single atom counting and the efforts towards achieving quantum gases.

Part II comprises **Chapter 5** and **Chapter 6**. These chapters describe respectively the theory of random unitaries and our results on proposed methods for implementing them with bosonic and fermionic cold atom systems.

Part III comprises **Chapter 7** and it describes the architecture of *qlue*. It also presents example workflows to demonstrate some use cases of *qlue*.

Chapter 8 summarizes the thesis and discusses next steps for all three parts.

Appendix A contains technical details about some electronic components and layout of python analysis code used in the experiment.

Appendix B describes the python code for assembling a defect free array of optical tweezers.

PART I

Na-K ultracold mixture experiment

Experimental setup

Ultracold atoms can be used as pristine synthetic quantum systems, whose properties can be precisely tuned with external electromagnetic fields. Owing to their sensitivity, these systems pose demanding requirements for experiments which aim to employ them for quantum simulation. This comprises a wide range of criteria like an ultra high vacuum (UHV) environment, controlled addressing and manipulation of atomic energy levels, and noise-suppressed detection of the atomic signal etc. Ultracold mixture experiments exhibit even more complexity due to an interplay of the constituent atomic species. In this chapter, I describe various components of our sodium-potassium mixture experiment (SoPa) and discuss the methods we have used to improve the versatility and stability of each component. This chapter culminates into reporting the observation of the dual-species 3D-MOT on our setup thus establishing the status of the SoPa machine as a mixture experiment. Some parts of this chapter taken verbatim from [66] are put in “quotes”.

2.1 SoPa : at a glance

The SoPa experiment started in 2018 with the goal of setting up a compact, robust and modular mixture experiment capable of stable continuous operation. At that point, we already had a running Na-Li mixture experiment (called NaLi) in our group. As a part of the bigger SYNQS group another potassium experiment (called BECK) had recently achieved their potassium BEC in August 2018. Our idea was to take the lessons from both machines while making possible improvements. For setting up the experiment, we had to move into the same room as the NaLi. This posed immense space constraints and we had to design everything for optimized space usage.

After accomplishing a dual-species MOT, we have come a long way towards achieving a sodium BEC and potassium single atoms in optical tweezers. Although both these milestones are not yet reached, the journey itself has been very engaging. This journey is best summarized with pictures (figure 2.1 and 2.2) of the lab area in 2018 and now. As shown in the figure 2.2 the experiment consists of three main optical tables. Two of them, which are stacked vertically, have the laser systems for sodium (lower one)

2. Experimental setup

and potassium (upper one), and the third one is the main table where the experiment apparatus is set up. Comparing these two images indicates the amount of effort and time spent into transforming a cluttered space in a lab into a new experiment. In the coming sections, I report on the steps taken to achieve the current status of the experiment.



Figure 2.1: SoPa lab area in 2018.

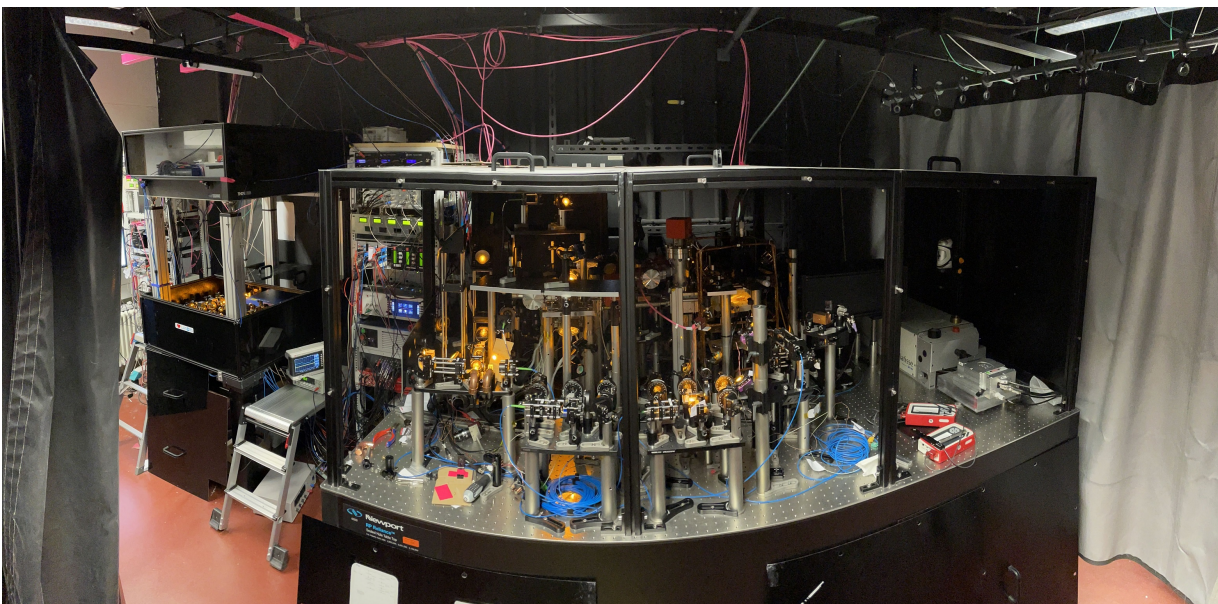


Figure 2.2: SoPa lab area in 2022.

2.2 Vacuum system

In order to isolate the cold cloud of atoms from the background gas, a clean vacuum is required. It is also necessary to sustain pressure gradients across different regions of the experimental setup. A higher pressure on the order of 10^{-8} mbar exists around the source of atoms (either alkali metal dispenser or an oven containing the pure

elemental form) while in the region where final experiments are performed, an ultra high vacuum (UHV) on the order of 10^{-11} mbar is required. “For ultracold mixture experiments with two atomic species, common designs [67–70, 50] consist of a dual-species oven and a single Zeeman slower connected to a science chamber where experiments are performed. In this design, the precooling stages of the two species are highly coupled, which makes the optimization of the system complex.”

“In our vacuum system, we decoupled the precooling stages of sodium and potassium up to the science chamber, as sketched in figure 2.3. The compact vacuum system contains two independent two-dimensional magneto-optical trap (2D-MOT) chambers[71, 72] for sodium and potassium and a dual-species science chamber. The two 2D-MOT chambers are connected to the science chamber from the same side under a 12.5° angle. Two gate valves ensure full decoupling of both species by isolating different chambers. Each region is pumped with its separate ion getter pump (*SAES Getters* pumps : *NEXTorr Z100* for 2D-MOTs and *NEXTorr D500* for science chamber).” The vacuum system is an assembly of several parts which have to be connected in a leak-tight manner. For this we mostly use Conflat (CF) flanges with copper/silver gaskets. In case the connection does not have a standard shape (for e.g. the viewports of our science chamber), we use an indium wire to get a UHV leak-tight seal.

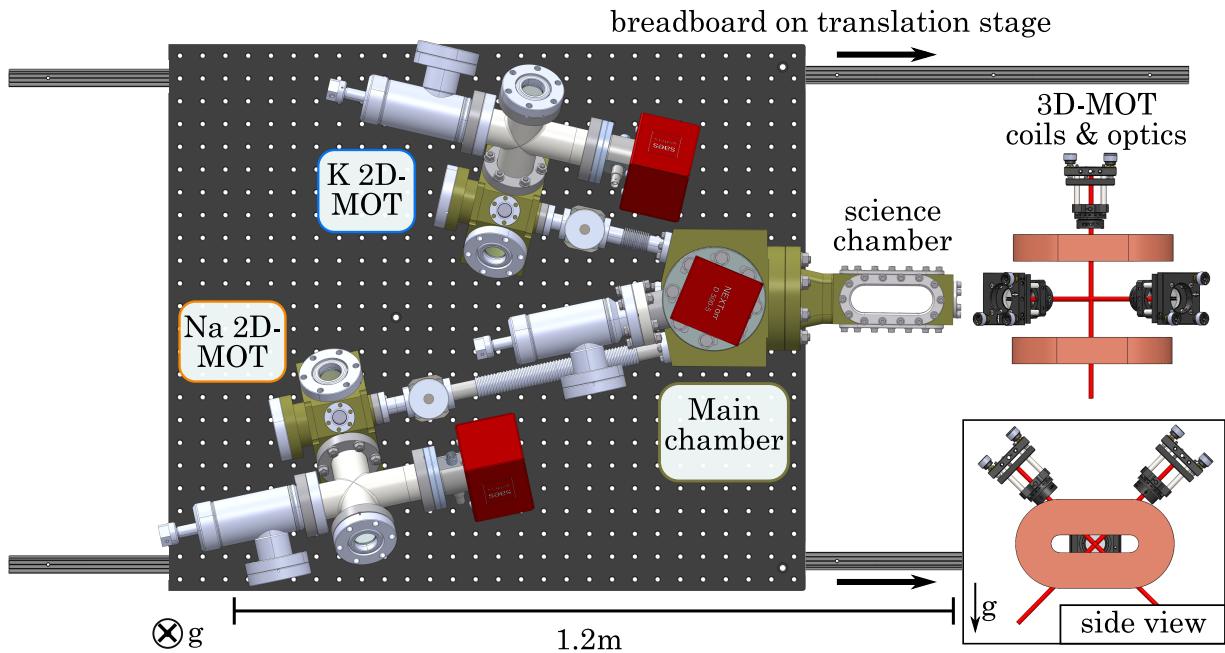


Figure 2.3: “Vacuum system. The separated 2D-MOT chambers are connected from the same side to the dual-species science chamber. The vacuum pumps are shown in red. The whole vacuum system is mounted on a translation stage, such that the science chamber can be moved out of the region of the 3D-MOT coils and optics.”

“The entire apparatus is mounted on a 600 mm x 700 mm aluminium breadboard, which is fixed to a linear translation stage (*Igus TS-01-25* and *TW-01-HKA*), inspired by a similar design in the group of Manuel Endres at Caltech [73]. Therefore, we are able to move the science chamber out of the contraction of 3D-MOT magnetic field

coils and optics. This allows for independent improvement of the vacuum system and in-situ characterization of the magnetic field at the position of the atoms.” Details of the translation stage can be found in the Masters thesis of Lilo Höcker [74].

2.2.1 2D-MOT chamber

“The design of our 2D-MOT setup is inspired by the design in the group of Gabriele Ferrari in Trento [71]. The chamber body is octagonal shaped and manufactured from titanium (fabricated by *SAES Getters*), where optical access is ensured by standard CF40 and CF16 fused silica viewports with broadband anti-reflection coating (BBAR coating).” A schematic of the 2D-MOT chamber is shown in figure 2.4. The CF40 viewports are used for the MOT beams (from sides) and push beam (from front), while the CF16 viewport is used for the Zeeman slower beam (from top) . One of the CF40 flanges hosts a four-way cross which has a MOT beam viewport and connects to the ion-getter pump. An all-metal angle valve is also attached to the cross and is used for connecting to the pre-pumping station during the pumping down process.

The back side of the 2D-MOT chamber is connected to the main chamber via a CF16 nipple, a gate valve and a CF16 bellow. The chamber body has four ridges for installing permanent magnets, two of them are above and below the push beam CF40 viewport and other two at the back (i.e. above and below the CF16 nipple flange). There are two additional ridges on the side which help in mounting of the 2D-MOT chamber.

“The 2D-MOT region has an oven containing a 1 g atomic ingot ampule. The oven is heated to 160 °C (70 °C) for sodium (potassium), thereby increasing the pressure to 10^{-9} mbar (high vacuum) in this region. To maintain an UHV in the science chamber, a differential pumping tube separates each 2D-MOT chamber from the science chamber.” It is a tube of 3 cm length and 3 mm inner diameter. In steady state the throughput across the differential pumping tube from both vacuum regions is same, i.e.,

$$\text{high vacuum throughput} = \text{ultra-high vacuum throughput} \quad (2.1)$$

$$C \times (P_{HV} - P_{UHV}) = P_{UHV} \times S_{UHV} \quad (2.2)$$

where we used two different expressions of throughput for the two vacuum regions. C denotes the conductance of the differential pumping tube, P_{HV} the high vacuum region pressure, P_{UHV} the ultra-high vacuum region pressure and S_{UHV} the ultra-high vacuum region pumping speed. Additionally since $P_{HV} \gg P_{UHV}$, so $P_{HV} - P_{UHV} \approx P_{HV}$ and we have :

$$\frac{P_{HV}}{P_{UHV}} = \frac{S_{UHV}}{C} \quad (2.3)$$

For a tube with circular cross-section, the conductance C in the molecular flow regime (in which collisions with chamber walls are much more frequent than collisions between gas particles) is given by the Knudsen equation[75]. For air at 20 °C, it

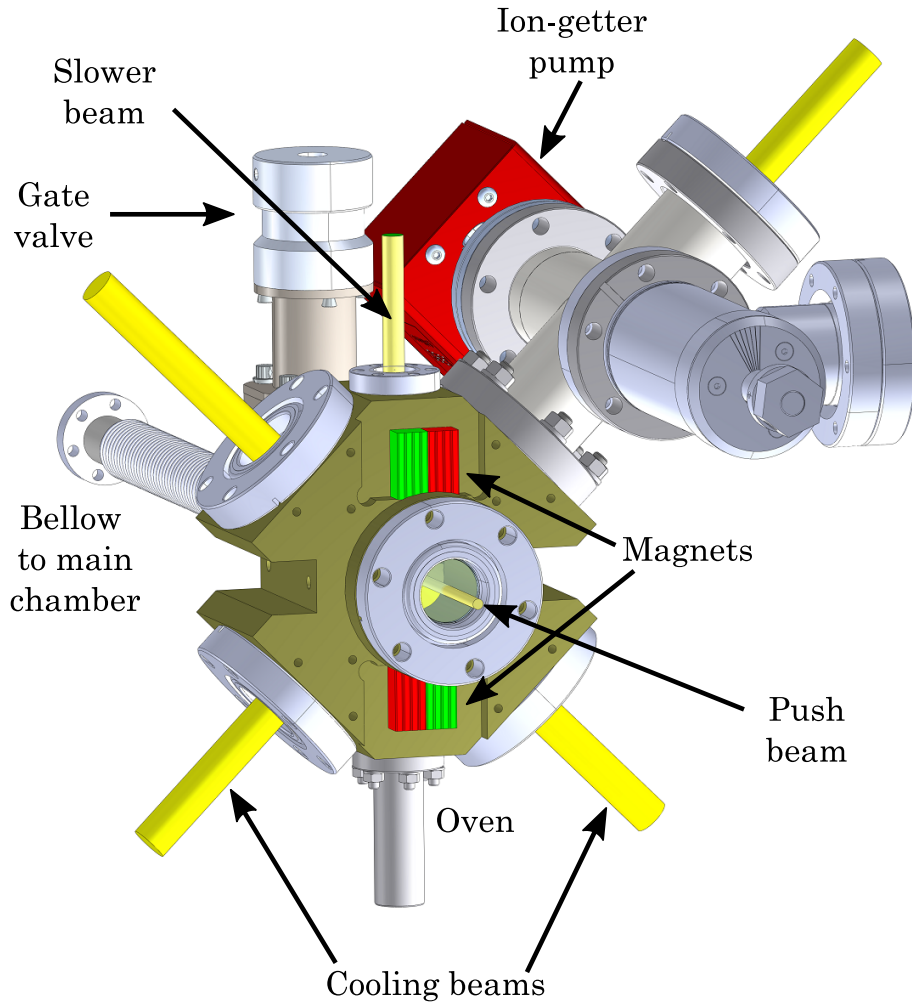


Figure 2.4: 2D-MOT setup and chamber design.

simplifies to[76]:

$$C \approx 12.1 \frac{d^3}{l} \quad (2.4)$$

where d and l are the tube diameter and length respectively. C is in litres per second when d and l are in centimeter. With our dimensions of $d = 0.3$ cm and $l = 3$ cm, we get $C \approx 0.1$ L s⁻¹. On the ultra-high vacuum side, we have a *NEXTorr D500* ion-getter pump with pumping speed $S_{UHV} = 500$ L s⁻¹. This gives:

$$\frac{P_{HV}}{P_{UHV}} \approx 10^4 \quad (2.5)$$

This means that we should be able to maintain a UHV on the order of 10^{-11} mbar while operating at 10^{-7} mbar in the 2D-MOT region. These numbers are by no means exact (for e.g. we do not have air at 20 °C) but they give a very rough estimate of the pressure regions we can operate in. In reality, we see that in our experiment when the 2D-MOT chamber pressure reaches $\sim 10^{-8}$ mbar, the science chamber reaches $\sim 10^{-10}$ mbar.

2.2.2 Science chamber

“The two atomic beams from the 2D-MOT chambers, coming in from the same side, have to intersect in the center of the science chamber. This constraint required us to use a custom rectangular titanium science chamber since for a glass chamber the glass-to-metal transition would require too much distance. Optical access for various laser beams and a high-resolution imaging system is maximized by four elongated oval viewports (fused silica, BBAR coating), which are sealed using indium wire[74].” The oval viewports on the side have dimensions of 90 mm x 15 mm while the ones on the top and bottom have dimensions of 90 mm x 30 mm. There is also a CF16 viewport on the front side. A schematic of the science chamber is shown in figure 2.5.

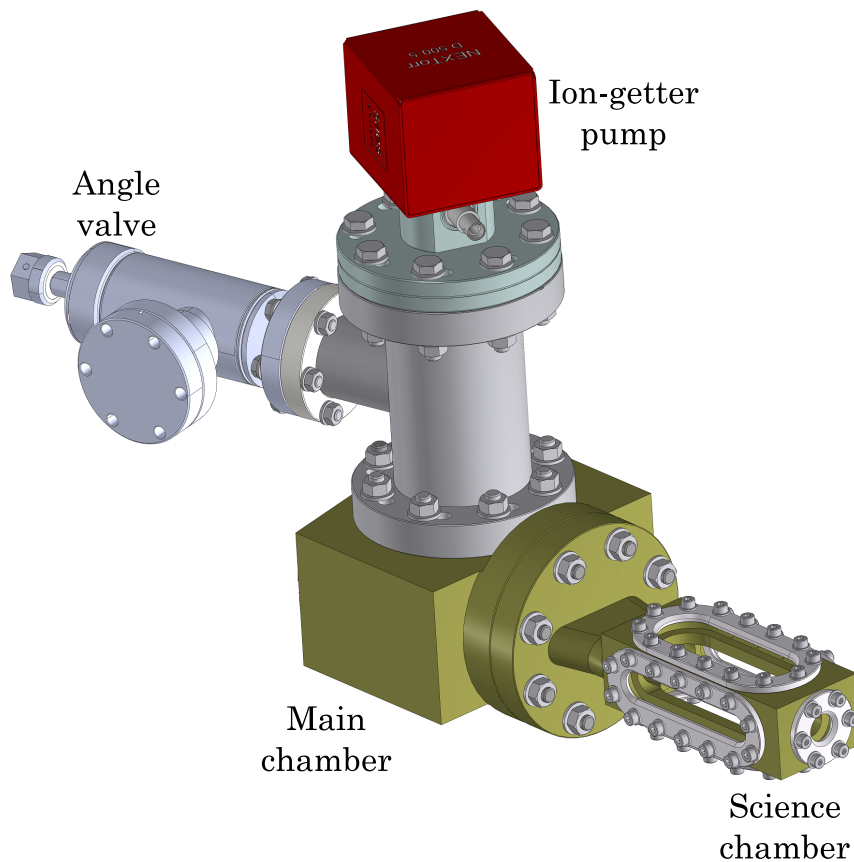


Figure 2.5: Science chamber design. The MOT beams and coils are omitted for clarity. For full setup see figure 2.3.

The science chamber is 55 mm wide and 40 mm high. Since our imaging system is below the science chamber, this asymmetric shape provides higher numerical aperture for imaging the atoms. On its rear side, the science chamber is attached to the main chamber which connects to the two 2D-MOT chambers via stainless steel CF16 bellows. The main chamber also has a T-connector which hosts on one side the ion-getter pump on a CF63 flange for creating UHV in the science chamber. The other side of the T-connector has an all-metal angle valve for connecting to the pre-pumping station during the pumping down process.

2.2.3 Achieving vacuum

To reach the vacuum levels required in our setup, we have to follow a sequence of pumping steps involving different types of pumps for different pressure ranges. We use the angle valves mentioned in sections 2.2.1 and 2.2.2 to connect a pre-pumping station to the setup. The pre-pumping station comprises a roughing pump, a turbo-molecular pump and a residual gas analyzer (RGA). While the roughing and turbo-molecular pump help in creating a fore-vacuum, the RGA is helpful in determining the composition of vacuum below a pressure of 10^{-5} mbar. We consider the vacuum composition clean if hydrogen has the highest partial pressure and contaminants like hydrocarbons are not present.

We start by connecting the pre-pumping station to the desired angle valve. Then we open the valve and turn on the roughing pump. After about 15 minutes, the roughing pump achieves a fore-vacuum of $\sim 10^{-3}$ mbar. This pressure is sufficient to turn on the turbo pump, which can achieve pressures of $\sim 10^{-8}$ mbar with overnight pumping.

If the vacuum composition is clean and the total pressure is $\sim 10^{-8}$ mbar, then we can start pumping with the ion-getter pump. For this, first the ion pump is flashed i.e. it is power cycled few times. Then we need to activate the getter material by heating it to elevated temperatures for some time (at about 500°C for 1.5 hours). After activation and cool-down, the ion-getter pump is turned on and the angle valve is closed, thus isolating the vacuum chamber from the pre-pumping station. Under the right conditions, the ion-getter pump can achieve $\sim 10^{-11}$ mbar in few hours.

To improve the quality of vacuum in terms of total pressure and clean composition, we perform baking. In this process we heat the vacuum chamber to temperatures above 100°C using glass fibre heaters. We maintain a heat gradient such that the temperatures decrease towards the pumping station. We also have to ensure the chamber itself is uniformly heated. This is done by wrapping the setup in aluminium foil. A typical baking curve showing evolution of the total pressure is presented in figure 2.6.

As is evident from the figure 2.6, with increasing temperature the pressure first increases to a maximum value (around 24 hours of baking), then it starts to drop even when the system is being heated and starts to settle at a steady value (around 50 hours of baking). The system is kept at this setting for 110 hours more and then at about 140 hours of baking, the temperature is lowered gradually which is followed by a sharp decrease of pressure. The final pressure at room temperature is about ten times smaller than starting pressure at room temperature.

Another factor which can hinder attaining UHV is any sealing between the flanges and gaskets that isn't leak-tight. This can be detected easily using the RGA and a Helium gas bottle. With a nozzle, we spray helium around all the connections in the vacuum setup while monitoring Helium levels on a RGA running in leak test mode. A leak is

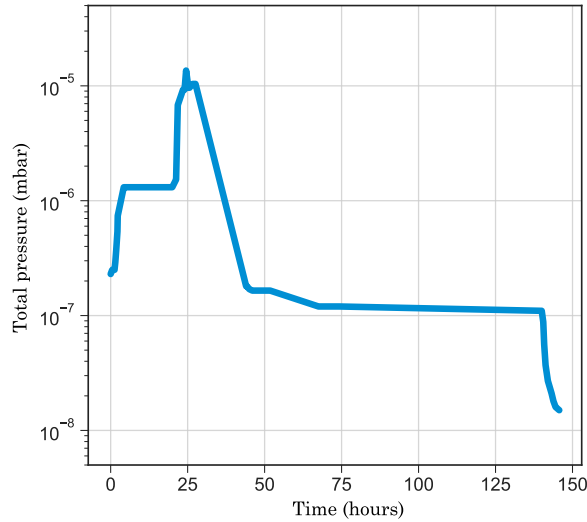


Figure 2.6: A typical baking curve showing evolution of total pressure.

indicated by increase in Helium partial pressure after it is sprayed around the leaky connection. Once a leaky seal has been detected, we re-seal it to remove the leak.

2.3 Laser setup and frequency stabilization

In order to cool and trap the atoms, the laser light is amplified and frequency-stabilized on a dedicated optical table for each atomic species. For efficient space utilization, the potassium laser table is stacked on top of the sodium one. The laser light is split into different paths for the 2D-MOT, the 3D-MOT, and the push beam etc. The frequency and intensity of these beams is controlled with the help of acousto-optic modulators (AOMs) in double-pass configuration [77, 78]. The layout of the laser systems for both species is shown in figure 2.7.

The light is transferred to the main experiment table via polarization-maintaining single-mode optical fibers. For ensuring laser safety, we have laser protection covers around both laser tables (made of 3 mm dibond plates) and the main table (made of 3 mm anodized aluminium plates). We also have a laser curtain (*Laservision BC1.FIP01.3201*) to isolate the working space at the front of the lab from the experiment.

For frequency stabilization of the lasers, we use atomic transitions as the reference. To locate the atomic transition, we can monitor the absorption of laser light passing through a thermal vapor of atoms, while we scan the laser frequency. At the resonance the absorption should be maximum. However, the thermal motion of atoms broadens the atomic absorption spectrum due to Doppler effect. To overcome this, we use saturated absorption spectroscopy (SAS)[79] as the Doppler-free spectroscopic technique. It involves using a pair of counter-propagating beams with different

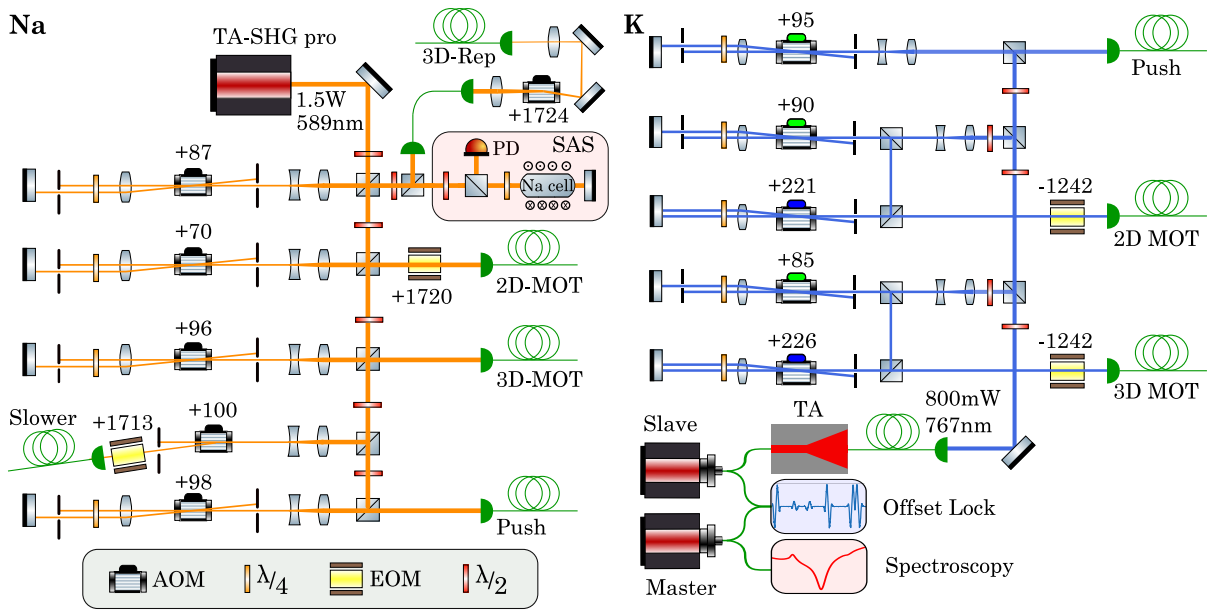


Figure 2.7: “Sketch of the optical setup for laser cooling of sodium and potassium atoms. The rf-frequencies for AOMs and EOMs are given in MHz. **Na:** The repumping light for the 2D-MOT and Zeeman slower is generated by electro-optic modulators (EOM) while for the 3D-MOT repumper, a 1.8 GHz AOM is used. **K:** In the 2D- and 3D-MOT paths the green AOM controls the ^{39}K cooling frequency and the blue AOM is responsible for the creation of the ^{39}K repumping light. The repumping light for ^{40}K is generated by EOMs.”

intensities passing through the thermal vapor. One of the beams called the pump beam is very intense while the second one called the probe beam is very weak. We monitor the probe absorption as the laser frequency is scanned.

The Doppler-Shift in laser frequency due to atomic motion is given by:

$$\omega = \omega_0 - \vec{k} \cdot \vec{v} \quad (2.6)$$

where ω and ω_0 are the apparent and original laser frequencies respectively, \vec{k} is the laser beam wave vector, and \vec{v} the velocity of an atom. When ω_0 is far detuned from an atomic transition, those atoms which see the probe beam on resonance due to equation 2.6 cannot see the pump on resonance, because it is counter-propagating to the probe. Since we only monitor the probe, we will see its Doppler-broadened absorption profile.

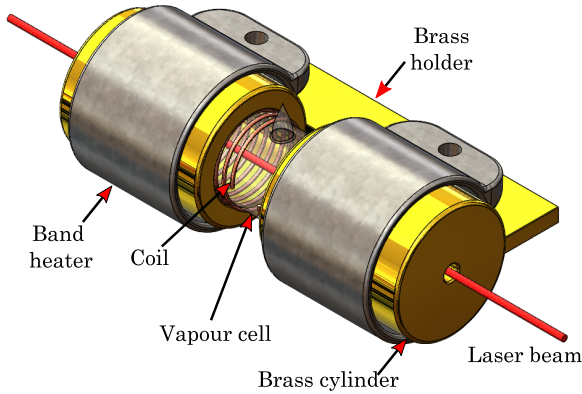
The situation is very different when ω_0 matches an atomic transition. Both pump and probe can be simultaneously absorbed by the zero velocity class of atoms (for which $\vec{k} \cdot \vec{v} = 0$). The atomic transition gets saturated by the intense pump beam, thus decreasing the probe absorption and giving rise to a peak (called Lamb peak) in the SAS signal. The simultaneous absorption of pump and probe happens only for the zero velocity class, so the sharpness of the Lamb peak is limited by the natural Lorentzian linewidth, pressure and power broadening effects, whereas it is independent of Doppler broadening.

2. Experimental setup

The hyperfine multilevel structure of atoms gives extra peaks in a SAS signal known as crossover peaks. The crossover peak of two transitions is located exactly midway between them [80]. For the case of sodium and potassium, the optical pumping of atoms amongst the hyperfine levels causes an inverted crossover peak.

The vapor pressure of potassium and sodium is not high enough at room temperature to get a SAS signal, so we have to heat their samples to 70°C (210°C) for potassium (sodium). These samples are contained in a glass cell, which we heat using band heaters (*Acim jouanin L3420C9A5*). The figure 2.8 shows the arrangement of the band heaters around the vapor cell. More details about temperature stabilization of the vapor cells are given in appendix section A.1.

A :



B :

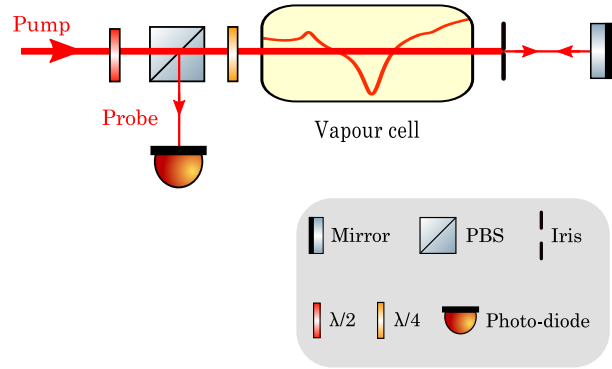


Figure 2.8: Spectroscopy setup **A:** The band heaters around the vapor cell are supported by a brass holder. A coil is wound around the vapor cell to generate the modulating magnetic field. **B:** Schematic of the SAS setup. The probe beam is derived by retro-reflecting the pump beam and partly blocking it with an iris. Figure taken from [81]

The SAS signal itself cannot be used to correct laser frequency drifting away from resonance because it is symmetric about the atomic resonance. So we need to obtain an anti-symmetric error signal from the SAS signal. For this, we use a technique called Zeeman modulation locking [82, 83] in which circularly polarized pump-probe beams and a sinusoidally modulated magnetic field are applied to the atomic vapor causing modulation of the energy levels and therefore SAS signal. This modulated SAS signal is then fed to a lock-in amplifier for demodulation and an anti-symmetric error signal is obtained. The error signal is fed into a PID module for stabilizing the laser frequency.

2.3.1 Sodium

The energy level diagram for sodium is shown in figure 2.9. “Laser cooling and trapping of sodium atoms is achieved using the D_2 -line at 589 nm, which is obtained from a high-power, frequency-doubled diode laser (*TA-SHG pro*, from *Toptica Photonics*).

The laser light is stabilized to the excited-state crossover transition of the D₂-line using SAS and Zeeman modulation locking.” To generate the modulating magnetic field, a coil is wound around the vapor cell. This coil is then driven with a sinusoidal current using a *Red Pitaya* and an audio amplifier *OPA 548*. The modulated SAS signal is fed into a digital lock-in amplifier and PI-controller, which are programmed on a *STEMLab 125-14* board from *Red Pitaya* using the *Pyrpl* module [84].

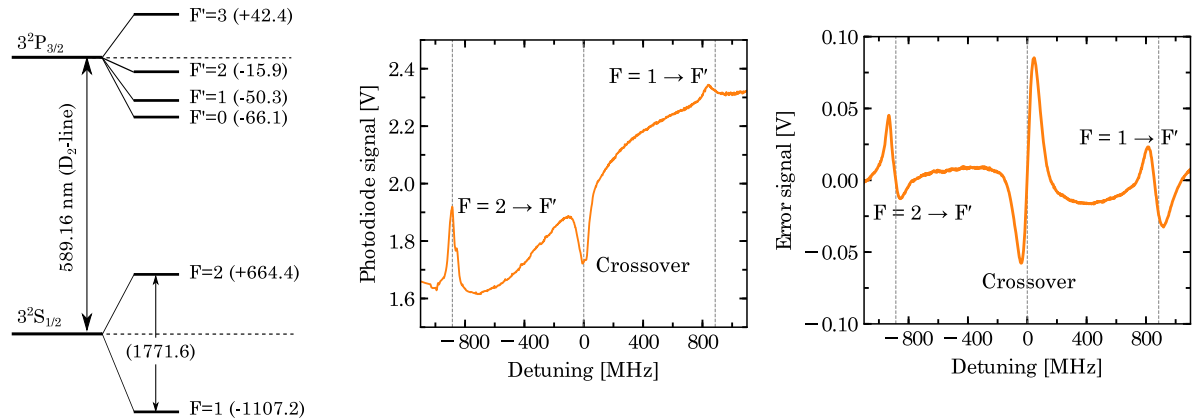


Figure 2.9: Sodium spectroscopy. **A:** Energy level diagram of sodium. **B:** The SAS signal. **C:** The corresponding error signal from Zeeman modulation. The detuning is measured from the crossover between the two hyperfine ground states. Figure taken from [81]

2.3.2 Potassium

The energy level diagram for potassium is shown in figure 2.10. “Laser cooling and trapping of potassium atoms is achieved using the D₂-line at 767 nm. The light is obtained from a master-slave laser configuration (both *DL pro*, from *Toptica Photonics*). The master laser frequency is locked to the ground-state crossover transition of the D₂-line of ^{39}K with a scheme similar to sodium. The slave laser is frequency-stabilized through an offset beat lock (-405 MHz) and its output is amplified using a home-built tapered amplifier (TA) module to a power of 800 mW (after mode cleaning with a single mode fiber).” This light is used to supply all the cooling and trapping beams. Details of the TA and the offset beat lock can be found in the Masters thesis of Jan Kilinc [81].

“The offset locking scheme is designed to facilitate switching between the two isotopes, ^{39}K and ^{40}K . To cool the fermionic ^{40}K , the slave laser frequency should be increased by approximately 810 MHz via the offset lock, with the blue acousto-optic modulators (see figure 2.7) turned off and the electro-optic modulators turned on.”

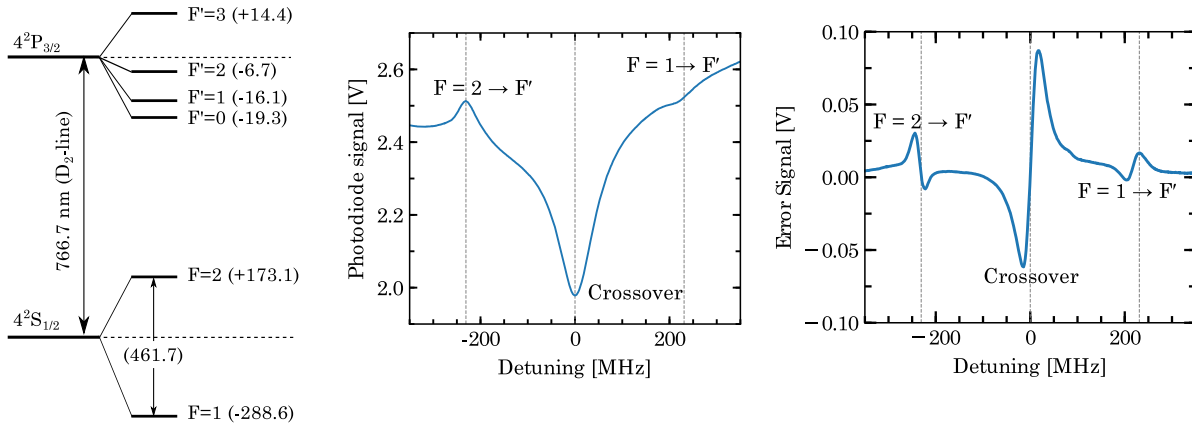


Figure 2.10: Potassium spectroscopy. **A:** Energy level diagram of potassium. **B:** The SAS signal. **C:** The corresponding error signal from Zeeman modulation. The detuning is measured from the crossover between the two hyperfine ground states. Figure taken from [81]

2.4 Laser cooling and trapping

2.4.1 Laser cooling

Laser cooling [85] uses the momentum of photons to slow down atomic motion. The Doppler shift mentioned in equation (2.6) causes the atoms to preferentially absorb photons from a red-detuned laser beam counter-propagating to atomic motion and giving a momentum kick opposite to the atom momentum. We consider two-level atoms, wherein after absorbing the photon, the atom transitions to an excited state and decays back to the ground state by spontaneous emission. This emission has a random direction and thus the associated momentum kick is also randomly oriented. Several repetitions of this absorption-emission cycle slows down the atomic motion because all absorption momentum kicks add up but emissions are random so their net momentum transfer averages to zero.

The scattering force F_{sc} produced by such a mechanism can be calculated using optical Bloch equations [86, 87]. The result is the well-known expression:

$$F_{sc} = \hbar \vec{k} \left[\frac{\Gamma}{2} \frac{\Omega^2/2}{\Delta^2 + \frac{\Omega^2}{2} + \frac{\Gamma^2}{4}} \right] \quad (2.7)$$

The above equation can be re-expressed in terms of laser intensity I and saturation intensity I_{sat} as:

$$F_{sc} = \hbar \vec{k} \left[\frac{\Gamma}{2} \frac{I/I_{sat}}{1 + I/I_{sat} + \frac{4\Delta^2}{\Gamma^2}} \right] \quad (2.8)$$

where I_{sat} is defined such that $I/I_{sat} = 2\Omega^2/\Gamma^2$. In both equations $\hbar \vec{k}$ is the photon momentum and the quantity inside the square brackets is called the photon scattering rate γ_{ph} . Other terms in these equations are: rabi frequency Ω , the laser detuning Δ and the natural linewidth Γ of the atomic transition.

So far only one laser beam was considered, which damps the motion of atoms only opposite to the beam direction. For damping velocities parallel or antiparallel to a given direction, two oppositely directed beams are required. Calculating the total contribution from both beams gives the expression for the cooling force as [88] :

$$F = -\alpha v \quad (2.9)$$

with

$$\alpha = -4\hbar k^2 \frac{I}{I_{sat}} \left[\frac{2\Delta/\Gamma}{\left(1 + I/I_{sat} + \frac{4\Delta^2}{\Gamma^2}\right)^2} \right] \quad (2.10)$$

where we assumed $\vec{k} \cdot \vec{v} \ll \Gamma$. For a damping force, we require a positive value of α and hence a red-detuning (i.e. $\Delta < 0$). To cool atoms moving in three dimensions, three mutually orthogonal pairs of counter-propagating beams along each of the X, Y, Z axes are required. Such an arrangement is called Optical Molasses [89].

The Doppler cooling mechanism discussed so far only works when the Doppler shift compensates the laser detuning from the atomic resonance. For a given atom this resonance condition is only satisfied for a specific velocity, therefore the cooling will only work in a narrow range around that velocity. To maintain the resonance condition over a wide range of velocities, one approach is to use the Zeeman effect to shift the atomic energy levels for maintaining the resonance. This can be achieved by the use of a spatially varying magnetic field. Such a device, called Zeeman slower [90, 91], is a common method to cool atomic velocities from several hundred meters per second to few tens meters per second.

2.4.2 Doppler limit and sub-Doppler cooling

The random nature of the spontaneous emission gives a random recoil to the atom after each emission. Also the atom does not always absorb the same number of photons in a given time duration. These two fluctuations cause a random walk of the atomic velocity on top of the damping. This competition between diffusion and damping culminates into a steady-state velocity for the atoms, which is the origin of the Doppler temperature limit. A detailed mathematical analysis can be done either semi-classically [92] or quantum mechanically [93, 94]. The analysis results in an expression for Doppler temperature limit as :

$$T_{min} = \frac{\hbar\Gamma}{2k_B} \quad (2.11)$$

The Doppler cooling limit in equation (2.11) holds for two-level atoms. The first experimental results [89, 95] seemed to agree with the theory prediction. However, careful experimental re-investigation [96, 97] showed that the temperatures achieved were much lower than the predicted limit. For explaining this discrepancy, new explanations [98, 99] were proposed considering the polarization gradients, multilevel structure of atoms and optical pumping among these levels. Two main scenarios were considered in these proposals regarding the polarization of counter-propagating beams

2. Experimental setup

: $\sigma^+ - \sigma^-$ polarization (corkscrew cooling) and $\text{lin} \perp \text{lin}$ polarization (Sisyphus cooling). Although the $\sigma^+ - \sigma^-$ polarization for counter-propagating beams is more common (for e.g. in a MOT), the total polarization in 3D depends on the phase factors for the laser beams. As a result of this both Sisyphus cooling and Corkscrew cooling are present to different extent [100].

The equilibrium temperature T_{OM} predicted by [99] for the sub-Doppler cooling mechanisms in optical molasses is :

$$T_{OM} \approx \frac{\hbar\Gamma}{2k_B} \frac{\Omega^2}{\Delta} \approx \frac{\hbar\Gamma}{2k_B} \frac{I}{\Delta} \quad (2.12)$$

This has been verified experimentally [101]. Sub-Doppler cooling can achieve temperatures of about few times the recoil limit which is the temperature associated with the momentum kick from emitting a single photon. Another important point is that while for most alkali atoms, molasses cooling can be observed rather easily, for few of them like potassium one needs to carefully choose the parameters [102]. This is because of the poorly resolved excited state hyperfine structure which inhibits efficient molasses cooling.

2.4.3 Magneto-Optical Trap

The cooling techniques discussed so far damp the atomic motion, but they cannot trap atoms in space. This is because the force arising from Doppler cooling is velocity dependent so it only confines the atoms in momentum space (i.e. cooling). For trapping the atoms spatially, a position dependent confining force is required. A routine technique employed for this purpose is the use of a Magneto-Optical Trap (MOT) [103, 104] which uses the multilevel structure of atoms to create a position dependent force along with a damping force.

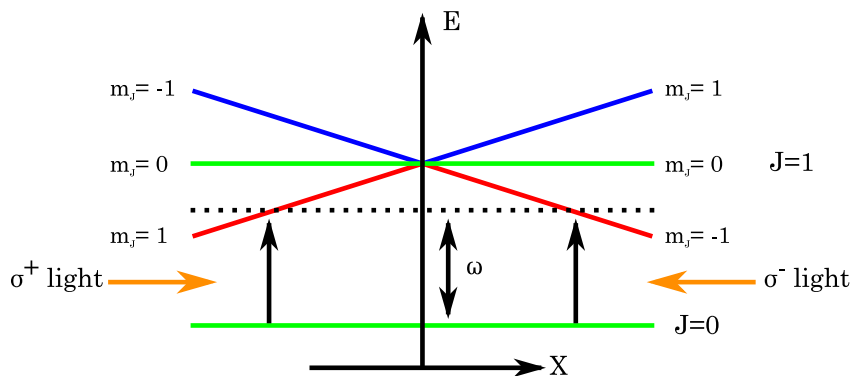


Figure 2.11: MOT working principle.

A MOT consists of the Optical Molasses arrangement (with $\sigma^+ - \sigma^-$ polarizations) in addition to a pair of anti-Helmholtz coils carrying current in opposite direction. This creates a quadrupolar magnetic field with value of zero in the middle of the common

axis of the coils and linearly increasing for small distances away from the center. For simplicity, we explain the trapping force by considering the 1D case (say along X axis) and a simple atomic transition from $J = 0$ to $J = 1$. In presence of the magnetic field B_x , the excited state $J = 1$ splits into Zeeman sublevels $m_J = 0, \pm 1$ whose degeneracy is lifted (as shown in figure 2.11) according to the relation :

$$E = g\mu B_x m_J \quad (2.13)$$

where E is the energy, g is the Lande factor and μ is the Bohr magneton. Around the MOT center, B_x and E vary linearly with position. Now consider an atom at the MOT center moving along the X-axis. As it moves towards $x > 0$, the transition $J = 0 \rightarrow J = 1, m_J = -1$ comes closer to resonance with the red-detuned σ^- light. For $x < 0$, the transition $J = 0 \rightarrow J = 1, m_J = +1$ comes closer to resonance with the red-detuned σ^+ light. So the atom is always pushed back to the MOT center by the beams. Note however, that the forces exerted by the beams have not lost the velocity dependence, rather an additional position dependence has emerged. Thus the system now resembles a damped harmonic oscillator [103]. The total force is given by [105]:

$$F_x = -\alpha v - \beta x \quad (2.14)$$

where α is same as in equation 2.10 and β is given by :

$$\beta = \frac{\alpha g\mu}{k \hbar} \frac{dB_x}{dx} \quad (2.15)$$

For continuous operation of a MOT, it is important that atoms undergo a closed cycle of transitions which provide the trapping force and cooling. Although laser frequency can be precisely tuned to particular atomic transition, the multilevel hyperfine structure of atoms has extra states to which the atom can decay. This can take the atom to a dark state where it can no longer experience the trapping and cooling. Therefore, one needs extra laser called “repumper laser” to optically pump atoms back to the cooling and trapping cycle.

2.4.4 2D-MOT

The atoms emerging from the oven form a vapor in the 2D-MOT chamber. They are captured from the vapor by cooling and trapping them in two dimensions using a combination of a quadrupole magnetic field and laser beams with circular polarization. The quadrupole magnetic field required for the 2D-MOT operation, is generated by four stacks of nine (four) neodymium bar magnets for sodium (potassium). Two retro-reflected circularly polarized laser beams with opposite handedness provide the necessary cooling and trapping forces. The pre-cooled atoms in the 2D-MOT region are transported to the 3D-MOT with a push beam. This arrangement is shown in figure 2.4. As an example, plot of magnetic field strengths in the sodium 2D-MOT setup is shown in figure 2.12.

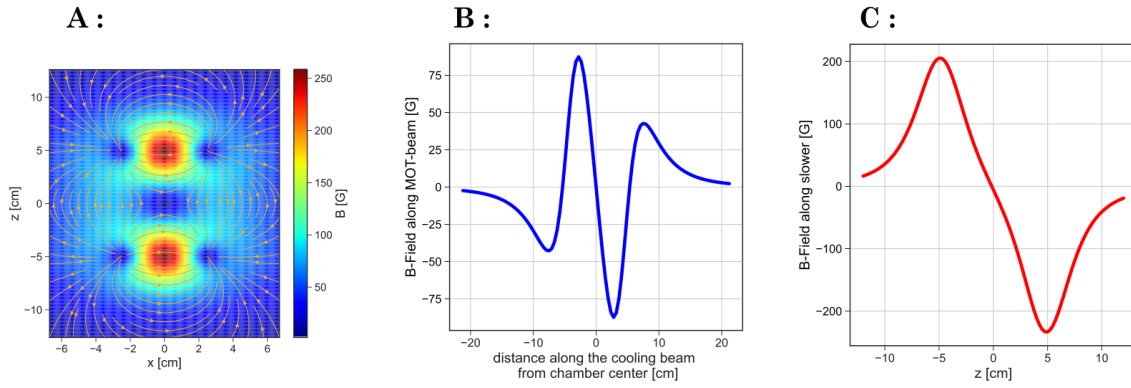


Figure 2.12: Sodium 2D MOT magnetic field. **A:** The magnetic field in the plane of the cooling beams. The z direction is along the Zeeman slower, y direction along push beam and x direction is left to right of 2D-MOT chamber. The cooling beams are along diagonal direction in this plot. **B:** Magnetic field along the cooling beams. The estimated gradient is 65 G cm^{-1} . **C:** Magnetic field along the slower beams.

The 2D-MOT acts as a low velocity intense source (LVIS) [106] for loading the 3D-MOT. It provides a narrow distribution of thermal velocities of atoms as compared to the case where the 3D-MOT is directly loaded from a vapor. This improves the capture efficiency of a 3D-MOT, resulting in faster loading rates and larger atom numbers. Such a setup has been used for the fast creation of Bose-Einstein condensates of sodium and potassium, as described in ref. [71, 107]. The 2D MOTs in our setup can produce an atomic flux of at least 10^8 atoms per second for sodium and at least 10^5 atoms per second for potassium.

2.4.5 3D-MOT

In the science chamber we operate the 3D-MOT in retro-reflected configuration for both species where atoms are cooled and trapped in three dimensions using a combination of magnetic field gradients and circularly-polarized laser beams. For this we first split the light transported (via optical fibres) from the laser table into three paths for each species on a separate breadboard and then overlap the two wavelengths on dichroic mirrors in each path. On the splitting breadboards, we adjust the circular polarization for the beams using quarter wave-plates. The correct circular polarization for a 3D-MOT is achieved when the MOT beam which travels along the MOT coil axis has opposite handedness from the other two MOT beams which have the same handedness. The overlapped laser beams are routed to the science chamber using mirrors such that they intersect inside the science chamber while forming a XYZ coordinate system. Each beam is then retro-reflected using an achromatic quarter wave-plate and plane mirror.

The quadrupole magnetic field required for the 3D-MOT is produced by a pair of coils connected in anti-Helmholtz configuration, which are placed on the sides of the science chamber. The magnetic field gradient produced by the 3D-MOT coils

is approximately $1 \text{ G cm}^{-1} \text{ A}^{-1}$. The first version of our MOT coils were made by winding 30 turns of a rectangular copper wire on an aluminum frame without any active cooling. This was upgraded later to a compact water cooled design [108] as described in section 4.2. The fast control of the current in the coils, required during an experimental sequence, is achieved through an insulated-gate bipolar transistor (IGBT) switching circuit (see appendix section A.1) and a programmable power supply. In order to cancel stray fields in the vicinity of the atomic clouds, we use three independent pairs of Helmholtz coils carrying small currents ($< 1 \text{ A}$).

With this setup, we achieved our first dual species MOT on 8 January 2020. Figure 2.13 shows a picture of our first dual species MOT taken from a side camera. The values of latest optimized parameters for laser intensities and detunings in various beam paths are given in table 2.1 for both sodium and potassium.

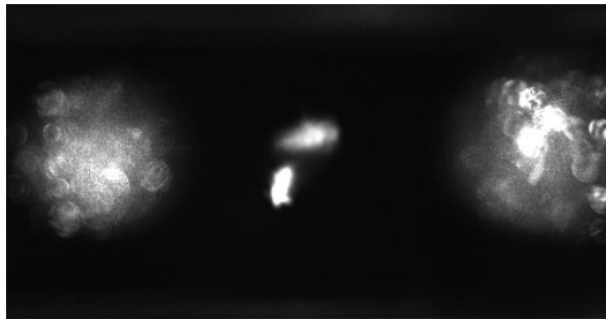


Figure 2.13: First picture of our sodium-potassium dual-species MOT.

Path	Sodium		Potassium	
	Intensity (mW cm^{-2})	Detuning (MHz)	Intensity (mW cm^{-2})	Detuning (MHz)
2D-MOT cooling	11.7	-63	5.7	-16
2D-MOT repumper	5.6	-56	4.5	-15
3D-MOT cooling	8.4	-11	1.9	-26
3D-MOT repumper	2.5	-18	1.9	-15
Zeeman slower cooling	39	-103	-	-
Zeeman slower repumper	13	-103	-	-
Push	28	-7	14	-5

Table 2.1: Table of values for powers and detunings in various paths. The intensities are reported per beam.

2.5 Fluorescence imaging

Common imaging techniques for ultracold atoms include absorption imaging [16, 18, 109], phase-contrast imaging [110, 111] and other non-destructive techniques [112–115], dark-ground imaging [116, 117] and fluorescence imaging. We use fluorescence imaging in our experiment. It is based on illuminating atoms with resonant/near-resonant light and collecting the photons scattered by atoms to form an image of the

atom cloud. An important goal for our experiment is to perform single atom imaging in optical tweezers. Although absorption imaging has been used for single atom imaging [118], fluorescence imaging is better suited for such experiments (involving dilute atomic samples) as it provides a higher signal to noise ratio. Single-atom imaging using fluorescence imaging has been demonstrated in several experiments [31–34, 119–121].

2.5.1 Imaging system

For monitoring the 2D-MOT, 3D-MOT and for some experiment sequences (like MOT atom number and temperature measurement/optimization), we use CMOS cameras (Mako-030B from Allied vision technologies). For imaging we either use an adjustable compound objective (*Thorlabs MVL25M23*) screwed on the c-mount thread of the camera or a single lens. Mako-030B also supports Power Over Ethernet (POE) which allows us to connect to the camera and power it using a single Ethernet cable. The cameras can be triggered via a 3.3 V TTL signal for accurate timings of the imaging exposures.

The main imaging setup is below the science chamber as shown in figure 2.14. It comprises an apochromatic high-resolution objective, which features a numerical aperture of 0.5 and chromatic focal correction in the wavelength range 589 – 767 nm (fabricated by *Special Optics*). The objective has been designed considering our requirements of a working distance of 22 mm, which comprises 14 mm vacuum, 6 mm glass and 2 mm air. The effective focal length of the objective is 30 mm. The objective operates in an infinity-corrected alignment such that the atoms are situated in the focal plane of the objective which collimates their fluorescence. The imaging objective is also used as a focusing objective for optical tweezers as described in section 4.7.

The structural framework for the imaging setup is provided by a vertical cage system which is mounted on stages for x-, y- and z-translation along with tip-tilt adjustment. The cage system holds the objective at the top. The fluorescence of sodium and potassium is separated by a dichroic mirror, built into the cage system. Imaging paths for both species contain a secondary lens and an additional relay telescope. This allows us to do spatial filtering with an iris in the intermediate image plane of the secondary lens.

For imaging the sodium atoms we use an sCMOS camera (*Andor ZYLA 5.5*), while for the potassium atoms we use an EMCCD camera (*NuVu HNu-512*). The sCMOS camera provides a cost-effective solution for low-noise detection of cold atoms [119] while an EMCCD camera offers sensitive detection for single atoms in optical tweezers [120]. The cameras are mounted on linear translation stages for ease of camera alignment with atom image. Additionally we use narrowline bandpass filters for both sodium (*Edmundoptics 65-162*) and potassium (*Edmundoptics 65-177*) to

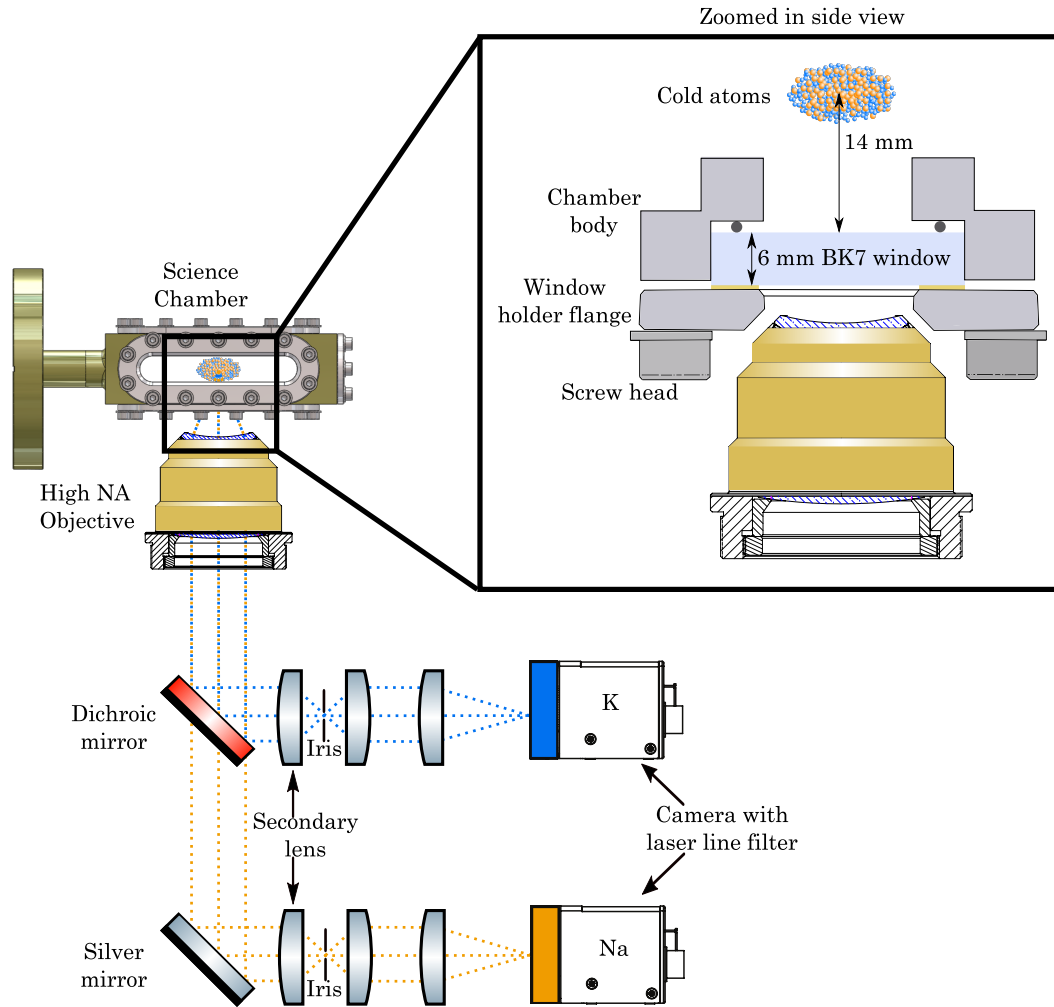


Figure 2.14: Schematic of the imaging setup. The cage system is not shown for clarity. A zoomed in view from the side shows a cut view through the science chamber and positioning of the objective relative to the chamber.

block unwanted wavelengths from falling on the respective camera sensor.

2.5.2 Extracting atom number and temperature

The average number of photons scattered by an atom when illuminated with a laser beam depends on the scattering rate γ_{ph} . It was already defined in equations 2.7 and 2.8 for one beam. For 6 beams with each beam having intensity I , the scattering rate is :

$$\gamma_{\text{ph}} = \frac{\Gamma}{2} \left(\frac{6I/I_{\text{sat}}}{1 + 6I/I_{\text{sat}} + \frac{4\Delta^2}{\Gamma^2}} \right) \quad (2.16)$$

So the average number of photons N_{ph} scattered in a given time duration τ by N_{at} atoms is :

$$N_{\text{ph}} = \gamma_{\text{ph}} \tau N_{\text{at}} \quad (2.17)$$

These photons are collected via an imaging system on a camera which converts the photon numbers into counts N_c as :

$$N_c = N_{ph} (\Omega * \eta_{det}) = N_{ph}\eta \quad (2.18)$$

where Ω is the solid angle, η_{det} is the detection efficiency taking into account the camera quantum efficiency and transmission losses in the optics, and $\Omega * \eta_{det} = \eta$ is the overall photon collection and conversion efficiency. The equations 2.16 to 2.18, can be used to estimate the atom number N_{at} from the camera counts N_c . Since some counts originate from background light sources, camera offset counts etc., a reference image is also taken which is subtracted from the atom image.

Apart from atom number, we can also estimate the temperature of atoms. For this, we perform a Time-of-Flight (TOF) measurement. In TOF measurement, the trap holding the atoms is suddenly switched off so the atom cloud expands ballistically. After a variable time delay (usually in few milli-second range), an image of the cloud is taken. The free ballistic expansion of the cloud follows the equation :

$$\sigma^2(t) = \sigma^2(0) + \left(\frac{k_B T}{m}\right) t^2 \quad (2.19)$$

where σ^2 is the width of atom cloud, t is the time-of-flight, k_B is the Boltzmann constant, T is the absolute temperature of the atomic cloud and m is the mass of atoms. Thus by performing a linear fit on σ^2 vs. t^2 we can estimate the temperature of atoms.

Note that equation 2.19 is written for the atom cloud but we work with the images. All lengths in the images are a scaled version of the real lengths in the atom cloud (by the magnification factor of the imaging lens system). So the modified equation for the images becomes:

$$\sigma_i^2(t) = \sigma_i^2(0) + \left(\frac{\alpha^2 k_B T}{m}\right) t^2 \quad (2.20)$$

where σ_i^2 is the width in images and α is the magnification factor of the imaging system. Generally σ_i^2 is obtained in $pixel^2$, so unit conversion must be done carefully.

2.6 Experiment control system

For performing experiments with ultracold atoms, we need to ensure an accurate control of different devices involved in the setup. This is achieved with an external control of the devices via either digital or analog signals. Furthermore, various signals have to be mutually synchronized for properly executing an experiment sequence. Thus for experiment control we need two kinds of hardware : those that generate the digital/analog signals and those that produce clock ticks to synchronize these signals.

Apart from the hardware requirements, there are also some software requirements. An experiment control system is a set of software packages which can interface with

various hardware for automatized execution of the experiment sequences. It should allow user-friendly description of the experiment logic/parameters and support a wide range of hardware with the possibility to add new ones. It should also offer a methodical approach for self-sufficient representation of the experiment logic, the parameters and the data obtained during the sequences such that the experiments can be stored(or analysed) in a reproducible format. Another aspect which was very important to us was that the software should be free and open-source. This allows efficient code-maintenance and prevents the tendency to re-invent the wheel by every new research group. It also easily allows others to contribute to the project thereby widening the scope of its application.

The older NaLi experiment in our group used a `MATLAB-C++` experiment control system described in [122]. Although this software framework was capable of running experiment sequences, it had some shortcomings making its use inconvenient. The framework assumes the use of a fixed frequency clock which results in long compilation times and a large file size for typical experiment sequences involving a duration of about a minute and a timing resolution of few microseconds. The experiment sequences were written in a custom built file format called Sequence Definition File Format (SDFF). This file was not nicely integrated with the workflow with which data (mainly camera images), acquired from the experiment was stored. This is not a self sufficient representation of data and can be a source of confusion.

With all the understanding and motivation, for our SoPa experiment we decided to use the *labscript suite* [123] for experiment control. *Labscript suite* is a free and open-source ¹ python based modular experiment control framework. It is becoming an increasingly popular choice in the ultracold atom community with a very responsive forum support². It is described in detail in the PhD theses of Phillip Starkey and Chris Billington [124, 125]. Here I will summarize the *labscript* architecture and then describe our hardware and software configuration and our workflow for daily use in the lab.

The framework of *labscript* assumes a hierarchy of devices in a parent-child relation. This hierarchy is a representation of how devices are connected to each other. In its simplest form it comprises two levels. The top level consists of a master *psuedoclock* as a parent device. It is a variable frequency clock which produces clocking signals only when the child devices need to update their inputs/outputs. This saves memory by avoiding redundant information. The next level includes various digital and analog devices for generating/acquiring signals. All these are children of the master *psuedoclock* and must be connected to it for receiving the clock signal. These child devices can then control other hardware in the lab.

The type of experiments *labscript* addresses are shot-based. A shot is a sequence of

¹<https://github.com/labscript-suite>

²<https://groups.google.com/g/labscriptsuite>

2. Experimental setup

instructions to be executed with precise timings on the devices. For building a shot, *labscript* uses two main python files : one for specifying the sequence of instructions or the experiment logic (called the *labscript file*) and the other for description of the connections in the lab (called the *connection table*). Each shot is saved in HDF5 file format with the file containing a comprehensive amount of information pertaining to the shot.

Labscript suite is divided into five components, each of which performs specific tasks as described below:

- *labscript* : *labscript* is a compiler which comes with an Application Programming interface (API). The API provides a set of functions to interact with the *labscript* compiler. These functions are called within the *labscript file* to define the experiment logic. In the *labscript file* the hardware devices are represented as Python objects and the API functions are class-methods available to these objects. During compilation, the compiler converts the *labscript file* python script into low-level instructions specific to each hardware and also generates timing instructions for the psuedoclock.
- *Runmanager* : Runmanager is responsible for the compilation of shots into HDF files and managing related information. It also manages the parameters used in the experiment logic. These parameters called globals can be any valid python expression. In case some parameters are arrays or lists, runmanager creates shots for each possible combination of parameters by performing a Cartesian product.
- *BLACS* : BLACS is responsible for executing the low level instructions on the hardware according to the HDF file for each shot. BLACS operates in two modes : manual mode in which the user can control the hardware and buffered mode in which shots are executed under hardware timing. In its GUI BLACS dynamically generates a tab for each device according to the *connection table*. For details see [124].
- *Lyse* : After a shot has been executed, the data in its HDF file can be analysed using *lyse*. There are two types of *Lyse* analysis routines : single- shot and multi-shot. Single-shot routines are run for each shot independently while multi-shot routines access data in several shots. e.g. MOT loading rate . For all loaded shots *Lyse* produces a pandas dataframe with the details of all parameters, analysis results etc.
- *Runviewer* : Runviewer is used for viewing graphically, the expected changes in each output across one or more shots. Its use is optional, but can be useful for debugging the behaviour of experiment logic. The output traces are generated directly from the set of hardware instructions stored in a given HDF file. This provides a faithful representation of what the hardware will actually do.

The hardware for our experimental control is purchased from Spincore and National Instruments (NI). We chose the following hardware configuration as it was already compatible with *labscript*. To use the following hardware with *labscript* we need to use the device drivers supplied by the respective vendor.

- Spincore pulseblaster USB : It is a multichannel pulse/delay generator and is used as the master *pseudoclock*. It is capable of generating pulses and delays ranging from 50 ns to 4.5×10^7 s in length when operating with a 100 MHz clock frequency. The clock oscillator signal originates from an on-chip PLL circuit with a base frequency of 50 MHz. The PulseBlasterUSB has up to 24 independent output channels with 3.3 V TTL signals, and can accommodate pulse programs containing up to 4096 instructions. The clock signal from a output channel of pulseblaster can be received by an NI card using its Programmable Function Input (PFI) channel.
- NI PXIe chassis 1073 : It provides a rugged enclosure for upto five NI cards. The integrated controller built into the chassis can connect to a computer using a PCI connector card NI PCI-8361, which plugs in to the motherboard of the computer. This allows a convenient control of all the NI cards plugged in the chassis.
- NI PXIe 6535: It is a digital output card having 32 channels with 3.3 V TTL signals. The on-board memory can store upto 2048 samples.
- NI PXIe 6738: It is an analog output card having 32 analog channels with ± 10 V output range and a maximum update rate of 1 million samples per second. It also has 10 digital input/output channels. The on-board memory can store upto 65, 535 samples.

The spin core pulseblaster USB comes in a 19-inch rack mountable enclosure with all its connections exposed on BNC outputs. For the digital and analog cards, we made our own breakout boxes. The cable attached to PXIe 6535 has 68-pin VHDCI connector on both sides. On one side, we expose these pin connections using the breakout PCB *BRKAVH68FV1-R* from *Winford*. The PCB offers both screw terminals and solder pads. The exposed pins on the PCB are then connected to a PVC panel which has an array of BNC connectors. Finally using an aluminium blase plate and PVC plates for other sides we assemble our breakout box. The same procedure is used for the analog card PXIe 6738, but since it has a 68-pin SCSI connector cable, we use a different breakout PCB *BRKBDP68FV1* from *Winford*.

Labscript supports a wide variety of hardware and offers the possibility for including new hardware. At the beginning of our experiment, we purchased hardware which was already supported. As the experiment progressed, we started buying new hardware which was not supported in *labscript*. So we had to write custom code to

2. Experimental setup

use those devices. For some of the use cases we could also implement easier work-around by writing lesser code. Given below are three such use cases with different devices:

- *MaKo camera* : We integrated Mako camera into *labscript* by writing the corresponding *BLACS* worker class and other helping classes for it. For this we could use a lot of functionality of the existing IMAQDX camera class in *labscript*. We also needed to use the *pymba* package, which is a python wrapper around the official C-API called Vimba provided by *Allied Vision technologies*. After this integration, Mako can now be used as a regular device in *labscript*. Also since the Mako class uses *pymba* which supports other cameras from *Allied Vision*, it should be possible to use them as well with *labscript*. The code for Mako integration into *labscript* is publicly available at : https://github.com/fretchen/synqs_devices
- *Andor ZYLA 5.5 camera* : Although *labscript* has support for some camera models from ANDOR, there were some glitches while using our camera model Andor ZYLA 5.5 as a BLACS device with *labscript*. To resolve this we came up with a work-around which is very general purpose. It uses an existing dummy device in *labscript* called `function_runner`. This device can execute arbitrary python code at the beginning or end of each shot. So we just trigger the camera using NI PXIe 6535 and program the camera settings using a free software called Micromanager. Micromanager supports Andor ZYLA 5.5 and uses ANDOR SDK to program the camera. The camera is programmed to take a fixed number of images for fixed number of triggers as required by the number of shots to be run. Whenever the camera takes an image during a shot by receiving a TTL trigger it saves them in a predefined folder. At the end of that shot the `function_runner` device packs all images from the storage location into the HDF file for that shot. This way we can easily use any camera for which the support is not yet in *labscript*. An example of this is our NuVu camera, for which also there is no existing support in *labscript*, but we can still use it with the above method.
- *SynthHD* : This is a microwave generator which can generate any frequency between 10MHz and 15GHz in 0.1Hz resolution. We use it for microwave frequency sweep for performing evaporation in the magnetic trap. The vendor Windfreak provides a GUI to control the SynthHD. At the beginning of each scan with several shots, we set all parameters for the frequency sweeps like frequency, amplitude, sweeping duration and range etc. and we set the SynthHD in trigger mode. During a shot whenever SynthHD receives a TTL trigger from the NI PXIe 6535, it will perform the microwave sweep. Note that we can also achieve a similar functionality using the python API provided by Windfreak and using `function_runner` to set the parameters for SynthHD at the beginning of each shot. This method can have the added advantage that one can perform scans

of sweep parameters in *Runmanger* and use those variables automatically for setting the SynthHD via `function_runner`.

For our use in the lab, we install *labscript* in a python virtual environment using miniconda. For shot storage, we use a network storage service (called Scientific Data Storage or SDS) provided by University of Heidelberg. This allows us to run analysis on the data from other computers on the network. For better code maintenance of our experiment logic, we define various functions corresponding to different actions (e.g. load MOT, perform molasses, take image etc.) in a separate file. These functions are then imported into the main experiment logic file. This file containing all the functions is also saved in the HDF file for each shot.

For analysis, we use *Lyse* without its GUI. For analysing a particular scan folder with several shots, we import the *lyse* library in a Jupyter notebook and use the function `dataframe_utilities.get_dataframe_from_shots()` to get a pandas dataframe with details of all parameters for all shots. With this dataframe its very easy to perform analysis on the shots for e.g. conditionally selecting those shots which fulfill some criteria for values of parameters etc. Our analysis functions are written in a separate python file in a class structure. In the Jupyter notebook we instantiate an object of this class and then call the desired analysis functions. A framework of our analysis code is given in appendix section [A.2](#).

2.6.1 Is *labscript* enough?

A very powerful feature of *labscript* is that the different components of *labscript* have an API which can be used to communicate with them. Let us consider *Runmanger* for example. Any running instance of *Runmanger* can be communicated using its remote API. This API provides functions with which one can perform the same tasks as with *Runmanger* GUI e.g. specifying the *labscript file* or the shot output folder, setting values of globals, starting compilation of shots etc. This ability is very helpful if one wants to incorporate *labscript* as a part of a bigger software framework, because other software components can communicate with *labscript* components using their respective APIs.

The obvious next question is why would one want to include *labscript* into a larger framework. The exact details are explained in [Chapter 7](#), but suffice it to say that for those use cases where several remote users could submit “instructions” to an experiment, *labscript* does not provide all requirements. Although it supports a distributed web architecture where different modules of *labscript* running on different computers can communicate via a network, this is limited to a local network. For communicating over the internet for several remote users, the following challenges exist:

- There is no built-in user management in *labscript* although workarounds can be

used to some extent.

- A robust user authentication is also missing.
- Transferring the HDF shot files over the internet is not so easy. So a different method has to be used.
- Given that *labscript* allows the remote users to compile HDF files from python code, there is also a security risk of arbitrary code execution.
- Not every user might be willing to adapt to learning *labscript*. So we have to create a solution based on something that is already very popular amongst remote users.

These reasons (among others) lead us to developing *qlue*³, which is a backend agnostic web-interface allowing users to access remote cold atom platforms in a secure way. The details about *qlue* are mentioned in [Chapter 7](#).

³<https://github.com/synqs/qlue>

Atom number dynamics in a MOT

As explained in [Chapter 2](#), we have set up a new Na-K cold atom mixture experiment. After we got our first dual-species 3D-MOT, we wanted to perform experiments with our setup, which would not only allow us to benchmark it, but also show the potential of our machine for fundamental physics studies. We concluded that atom counting in a MOT is a very good candidate for this purpose. In this chapter, I describe our atom counting experiments and the use of atom counting for quantifying the effect (or absence) of inter-species interaction. This chapter closely follows our publication [66]. The parts of this chapter taken verbatim from [66] are put in “quotes”.

3.1 Atom counting in a MOT

In a MOT, the trapped atoms go through a cyclic cooling process and continuously emit fluorescence which can be used to image them as explained in [section 2.5](#). For atom counting in a MOT, we monitor this fluorescence and try to detect changes in the fluorescence signal due to load or loss of a single atom in the MOT region. The experimental feasibility of this technique to count single atoms was first demonstrated in [126]. In our imaging setup, the light collected from both atomic species with our imaging objective is routed to separate cameras. To record the atom fluorescence, we expose the camera for a given time interval in which the fluorescence photons impinging on the camera sensor create pixel counts in the region of the sensor, where the image of the MOT is formed. Since the motion of the atoms during the exposure time washes out any spatial information, we sum up the pixel counts over the entire MOT region. Taking several images, and performing the summation for each image results in a time trace of camera pixel counts. We then try to measure jumps in this time trace due to change of atom number by one.

3.1.1 Why do atom counting in a MOT?

Single particle resolution is a requirement for numerous experimental protocols [127]. Here, I list few examples where atom counting in a MOT has been used as a sensitive technique for accurate determination of particle number.

- Metrology : Atom interferometry experiments aim for accurate measurement of a physical quantity by encoding it into the relative phase between two modes of an atom interferometer [128]. The relative phase is estimated by combining the two modes and measuring their population. Accurate particle number detection is thus directly linked to the accuracy of phase estimation. Although these interferometric applications use Bose-Einstein condensates, atom counting in a MOT can still be helpful by using it at the end for only detection purposes. This is done by recapturing atoms into a MOT to perform state resolved population readout [129, 130].
- Fermionic systems : Deterministic preparation of few fermion systems [131] has attracted a lot of research interest in the last decade. This bottom-up approach paved the way for several studies [132, 133] of the basic properties of fermionic quantum systems starting from the few-body limit and building towards the collective states seen with many particles. Accurate detection of particle number was a crucial part of these experiments and was accomplished using atom counting in a MOT.
- Inter-species cold collisions : Dual-species MOTs exhibit inter-species collisions which can lead to dramatic losses compared to the single species case. A sensitive probe for measuring these inter-species interaction processes can be obtained by operating one species as a tracer (in few atom regime) in the MOT while maintaining a large cloud of the other species. The technique of accurate atom counting in a MOT can precisely measure the effect of inter-species interactions on the atom number dynamics of the tracer species. This method has been used to extract a precise value of the inelastic inter-species collision coefficient in a rubidium-caesium dual-species MOT [134].
- Interdisciplinary research : Accurate atom counting in a MOT has been used in studies of environmental systems by measuring the concentration of tracer atoms in a system [135]. This technique called atom trap trace analysis (ATTA) [136] provides a robust and sensitive [137] way to determine tracer concentration in a given sample. By measuring the tracer concentration, one can estimate the age of the sample (usually water) which in turn allows to infer information on environmental and anthropogenic processes, such as groundwater flow dynamics, ocean ventilation and climate change [138].

The versatility of single atom counting in a MOT for accurate particle number detection has led to its implementation by several research groups [130, 131, 134, 136, 139–144]. For us, the main motivation was using the technique to benchmark our setup in terms of continuous operation, imaging sensitivity etc. Furthermore we wanted to use this technique for studying inter-species collision properties on our setup.

3.1.2 Controlling atom number in a MOT

A key requirement for accurate atom counting is to ensure, that during a camera exposure, there should be no load or loss event because that would smear out the fluorescence signal level. For this reason, we need to drastically reduce the vapor pressure of atoms around the MOT region. In order to achieve this in our setup, we turned off the oven heating and the 2D MOT beams.

The MOT is loaded from a very dilute vapor in the science cell. The loading rate for such a vapor-loaded MOT is given by [104] :

$$\Gamma_{load} = 0.5nV^{\frac{2}{3}}v_c^4 \left(\frac{m}{2k_B T} \right)^{\frac{3}{2}} \quad (3.1)$$

where n is the density of atoms in the vapor, m is the mass of atoms, T is the absolute temperature, v_c is the capture velocity, k_B is the Boltzmann constant and V is the trapping volume. The loading rate can be reduced by reducing the size [145] of the MOT beams (and hence the trapping volume V) and increasing the magnetic field gradient [139, 140]. We decreased the beam size to about 2 mm diameter while the maximum magnetic field gradient was limited to 21 G cm⁻¹ by the heating of our MOT coils. As a consequence of these steps, the loading rate could be reduced to few atoms per minute. Due to the UHV in the science cell, the loss rate also got reduced by a suppression in the amount of background collisions.

3.1.3 Imaging noise

The fluorescence signal of a single atom in a MOT differs from one image to the other because of the noise sources in the imaging setup. The list below summarizes the major sources of noise and how they can be suppressed :

- **Photon shot noise** : It refers to the fluctuation in the number of photons emitted by an atom in a given time duration τ due to discrete nature of the process. The average number of photons emitted by N_{at} atoms in detection time τ is given by equation 2.17. The fluctuations in fluorescence arising from the shot noise is characterized by a variance of N_{at}/N_{ph} in units of atom number. So the shot noise decreases with increasing τ , however τ cannot be increased indefinitely due to finite lifetime of atoms in the MOT and the loading of atoms into the MOT, so an optimum exposure time has to be determined [143].
- **Intensity and polarization fluctuation of the MOT beams** : As shown in equation 2.17 the average number of photons also depends on the scattering rate R_{sc} expressed in equation 2.16. The scattering rate R_{sc} depends on the intensity of the MOT beams and their detuning. Since the laser is frequency stabilized the detuning is fixed, but for a stable scattering rate the intensity of the MOT beams should also be stabilized. Furthermore, this intensity is considered

for a given polarization. Although the MOT light is transferred from the laser table to the optical table using polarization maintaining fibers, we still need to clean the polarization after the fiber using a PBS. This polarization cleaned light is then actively intensity stabilized using a PID loop.

- **Stray light noise on the camera sensor** : The stray light falling on the camera can suppress the signal of a single atom. Although for fluorescence detection we subtract a background signal from each image, but fluctuations in stray light on the camera can cause an imperfect subtraction thus leading to noise. For this purpose we try to reduce the stray light noise by using bandpass filters in front of the camera (for filtering other wavelengths) and use spatial filtering (see figure 2.14) in the imaging system to reduce the stray light from the MOT beams on the camera. This allows us to keep the MOT image in a relatively dark region of the camera sensor with low background light.

Apart from these steps we also had to optimize the MOT alignment, frequencies of the MOT beams and offset magnetic fields to get single atom counting sensitivity. After implementing these measures we could achieve single atom counting for sodium but not for potassium. This is because the temperature of a potassium MOT is much higher than sodium [146]. This did not allow us to decrease the size of the potassium MOT. Thus the image of the MOT on the camera was not well separated from stray light from the MOT beams leading to poor detection quality. So we decided to work in few atom regime for sodium and high atom number regime for potassium.

3.2 Atom number dynamics and statistical analysis

In this section, I describe the experiment we used for tracing the atom number dynamics in the 3D-MOT for both species. We study them separately with different atom number regimes but analyse their evolution with a common statistical approach. This enables us to separate the fluctuations induced by the statistical loading process from those caused by technical limitations because the two types of fluctuations show very different behaviour in the data. Our experimental sequence is shown in figure 3.1, where the timings are controlled with the *labscript suite*.

During a sequence, the 3D-MOT beams are always ON. “We start the atom dynamics by switching on the MOT magnetic field (with a gradient of 21 G cm^{-1}) and then monitor the fluorescence in $N_{\text{img}} = 200$ images. Each image has an integration time τ , such that the atom fluorescence overcomes the background noise. We sum the camera counts over the entire MOT region to get a single number N_c for each image which results in a time trace of N_c . Each experimental run contains a series of 100 reference images at the beginning and at the end to quantify the background noise Δ_{bg} , induced by the fluctuations in the stray light on the camera from the MOT beams.”

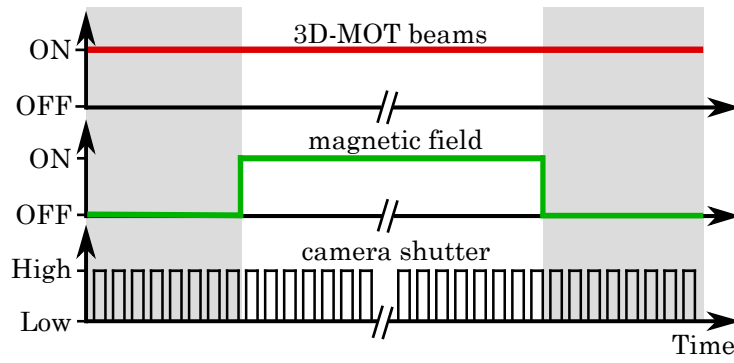


Figure 3.1: “Experimental sequence. A series of images (black) is taken. While the MOT beams (red) are always on, the magnetic field (green) is switched off for reference images marked in grey.”

For both species we repeat the sequence to get 100 time traces. As an example, five time traces for each species are shown in figure 3.2.

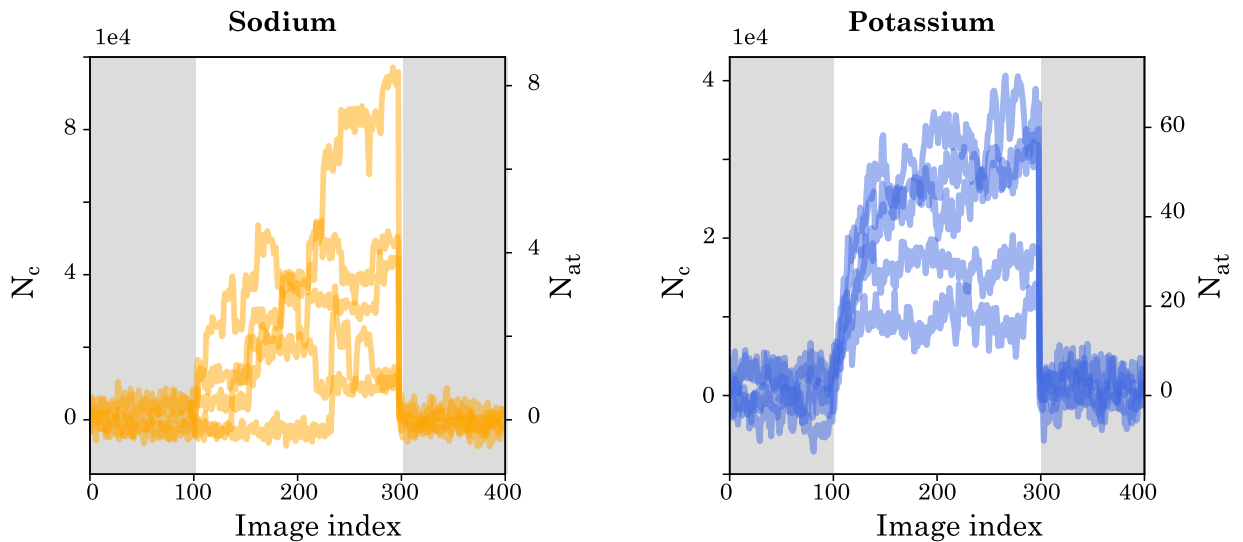


Figure 3.2: Five time traces of sodium and potassium showing evolution of total camera counts. The right axis shows atom number obtained using the calibration process explained later.

3.2.1 Sodium

For sodium we use an integration time $\tau = 200$ ms. We turn off the 2D-MOT completely and load the 3D-MOT from background gas in the science chamber. “The camera counts for sodium exhibit random jumps, corresponding to single atom load and loss events. The dynamics is extremely slow and never reaches a stationary state during a time trace. The stochastic nature of the observed signal and large relative fluctuations require a statistical analysis of the dynamics in terms of expectation values.”

We also aggregate the camera counts of all 100 time traces into one histogram, which shows distinct atom number peaks as shown in figure 3.3. By performing Gaussian

fits to individual single atom peaks in this histogram, we can extract two quantities: the calibration factor C_{Na} for converting camera counts to atom counts and the background noise limit $\Delta_{\text{bg,Na}}$. “The distance between consecutive peaks corresponds to the calibration factor $C_{\text{Na}} = 1.15(5) \times 10^4$ for $\tau = 200$ ms and the width of the zero atom signal sets the background noise limit to $\Delta_{\text{bg,Na}} = 2201(2)$ counts. From the overlap of the peaks, we estimate the detection fidelity of atoms to 96(3)%.”

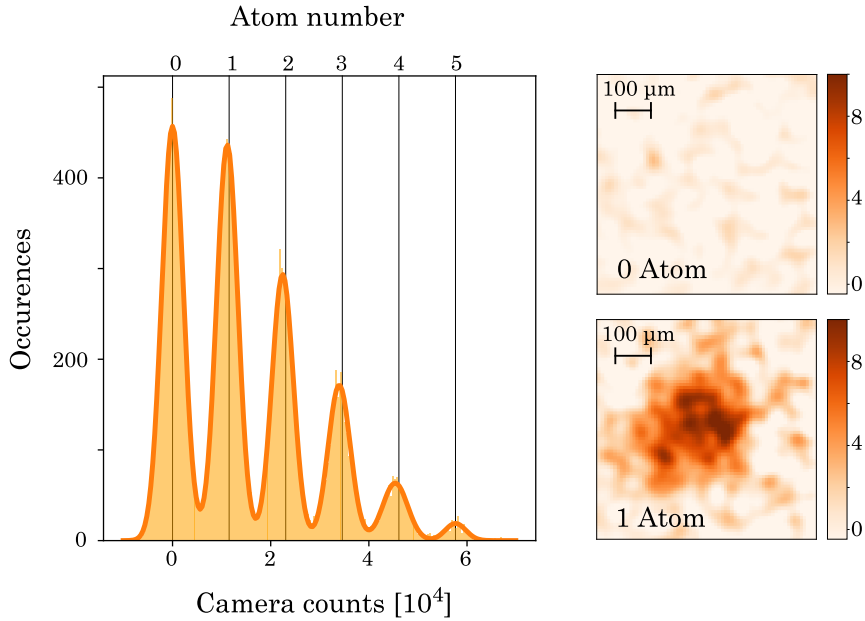


Figure 3.3: “Accurate atom counting of sodium. Histogram of recorded camera counts. The calibration from camera counts to atom number is accomplished through Gaussian fits to distinct single atom peaks. Side images show average images of zero and one atom.”

3.2.2 Potassium

For potassium we use an integration time $\tau = 100$ ms. The 3D-MOT is loaded with upto 60 atoms using very low power in the 2D-MOT beams. “In contrast to sodium, we do not observe discrete jumps, but rather a continuous loading curve with higher counts and smaller relative fluctuations. The stationary situation is achieved on average after a few seconds of loading. These are typical features of a bath, which can be characterized by its mean and variance.”

3.2.3 Unified statistical analysis

The atom number dynamics for both species show fluctuations across the 100 time traces recorded for each one of them. “We study these fluctuations by extracting expectation values through an ensemble average. We calculate the mean \overline{N}_c and the variance Δ_c^2 of camera counts at each image index of the time traces and plot Δ_c^2 as a function of \overline{N}_c in figure 3.4. For both sodium and potassium we observe a strong dependence of Δ_c^2 on \overline{N}_c .”

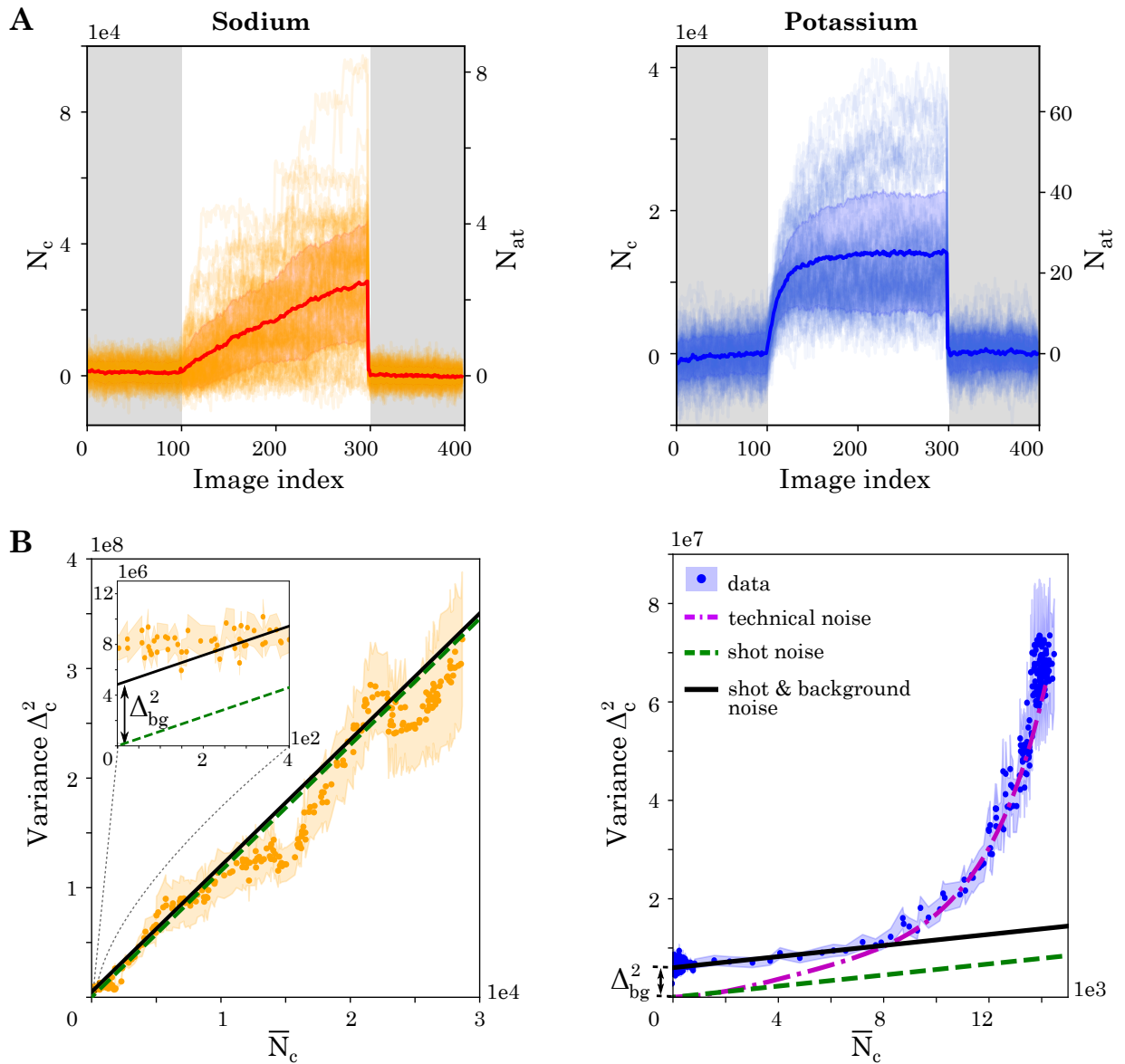


Figure 3.4: “Characterization of atom number fluctuations for sodium (left) and potassium (right). **A**: Hundred time traces of sodium and potassium with mean and error band (shown as thick lines with shaded region around them). **B**: Dependence of variance on mean camera counts. For sodium (left) the inset shows the background noise level.”

“For sodium, the variance shows a linear dependence on the average counts with an intercept. This behavior can be understood by considering two independent noise sources. The first one is a background noise Δ_{bg} , which is independent of the atom number and adds a constant offset to the variance. It originates from the readout noise of the camera and intensity-varying stray light. The second noise source is the atom shot noise, which describes the random variations due to the counting of atoms loaded until a given image index in the time trace. Its variance is equal to the average atom number. The recorded camera signal is directly proportional to the atom number $N_c = C N_{at}$, leading through error propagation to a variance of $C \bar{N}_c$. The two independent noise sources add in their variances”

$$\Delta_c^2 = C \bar{N}_c + \Delta_{bg}^2. \quad (3.2)$$

“This theoretical prediction agrees well with the experimental observations. The calibration constant $C_{\text{Na}} = 1.15(5) \times 10^4$ and the background noise $\Delta_{\text{bg,Na}} = 2201(2)$ were independently extracted from the histogram in figure 3.3. This validates our assumption that background and shot noise are the dominating noise sources for sodium. Converting the camera counts back into atom numbers, we obtain a resolution of $0.20(1)$ atoms, quantifying the quality of the observed single atom resolution.”

“For potassium, we observe a more complex behavior of the variance. In the regime of few counts, the variance is again dominated by the background noise and the atom shot noise. With the noise model in equation (3.2) validated for sodium, we perform a fit to extract the calibration factor $C_{\text{K}} = 560(140)$ and the background noise $\Delta_{\text{bg,K}} = 2450(140)$. The resulting atom resolution of $4.3(1.1)$ atoms is similar to that achieved in precision experiments with Bose-Einstein condensates [147, 148].”

“For higher atom numbers, we observe a non-linear dependence, which we attribute to technical fluctuations of the MOT. The MOT properties can be parameterized by the loading rate Γ_{load} and loss rate Γ_{loss} . Considering single atom load and loss only, they are connected to the atom number dynamics through:”

$$N_{\text{at}}(t) = \frac{\Gamma_{\text{load}}}{\Gamma_{\text{loss}}} [1 - \exp(-\Gamma_{\text{loss}}t)]. \quad (3.3)$$

“We fit each time trace with this solution and extract the distribution of Γ_{load} and Γ_{loss} across different runs. Then we regenerate smooth curves using equation 3.3 for each value of Γ_{load} and Γ_{loss} and perform variance vs mean analysis on these traces too. The variance in the atom number dynamics, resulting from these fluctuations, is traced as the dash-dotted curve in figure 3.4 B. In the high atom number regime it agrees well with our experimental observation.”

3.3 Simulation of the atom dynamics

In section 3.2.3, I described how we used the same technique to extract the calibration factor for both sodium and potassium. It is easy to see that the approach works correctly for sodium, but it is not obvious if the approach is also applicable to potassium, because we do not have single atom counting resolution for potassium. In this section, I present the simulation of the atom dynamics, which shows that our method allows to estimate the calibration factor, even when single atom counting resolution is not obtained. However, the method is sensitive to technical fluctuations of Γ_{load} and Γ_{loss} and will have a larger error for higher fluctuations in these parameters.

“We model the microscopic dynamics of the atom counts $\tilde{N}_{\text{at}}(t)$ by a statistical process whose macroscopic parameters are the loading rate Γ_{load} and loss rate Γ_{loss} . For short enough time steps dt the loading of a single atom is described by a Bernoulli trial $\mathcal{B}[1, p_{\text{load}}]$, where $p_{\text{load}} = \Gamma_{\text{load}}dt$. The loss of a single atom is described by a binomial

process $\mathcal{B}[\tilde{N}_{\text{at}}(t-dt), p_{\text{loss}} = \Gamma_{\text{loss}}dt]$. The change of atom number at time t is then given by:”

$$\Delta\tilde{N}(t) = \mathcal{B}[1, p_{\text{load}}] - \mathcal{B}[\tilde{N}_{\text{at}}(t-dt), p_{\text{loss}}] \quad (3.4)$$

“Additionally, interactions between atom pairs can be modelled microscopically by another binomial process $\mathcal{B}[\tilde{N}_{\text{pairs}}(t-dt), p_2]$, where $\tilde{N}_{\text{pairs}} = \frac{(\tilde{N}_{\text{at}}-1)\tilde{N}_{\text{at}}}{2}$ and p_2 is the loss probability of a pair. However, we neglected this process in our analysis, as all our observations are compatible with $p_2 \approx 0$.”

“The atom number at time $t = k \cdot dt$ is then given by:”

$$\tilde{N}_{\text{at}}(t) = \sum_{k=1}^{t/dt} \Delta\tilde{N}(k \cdot dt), \quad (3.5)$$

“with $\tilde{N}_{\text{at}}(0) = 0$ and $k \in \mathbb{N}$. From (3.5) we arrive at the collected fluorescence signal through:”

$$\tilde{N}_{\text{ph}}(t) = \eta \gamma_{\text{ph}} dt \tilde{N}_{\text{at}}(t), \quad (3.6)$$

“where η is the overall photon collection and conversion efficiency, and γ_{ph} the photon scattering rate.”

“While imaging the atoms in a magneto-optical trap (MOT), the atom fluorescence signal is integrated on a camera over an exposure time τ as visualized in Fig. 3.5 A.”

“The camera counts $N_c(m)$ of image $m \in \{0, 1, 2, \dots\}$ are then expressed as”

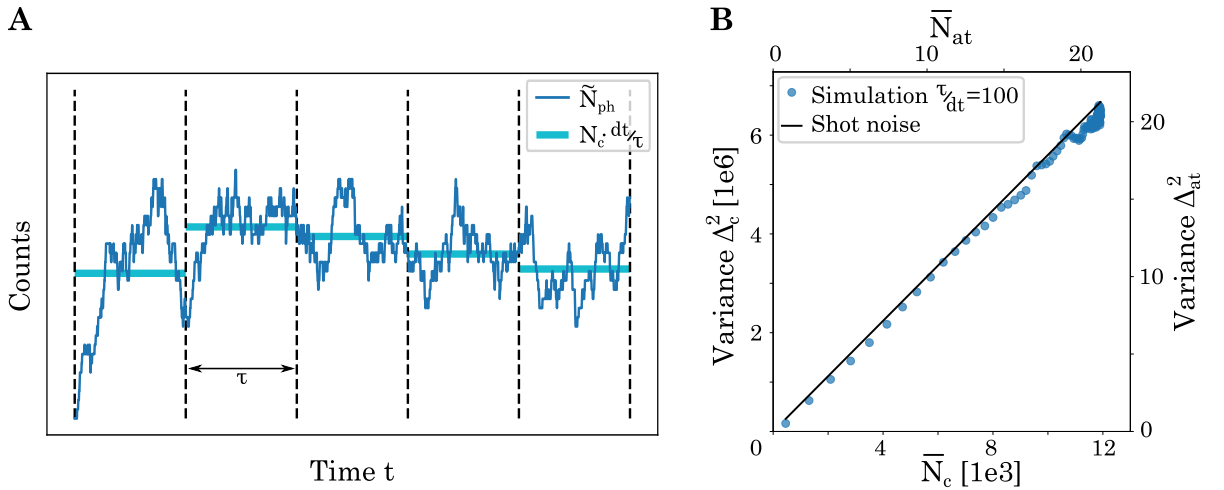


Figure 3.5: “Simulation of the imaging statistics. **A:** The loading and loss processes are characterized by the loading rate Γ_{load} and loss rate Γ_{loss} respectively, while the imaging happens over an exposure time τ . **B:** Dependence of variance on mean camera counts for an ensemble of simulated time traces. The shot noise line has a slope of $C = \eta\gamma_{\text{ph}}\tau$.”

$$N_c(m) = \sum_{k=1}^{\tau/dt} \tilde{N}_{\text{ph}}(m \cdot \tau + k \cdot dt) \quad (3.7)$$

$$= C N_{\text{at}}(m) \quad (3.8)$$

“The terms of equation (3.8) are the calibration factor $C = \eta\gamma_{\text{ph}}\tau$ and the observed atom number N_{at} from section 3.2.3. By repeating the sum in equation (3.7) for each image m we can simulate the experimental time traces. A typical result of the method is shown in Fig. 3.5 A.”

“The single atom counting regime is characterized by average atom number changes $\langle\Delta\tilde{N}\rangle \ll 1$ during the exposure time τ . We can then set $dt = \tau$ and experimentally observe the microscopic dynamics as $N_{\text{at}}(m) = \tilde{N}_{\text{at}}(m \cdot \tau)$. This is the experimentally realized case for sodium.”

“In the case of large atom numbers and fast dynamics, as observed for potassium in section 3.2.3, we observe a linear dependence of variance of camera counts on mean camera counts with the slope C . The extracted C from a linear fit agrees well with its value used for generating the time traces (i.e. $C = \eta\gamma_{\text{ph}}\tau$) as shown in Fig. 3.5 B. This allows us to estimate the calibration factor reliably as discussed in section 3.2.3.”

3.4 Immersing sodium in potassium

“The random evolution of a small system in a large bath can only be described by its statistical properties. Such stochastic dynamics occur in a wide range of settings including financial markets [149], biological systems [150], impurity physics [151] and quantum heat engines [152]. Their evolution is hard to predict from microscopic principles, stimulating strong efforts to realize highly controlled model systems in optomechanics [153], cavity QED [154], superconducting circuits [155], trapped ions [156] and cold atoms [157].”

“Cold atomic mixtures offer a natural mapping of physical phenomena involving system and bath, wherein one species realizes the bath, while the other species represents the system. If a mesoscopic cloud of the first species is immersed in a Bose-Einstein condensate formed by the second species, it implements the Bose polaron problem [48, 158–160]. In recent quantum simulators of lattice gauge theories, the small clouds of one species emulate the matter field, while the gauge field is realized by the second atomic species [50, 161, 162].”

“The feasibility of immersing a few atoms into a large cloud was demonstrated in a dual-species magneto-optical trap (MOT) of rubidium and cesium [134]. This was extended towards the study of position- and spin-resolved dynamics of a single tracer atom acting as a probe [163, 164]. These microscopic degrees of freedom remain inaccessible for the large cloud with macroscopic number of atoms [152]. While the study of the many-body systems will involve the controlled immersion of single atoms in tweezers into a BEC, it is already possible to benchmark the interaction of system and bath in simpler experimental setups.”

This section focuses on the stochastic dynamics of a few sodium atoms in a large

cloud of potassium atoms. I describe how we have used the sensitive probe of single atom counting achieved on sodium to study its full counting statistics. The statistical quantities extracted enable us to characterize the distribution of single atom events and compare the influence of a potassium cloud on the occurrence of these events.

3.4.1 Full counting statistics of sodium

For extracting full counting statistics of sodium, we record 500 time traces with an integration time $\tau = 75$ ms per image. The reduced value of τ is chosen to not overheat the MOT coils running a current of 21 A. These time traces are then digitized by converting camera counts into discrete atom numbers. For digitization we aggregate the camera counts of all 500 time traces into one histogram to extract the calibration factor as described in section 3.2.1 and shown in figure 3.6 A: “In the digitized time traces, each change in atom counts corresponds to a load or loss event with one or more atoms. We observe that the dynamics are dominated by single atom events, as only 3% involve two or more atoms. Therefore, we neglect them in the following.”

“On average we observe $\bar{N}_{\text{load}} = 2.02(6)$ loading events per time trace, which is much smaller than the total number of images $N_{\text{img}} = 200$ taken per time trace. Given that the atoms come from a large reservoir, namely the oven region, the loading rate is independent of the number of loaded atoms. From these observations, we describe the loading process statistically as a series of independent Bernoulli trials with a success probability p_{load} . Therefore, the single atom loading probability for a time trace is given by :”

$$p_{\text{load}} = \frac{N_{\text{load}}}{N_{\text{img}}} \quad (3.9)$$

“The large number of images and the low loading probability means that the number of loading events N_{load} converges towards a Poisson distribution with mean \bar{N}_{load} . This stands in full agreement with the experimental observation as seen in figure 3.6 B:”

“Once an atom is present, it can be lost from the MOT with a probability p_{loss} . We observe an average number of $\bar{N}_{\text{loss}} = 1.29(5)$ loss events per time trace. Since we do not distinguish between atoms, the number of atoms lost in each time step can be described by a binomial distribution. Therefore, the number of single atom loss events per time trace N_{loss} enables us to extract the loss probability for a time trace as:”

$$p_{\text{loss}} = \frac{N_{\text{loss}}}{\sum_i N_i} \quad (3.10)$$

“The normalization factor is the sum of the number of atoms present in each image i . Similar to the loading case, we observe a Poisson distribution for the loss events with mean \bar{N}_{loss} , which can be attributed to the occurrence of only a few loss events over a large set of images.” After extracting p_{load} and p_{loss} for each time trace, we perform bootstrap resampling to extract mean and error bars.

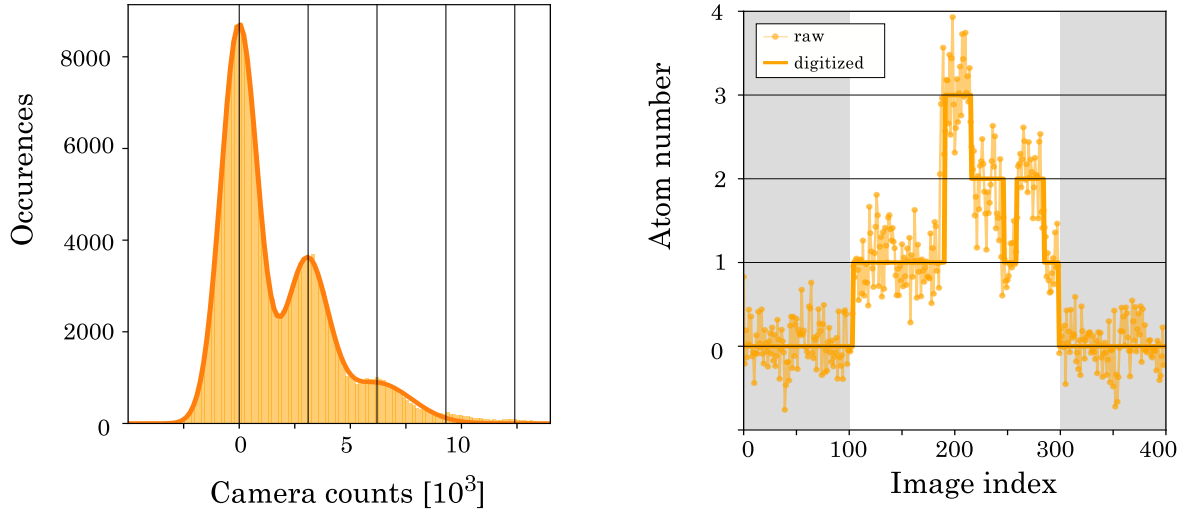
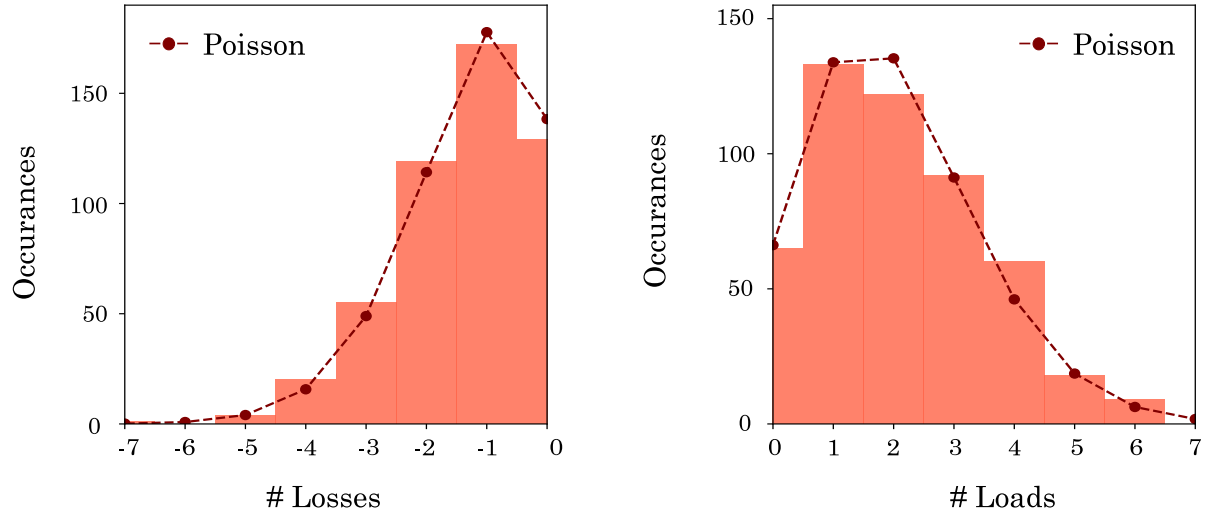
A

B


Figure 3.6: “Counting statistics of sodium. **A**: Histogram of recorded camera counts and digitization example for a time trace **B**: Histogram of the number of single atom losses and loads per time trace. The dashed lines show Poisson distributions with mean \bar{N}_{loss} and \bar{N}_{load} (extracted from the counting statistics).”

3.4.2 Effect of inter-species interaction

After characterizing the counting statistics in terms of load and loss probabilities, we can directly measure the effect of inter-species interaction. For this, we repeat the experiments of section 3.4.1 with and without the presence of a potassium MOT cloud (created by pre-loading for 5 s to ensure large atom numbers up to 20000). “To study the influence of the large potassium cloud on the dynamics of the few sodium atoms, we compare the load and loss statistics of the sodium atom counts with and without potassium atoms present. The extracted mean load and loss probabilities are summarized in table 3.1. The values corresponding to the absence and presence of potassium are indistinguishable to roughly within five percent.”

	p_{load} [%]	p_{loss} [%]
Without K	1.06(3)	2.76(23)
With K	1.02(3)	2.47(24)

Table 3.1: “Comparison of mean load and loss probabilities in a few atom sodium MOT with and without the presence of a potassium cloud. The uncertainties were obtained through bootstrap resampling.”

“To exclude experimental errors, we repeated the analysis for various configurations of relative positions of the two clouds, magnetic field gradients and laser detunings. All results were compatible with our observation of no influence of potassium on the sodium atom dynamics. We attribute these results to the extremely low density of the atomic clouds.”

Now, I give an estimate of quantities which cause the observed absence of inter-species interaction. Following [134] we can write down a rate equation for evolution of sodium atom number :

$$\frac{dN_{\text{Na}}}{dt} = R(N_{\text{K}}) - \gamma N_{\text{Na}} - \beta_{\text{NaK}} \int n_{\text{K}}(r, t) n_{\text{Na}}(r, t) d^3r - \beta_{\text{NaNa}} \int n_{\text{Na}}^2(r, t) d^3r \quad (3.11)$$

where the first term is the K atom number dependent loading rate $R(N_{\text{K}})$; the second term describes the loss of Na atoms due to collisions with background gas particles at a rate γ ; the third and fourth terms describe, respectively, the loss of Na atoms due to collisions with a K and Na atom, characterized by the inelastic collision coefficients β_{NaK} and β_{NaNa} .

In [134] the authors extract Rb-Cs inter-species inelastic collision coefficient β_{RbCs} to be $(6.6 \pm 0.3) \times 10^{-10} \text{ cm}^3 \text{ s}^{-1}$. To observe this, they reported a maximum atom density of rubidium to be $6.5 \times 10^{10} \text{ cm}^{-3}$. For a Na-K mixture the coefficient β_{NaK} has been measured (without using single atom counting) before to be $(3.0 \pm 1.5) \times 10^{-12} \text{ cm}^3 \text{ s}^{-1}$ [165]. Since the value of β_{NaK} is ~ 200 times smaller than β_{RbCs} , the density of potassium in our case should be about 200 times more than the rubidium density to observe a similar effect. However, the estimated density of our potassium MOT is about $3.7 \times 10^6 \text{ cm}^{-3}$ which is too small to observe an inter-species effect for Na-K.

The reason for such a low density of the potassium MOT is the higher temperature of a potassium MOT compared to a rubidium MOT due to the poorly resolved excited state hyperfine structure [146]. For this reason, to achieve a higher potassium MOT density, we would have required much higher MOT magnetic field gradients than the value of 21 G cm^{-1} used in our experiments. However, our passively air cooled MOT coils posed severe limitation to obtain any further compression. This study nicely demonstrated the limits of our air-cooled coils, which motivated us to look for a compact water-cooled design for further experiments [108]. This design is explained in detail in section 4.2.

Towards ultracold quantum mixture

An important goal of our SoPa experiment is building a quantum gas mixture experiment where we can controllably couple a quantum gas of one species with another. A first milestone towards this goal would be obtaining a quantum degenerate gas of sodium and trapping single potassium atoms in optical tweezers. In this chapter, I describe the steps we have taken so far and the problems we have faced towards realising this goal. At the time of writing this thesis we have not yet obtained a quantum degenerate gas of sodium, but we have demonstrated all the necessary experimental steps on our setup. Similarly for optical tweezers, we have observed a first indication of atoms being trapped in the tweezers. I conclude this chapter by mentioning the modifications we have planned based on what we learnt from our journey so far.

4.1 Bose-Einstein condensation

Bose-Einstein condensation is a quantum phase transition which occurs in a system of bosons usually at very low temperatures. It was predicted by Albert Einstein [166] following the work of Satyendra Nath Bose [167]. Consider an ideal gas of free bosons. This means the energy of each particle is given by $\epsilon = p^2/2m$ where p is the momentum and m is the mass of the particle. The occupation number of single particle energy states is given by:

$$f_\epsilon = \frac{1}{e^{\beta(\epsilon-\mu)} - 1} \quad (4.1)$$

where $\beta = 1/k_B T$ is the thermodynamic beta and μ is the chemical potential. The chemical potential is fixed by the following constraint:

$$N = \sum_\epsilon f_\epsilon \quad (4.2)$$

where N is the total number of particles. The maximum value of the μ is 0 because if $\mu > 0$ then the occupation number of the $\epsilon = 0$ state would be negative. We can separate the summation in equation 4.2 into two parts : occupation number of the $\epsilon = 0$ state and that of $\epsilon > 0$ states :

$$N = N_0 + \sum_{\epsilon>0} f_\epsilon \quad (4.3)$$

where N_0 is the occupation number of $\epsilon = 0$ state. Rearranging to get the occupation of all $\epsilon > 0$ states, we get:

$$N - N_0 = \sum_{\epsilon > 0} f_\epsilon \quad (4.4)$$

In the thermodynamic limit (i.e. $N \rightarrow \infty$, $V \rightarrow \infty$ such that $N/V = \text{constant}$ where V is the volume) we can replace the summation above by an integral by using the density of states for free particles $g(\epsilon)d\epsilon = 2\pi(\frac{2m}{h})^{3/2}V\sqrt{\epsilon}d\epsilon$:

$$\frac{N - N_0}{V} = 2\pi\left(\frac{2m}{h}\right)^{3/2} \int_0^\infty \frac{\sqrt{\epsilon}d\epsilon}{e^{\beta(\epsilon-\mu)} - 1} \quad (4.5)$$

The limit of the integral starting at 0 is fine because the factor of $\sqrt{\epsilon}$ assigns zero weight to the $\epsilon = 0$ state. The integral in equation 4.5 is maximized for maximum value of μ (i.e. $\mu = 0$). Performing the integral for $\mu = 0$ gives [168]:

$$\frac{N - N_0}{V} \leq 2\pi\left(\frac{2m}{h}\right)^{3/2} \int_0^\infty \frac{\sqrt{\epsilon}d\epsilon}{e^{\beta\epsilon} - 1} = 2.612\left(\frac{2\pi mk_B T}{h^2}\right)^{3/2} \quad (4.6)$$

Denoting $(N - N_0)/V$ by n and $h/\sqrt{2\pi mk_B T}$ by λ_{db} , we can re-write equation 4.6 as:

$$n\lambda_{db}^3 \leq 2.612 \quad (4.7)$$

The quantity λ_{db} is called the thermal de-broglie wavelength. If the quantity $n\lambda_{db}^3$ in equation 4.7 is increased beyond the saturation limit of 2.612 (by either increasing the density or decreasing the temperature), the excess particles cannot disappear because of number conservation. Instead they occupy the state with $\epsilon = 0$ (i.e. $p = 0$), this phenomenon is called Bose-Einstein condensation. It has been experimentally observed in several systems including ultracold atomic gases [16–18].

Bose-Einstein condensate (BEC) is a state of matter in which there is “macroscopic” occupation of the ground state. The quantity $n\lambda_{db}^3$ is called the phase-space density and it defines the precise meaning of the word “macroscopic” via equation 4.7 by setting a cutoff for the density of particles which are not in the ground state. In an ideal Bose gas at constant volume, Bose-Einstein condensation is a second order phase-transition showing a discontinuity in the derivative of the specific heat at the transition point [169–171].

There exist several criteria for a BEC [172]. A very general criterion is the Penrose-Onsager criterion or the off-diagonal long range order (ODLRO) criterion [173]. A system is said to possess an ODLRO if the single-particle density matrix

$$\rho_1(\mathbf{r}, \mathbf{r}') \equiv Tr \left\{ \hat{\rho} \hat{\psi}^\dagger(\mathbf{r}) \hat{\psi}(\mathbf{r}') \right\} \equiv \langle \hat{\psi}^\dagger(\mathbf{r}) \hat{\psi}(\mathbf{r}') \rangle \quad (4.8)$$

has a large eigenvalue, i.e., an eigenvalue proportional to the total number of particles N , where $\hat{\rho}$ is the density operator of the system and $\hat{\psi}^\dagger(\mathbf{r})(\hat{\psi}(\mathbf{r}'))$ is the field operator that creates (annihilates) a particle at $\mathbf{r}(\mathbf{r}')$. Stated differently, the one-particle reduced density matrix of the system tends to a non-zero value as $|\mathbf{r} - \mathbf{r}'| \rightarrow \infty$. If this condition holds, the system is said to maintain spatial coherence over a long distance. For

the case of an ideal Bose gas considered above ODLRO exists if and only if there is a macroscopic occupation of the $\epsilon = 0$ state [174, 175].

This section briefly explained BEC and related concepts with simplifying assumptions like no interactions, no trapping potentials etc. In experiments with ultracold atoms these assumptions are not satisfied and a more rigorous treatment has to be done. A detailed explanation is beyond the scope of this thesis and the reader is referred to the extensive literature [175–180, 109].

4.2 Electromagnet coil upgrade

The cooling steps for obtaining a BEC involve using magnetic field gradients of several hundred G cm^{-1} to confine the atoms. This involves using high currents and therefore an active cooling solution is required. As mentioned in section 2.4.5, the first version of our MOT coils was passively air cooled. To upgrade the coils we adopted a compact design [108] from the group of Prof. Jean-Philippe Brantut at EPFL Lausanne. The coil manufacturing uses the technique of wire erosion [181] to cut out a copper spiral in a solid block of copper. Apart from being compact an added advantage for this design is that our workshop has the capability of doing wire-erosion and so manufacturing could be performed much faster. Also we could easily design the shape of our coils adapted to our science chamber ensuring that optical access is not blocked by the coils.

We used this technique to cut a spiral such that each winding is 0.7 mm thick and 15 mm high. The pitch of the spiral is 1 mm per winding and the separation between consecutive windings is 0.3 mm. The total number of windings in the spiral is 40. After cutting out the spiral, we insert fiber glass spacer of 0.3 mm thickness between the spiral ridges to stabilize the structure and to prevent consecutive windings from short circuiting. Finally, the entire structure is filled (using syringe needles) with a mixture of epoxy plus aluminum nitride to make it stable and water-tight. We take care that there are no gaps or air bubbles during filling the epoxy.

Once the epoxy solidifies, we machine out a fresh surface of copper and remove the extra copper around the coil body. For water cooling, we screw (using titanium screws) a PEEK cap on this freshly exposed copper surface. The PEEK cap has inlets and outlets for water cooling pipes and the coil gets cooled by the contact of copper surface with water channel flowing on top of it. We produced two such coils for using them as the anti-Helmholtz pair for our MOT and magnetic trap. The pictures showing the process of coil manufacturing are presented in figure 4.1.

We use a programmable power supply from Delta Elektronika (*SM 15-400*) to run currents in the coils. The power supply can provide upto 400 A at 15 V. The current output in constant current mode can be remotely tuned by applying a control voltage from 0 V to 5 V. For water circulation, we use a chiller (*Riedel SC21*) which can

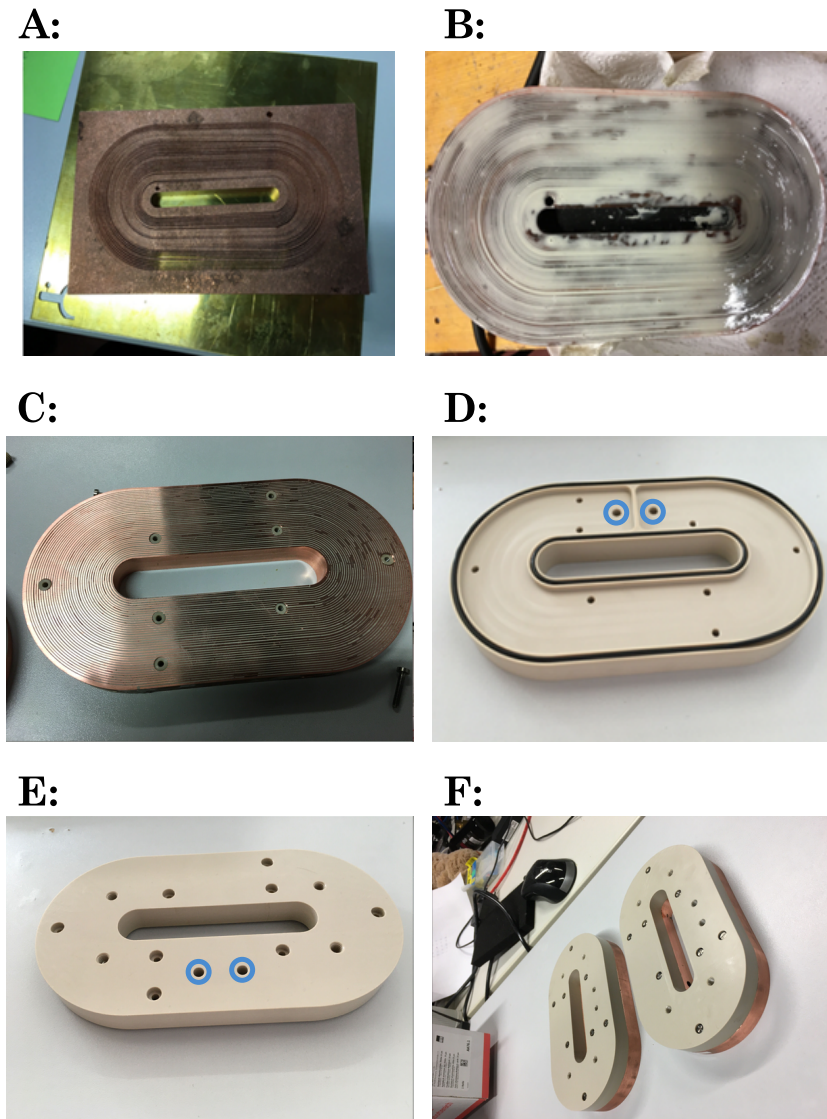


Figure 4.1: MOT coil manufacturing. **A:** A spiral is cut in the copper block. **B:** The spiral is filled with epoxy mixture. **C:** After epoxy solidifies extra copper is machined out and a fresh copper surface is exposed. Holes are drilled into the copper body to secure the PEEK part from top. **D: and E:** The PEEK part with holes for water marked with blue circles. **F:** Final assembled coil pair.

maintain a maximum pumping pressure of 3 bar and has a cooling capacity of 2.2 kW. The water cooling is least effective in the outmost and inner most edges of the coil due to poor contact with water. These areas are the hottest during operation, so for safety we monitor the temperature of the outermost winding of the coil by using a thermocouple connected to an Arduino and if it exceeds 50°C an interlock circuit controlled by the Arduino shuts down the power supply.

The coil pair in our setup is 75 mm apart. With this setup, we can get a gradient of $\sim 1 \text{ G cm}^{-1} \text{ A}^{-1}$. The figure 4.2 shows a measurement of magnetic field for a current of 20 A.

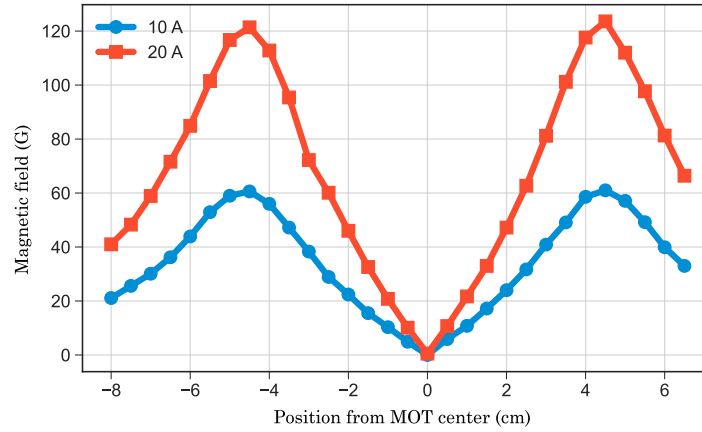


Figure 4.2: MOT coil magnetic field measurement.

4.3 Optimizing sodium 2D and 3D-MOT

For obtaining a quantum degenerate gas with a high phase-space density, it is important to have a large sample of laser cooled atoms. In our experiment this boils down to improving the loading rate, atom number and density of the 3D MOT. Our first 3D MOT of sodium had only few million atoms. To improve this we started by optimizing the 2D MOT. Since the 2D MOT captures atom from a vapor, we increased the vapor pressure by heating the sodium oven to a temperature of 215°C . At this temperature the pressure in sodium 2D MOT chamber reached $\approx 10^{-8}$ mbar.

To further increase the 2D MOT capture, we increased the diameter of the 2D MOT beams from ≈ 10 mm to ≈ 20 mm. As mentioned in section 2.4.4, our 2D MOT setup has a Zeeman slower like magnetic field configuration. This can be used to improve the 2D MOT by using a slowing beam [71]. However, when we tried to implement the Zeeman slower using the frequency detuning of ~ -300 MHz mentioned by previous experiments [71, 182], we did not see any effect of the slower beam in improving the 2D MOT. Then we tried to optimize the slower beam by scanning the laser frequency and found the working region for the slower very far away from what the previous experiments have mentioned. The optimum detuning for us was -103 MHz. A possible reason for the observed deviation in our setup could be our magnetic field configuration and the oven positioning not being exactly like those reported in literature.

After improving the 2D MOT, we also optimized the 3D MOT by making the beams bigger (from ≈ 10 mm to ≈ 14 mm diameter) and optimizing the MOT alignment. The optimized parameters are given in table 2.1. The optimized magnetic field gradient for the MOT is 8 G cm^{-1} . All these improvements helped us to increase the atom number from a few million to about a billion atoms in the 3D MOT. The improvement in MOT atom number is clearly visible in a comparison of pictures (shown in figure 4.3) taken before and after optimization.

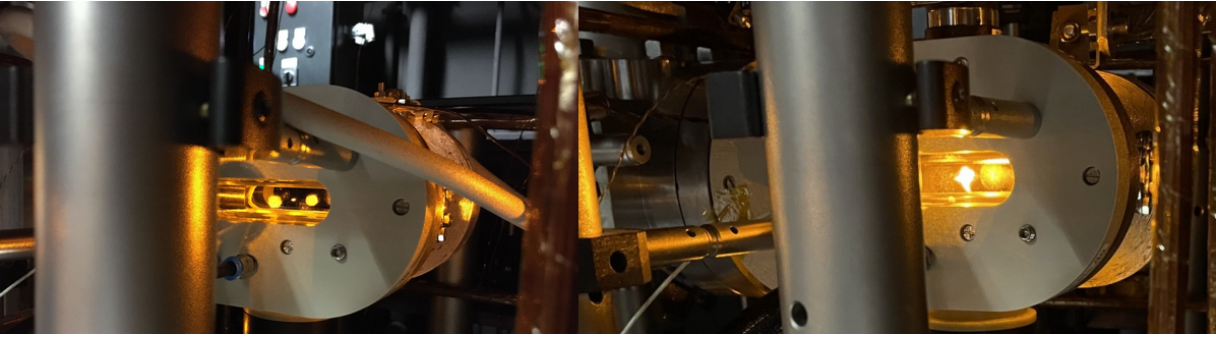


Figure 4.3: MOT optimization result. The left picture shows the bright MOT before optimization with an estimated atom number of a few million. The right picture shows the bright MOT after optimization with an estimated atom number of about one billion.

Although we improve the atom number in the MOT, this does not necessarily lead to increased phase space densities because the atom density in a MOT is limited to about 10^{11} cm^{-3} . After this limit increasing atom number increases the size of the MOT but not the density. This limit arises from two processes : photon re-scattering [183] and light assisted collisions [184]. Photon re-scattering refers to re-absorption of photons emitted by trapped atoms. It leads to an effective repulsive interaction between the atoms, thus preventing higher densities. Light assisted collisions involve excitation of an atom pair into a higher molecular potential. As the pair rolls down the potential, it gains kinetic energy. If the kinetic energy gained before it radiates back to the ground state, is more than the trap depth, the pair can get lost from the trap.

One way to increase the density of atoms in a MOT is to use a Dark-SPOT (Spontaneous optical trapping) MOT [185]. It is achieved by creating an annular re-pumping beam such that there is a circular region in its center where there is no re-pumping light. In this region after a few cooling cycles the atoms are optically pumped into the dark hyperfine state and thus they do not interact with the cooling light. This allows atoms to get accumulated at about an order of magnitude higher density than in a MOT. For our setup we shine the re-pumping beam from the side oval window of the science chamber. A dark spot is created by imaging a paper disk through the re-pumping beam onto the MOT region. We have observed that using a dark spot MOT leads to better efficiency (than a bright MOT) of atom transfer into the magnetic and optical dipole traps.

For characterizing the atom clouds at different stages, we need to know the atom number and temperature. We estimate these quantities with fluorescence imaging and time-of-flight technique as explained in section 2.5. As an example, figure 4.4 shows the temperature measurement for the dark-SPOT MOT. The imaging pulse length is $100 \mu\text{s}$ and the estimated atom number is 10^9 with an average temperature of about $505 \mu\text{K}$.

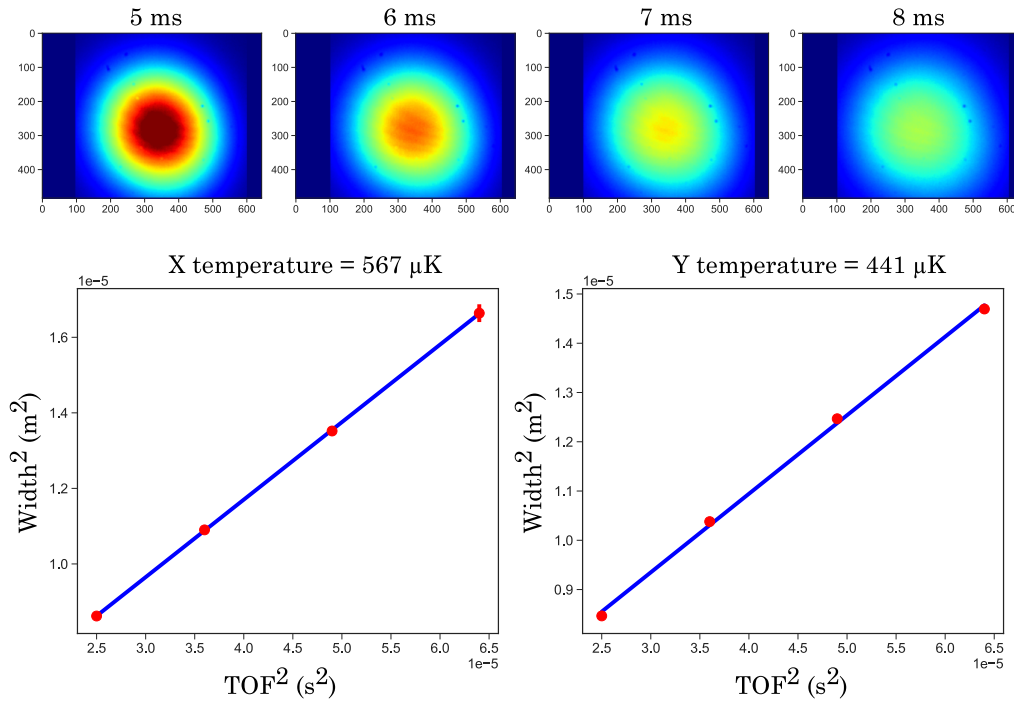


Figure 4.4: Dark-SPOT MOT temperature measurement. These measurements are done using a Mako camera from the side. The dots in the images are dirt particles on the camera sensor.

4.4 Compressed MOT and optical molasses

Although using a dark-SPOT MOT helps to increase the density, it is usually not enough because the temperature and densities are both limited which in turn limits the phase-space density [186]. To optimize the phase-space density for next steps of cooling, we first use a compressed MOT (temporal dark-spot MOT) [187, 188]. The idea is similar to a dark-SPOT MOT in that we want to reduce the coupling of atoms to the cooling light so as to be able to compress them more. This is achieved by detuning the cooling and re-pumping beams to -30 MHz and increasing the magnetic field gradient to 60 G cm^{-1} . This helps in achieving higher densities but it only exists transiently, so this is done at the end of MOT loading for about 6 ms.

Another method to improve the phase-space density is to decrease the temperature of atoms. As discussed in section 2.4.2, the temperature of atoms in Optical molasses can be much lower than the Doppler temperature limit. So we use optical molasses arrangement for 6 ms immediately after the compressed mot phase by turning of the MOT coils and keeping the beams on with compressed MOT detunings, to lower down the atom temperature. In our setup we have observed cooling in molasses to about $100 \mu\text{K}$.

4.5 Magnetic trapping and microwave evaporation

The laser cooling and trapping mechanisms described so far can not produce a BEC. These mechanisms are fundamentally limited by the scattering of photons by atoms which gives them heating momentum kicks. For further cooling, the atoms have to be captured in different types of traps where they do not scatter photons (or do it very slowly). For such traps the cooling mechanism is also different. The cooling mechanism for one such type of trap i.e. a magnetic trap, is described in this section, along-with the experimental results from our setup in trying to achieve the cooling.

A magnetic trap [189] uses the Zeeman effect to produce a trapping potential for atoms whose internal states have a non-zero magnetic dipole moment. For an atom in a given $|F, m_F\rangle$ state the Zeeman energy is :

$$E = g_F m_F \mu B \quad (4.9)$$

where E is the energy, g_F is the Lande factor and μ is the Bohr magneton. To trap the atoms one needs to provide a local minimum in the potential energy. This can be achieved using either a magnetic field local minimum (for states with $g_F m_F > 0$) or magnetic field local maximum (for states with $g_F m_F < 0$). However, a local maximum in magnetic field is prohibited by Maxwell's equations, so we work with a magnetic field local minimum. Atomic internal states which are trappable in such a configuration (i.e. those with $g_F m_F > 0$) are called low-field seekers.

The simplest configuration for producing a magnetic trap is using an anti-Helmholtz coil pair to produce a quadrupolar magnetic field. This configuration is already used in a MOT although the currents required for a magnetic trap are much higher. In such a setup the atoms in low-field seeking states accumulate near the magnetic field minimum at the middle of the coil-pair axis.

At the center of the trap the magnetic field is zero and all m_F states are degenerate. Atoms close to this point can thus transition from a trappable m_F state to a non-trappable one due to external perturbations. This loss called Majorana loss [190, 191] causes atom number loss and limits the lifetime of atoms in a magnetic trap. However, Majorana losses become significant only when the atoms are very cold ($< 50 \mu\text{K}$) and thus do not pose a serious obstacle in performing cooling to some extent as long as the temperatures are not too low [192]. Another factor limiting the atom lifetime in a magnetic trap is the collisions with the background gas. This causes an exponential decay of the atom number with time. For higher lifetime, a better vacuum is required.

To cool the atoms in a magnetic trap the process of evaporative cooling [193] is used. It involves removing the hottest atoms from the trap while the remaining atoms rethermalize by collisions. After rethermalization, the average temperature of the atom cloud is reduced because each of the hottest atoms carry away more energy than the average energy per atom. Continuing this process, very low temperatures (μK) can be achieved and this process has been used for obtaining BECs [16–18].

A crucial requirement for the cooling to work is that the atoms remaining in the trap rethermalize fast enough before more atoms are removed. This depends on the elastic collision rate which should be more than few Hz. To achieve this, the density should be sufficiently high. As the cooling takes place the velocity of atoms decreases which should decrease the elastic collision rate, but the increase in density due to cooling dominates over slower velocities such that overall the elastic collision rate increases as evaporation continues. This is called the runaway regime which causes tremendous increase in phase-space density.

In order to remove the hottest atoms from the magnetic trap, one can do either free evaporation or forced evaporation. In free evaporation the atoms are simply held in the trap and the hotter atoms leave the trap with time. This is usually a slower process than forced evaporation. In forced evaporation an external electromagnetic radiation (either radio frequency or microwave) is used to transfer atoms from the low-field to high-field seeking state which causes those atoms to be lost. The selection of hottest atoms is done by properly selecting the frequency of the electromagnetic radiation. For continuous removal of the hottest atoms, a frequency sweep has to be performed in given range.

4.5.1 Experimental results

After the compressed MOT and molasses phase, we optically pump all sodium atoms to the $F = 1$ hyperfine ground state by turning off the repumper 1 ms before the cooling light. The $F = 1$ state has three Zeeman sublevels : $m_F = 1, 0, -1$ out of which $m_F = -1$ is the low field seeker. The figure 4.5 shows the Zeeman shift of sodium $F = 1$ and $F = 2$ ground states.

After all laser beams are turned off, a magnetic field gradient of 160 G cm^{-1} is turned on in 100 ms by passing a current of 160 A through the coils. Since the atoms are equally distributed between the three m_F states, we can only catch maximum one-third of the initial population. For the trapped atoms, we estimate a trap lifetime of 20 s by observing the decay of the number of trapped atoms with time. A measurement curve for estimating the lifetime of the atoms in the magnetic trap is shown in figure 4.6.

On loading the atoms in the magnetic trap, we perform forced microwave evaporation to transfer the hottest atoms from $|F, m_F\rangle = |1, -1\rangle$ to $|F, m_F\rangle = |2, -2\rangle$. Using microwave instead of RF allows selectively evaporating only sodium and the possibility of sympathetically cooling potassium [194, 195]. For generating the microwave, we use a SynthHD microwave generator (as mentioned in section 2.6). The microwave signal from the SynthHD is connected to a 10 W RF amplifier which is connected to an antenna which is a single loop coil of diameter 2 cm. The antenna is placed closed to the top viewport of the science chamber. The microwave frequency is swept linearly from 1625 MHz to 1745 MHz in 7.2 s.

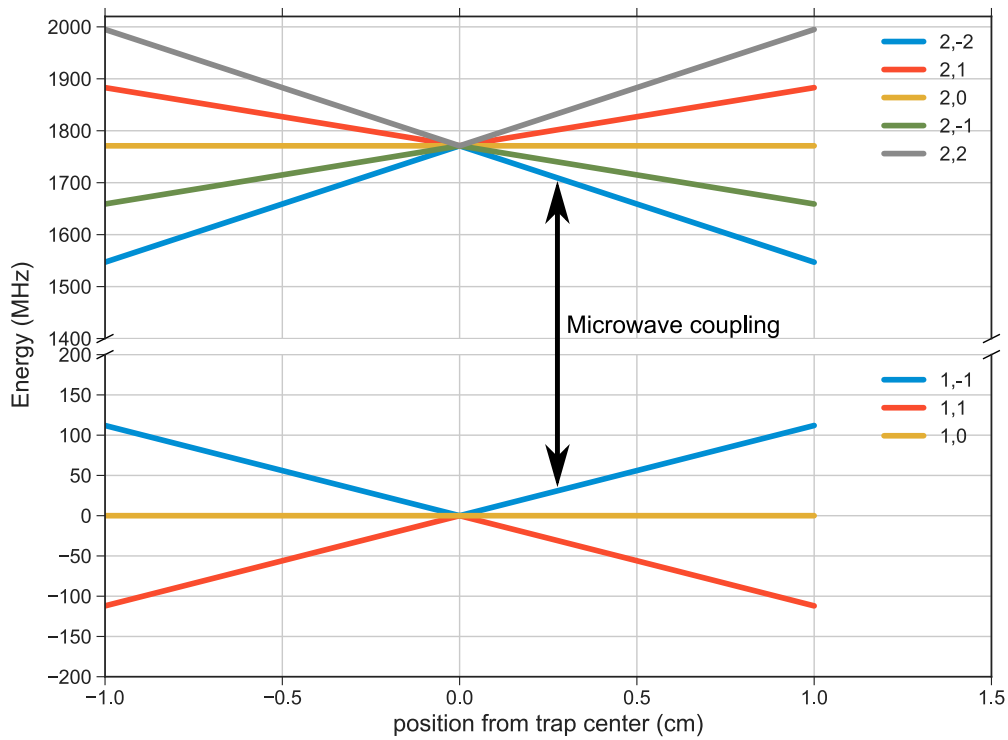


Figure 4.5: Zeeman shift of sodium $F = 1$ and $F = 2$ ground states.

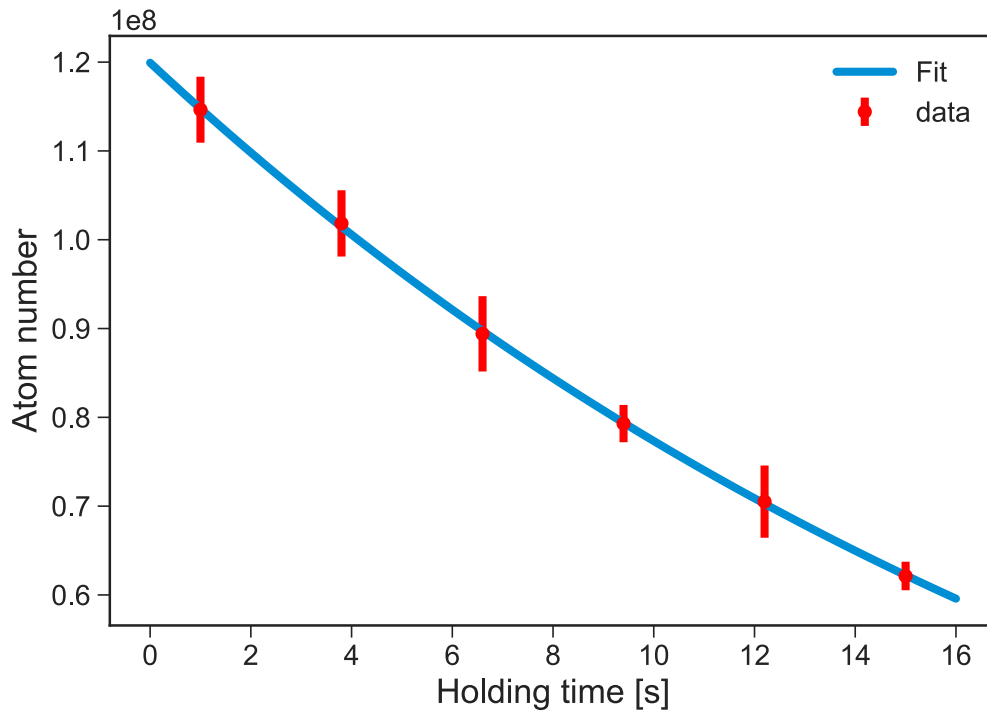


Figure 4.6: Magnetic trap lifetime measurement. The estimated $1/e$ lifetime from the exponential fit is 20 s.

The starting population of atoms in the magnetic trap is about 3×10^8 at a temperature of $\sim 238 \mu\text{K}$ and we are left with few million atoms at the end of the sweep. However, although we loose more than two orders of magnitude in atom number, the corre-

sponding reduction in temperature is negligible so we do not observe any gain in phase-space density from the evaporation. This shows that we are not operating in the runaway regime of evaporation. A comparison of atom number and temperature at three different stages is given in table 4.1. We repeated the evaporation for several values of microwave frequency range, sweep duration, magnetic trap gradients etc. but could not observe a runaway evaporation effect. For the magnetic trap gradients, we could only go upto a maximum value of $\sim 220 \text{ G cm}^{-1}$ (using a current of 220 A) which comes from the limit of water cooling efficiency of our current coil design.

Stage	Atom number	Average temperature (μK)
Dark-SPOT MOT	10^9	505
Magnetic trap before evaporation	3×10^8	238
Magnetic trap after evaporation	10^6	155

Table 4.1: Comparison of atom number and temperature at different stages in the experiment.

We have also compared the literature values cited by other research groups who perform evaporative cooling in the $|F, m_F\rangle = |1, -1\rangle$ state for sodium. This is summarized in the table 4.2. By comparing the values we realize that our starting atom number, temperature and density is still not optimized for runaway evaporative cooling. Therefore, the next steps on the experiment are planned to improve the value of these parameters.

Reference	Gradient (G cm^{-1})	Atom number before evaporation	Temperature before evaporation (μK)
SoPa	160	3×10^8	238
[196]	216	10^9	100
[197]	212	10^9	—
[18]	550	10^9	200
[182]	190	1.5×10^9	220
[198]	1000	3×10^9	100
[199]	118	1.4×10^{10}	350

Table 4.2: Comparison of magnetic trap evaporation settings of other experiments who perform evaporation in $|F, m_F\rangle = |1, -1\rangle$ state for sodium.

As next steps, we plan to use an independent laser which can be used to either put more power into the 2D-MOT (either cooling beams or Zeeman slower) to improve its efficiency (and hence the 3D-MOT) or use it for performing gray molasses cooling which has been reported to achieve temperatures as low as $9 \mu\text{K}$ [200]. We can only use the new laser for either one of the two purposes but both of them should lead to improved initial conditions for evaporative cooling. If we decide to go for gray molasses cooling, we will also change the 3D-MOT from retro-reflected configuration to a 6-beam configuration, because the performance of gray molasses is quite sensitive to counter-propagating beam power balancing.

We have also noticed that the power in our repumping beam for the dark-SPOT MOT is only 2 mW and most of it gets blocked by the dark spot. One reason for such a low power is the poor diffraction efficiency ($\sim 15\%$) of the 1.8 GHz AOM used for generating the repumping beam. We plan to replace it by a quadruple pass configuration using a 350 MHz AOM which we have already observed to give better efficiency of $\sim 40\%$. To reduced the wastage of repumper power blocked by the dark spot, we might implement better beam shaping with axicon lenses which will transfer more power to the outer parts of the beam. Finally, our current water-cooled coils can support a maximum gradient of about 220 G cm^{-1} before overheating starts. We would like to be able to reach higher gradients for stronger compression in the magnetic trap. So we are planning to upgrade the coil design using coils made with hollow copper pipes which should allow much better cooling efficiency and higher magnetic gradients.

4.6 Optical dipole trap

An optical dipole trap [201, 202] presents another possibility to trap ultracold atoms. It is based on the effect of laser light on the atomic energy levels (also called AC stark effect). Using second-order perturbation theory, one can show that interaction with a far-detuned laser light causes a shift of energy levels proportional to the intensity of the laser beam. When the electromagnetic field of the laser is spatially inhomogeneous, the energy shift also varies with position thus creating a potential landscape for the atom. Using a focused far red-detuned Gaussian laser beam, an attractive potential for atoms can be created wherein the atoms tend to accumulate at the focus.

Due to large detuning, this type of trapping mechanism is insensitive to the m_F sublevels and can provide trapping potential for all internal states. Since the internal states are then available as extra degrees of freedom, a dipole trap can be used for experiments which study dynamics in these internal states [203]. Additionally, the large detuning suppresses the photon scattering rate and atoms can be stored for several seconds in the trap. Using an optical dipole trap does not require high currents like a magnetic trap and also speeds up the experiment cycle.

For a dipole trap, the trap depth of the potential is given by:

$$U_{\text{dip}}(\mathbf{r}) = -\frac{3\pi c^2}{2\omega_0^3} \left(\frac{\Gamma}{\omega_0 - \omega} + \frac{\Gamma}{\omega_0 + \omega} \right) I(\mathbf{r}) \quad (4.10)$$

and the scattering rate is given by:

$$\Gamma_{\text{sc}}(\mathbf{r}) = \frac{3\pi c^2}{2\hbar\omega_0^3} \left(\frac{\omega}{\omega_0} \right)^3 \left(\frac{\Gamma}{\omega_0 - \omega} + \frac{\Gamma}{\omega_0 + \omega} \right)^2 I(\mathbf{r}) \quad (4.11)$$

where c is the speed of light, \hbar is the Planck constant, ω_0 is the atomic transition frequency, Γ is the natural linewidth, ω is the frequency of dipole laser and $I(\mathbf{r})$ is

the dipole laser intensity. An important parameter for a dipole trap is the trapping frequency which determines the timescales of spatial dynamics in the trap. Using harmonic approximation, for a single beam dipole trap, the radial ω_{rad} and axial ω_{ax} trapping frequencies can be estimated as :

$$\omega_{\text{rad}} = \sqrt{\frac{4U_{\text{max}}}{mw_0^2}}, \quad \omega_{\text{ax}} = \sqrt{\frac{2U_{\text{max}}}{mz_{\text{R}}^2}} \quad (4.12)$$

where U_{max} is the maximum trap depth, m is the mass of the atom, w_0 is the focus waist radius and z_{R} is the Rayleigh range. Due to a larger value of Rayleigh range compared to the focus waist, the axial trapping frequencies are much less than the radial one which leads to weaker trapping in axial direction. To get a stronger trapping also in the axial direction another focused beam can be used at an angle such that the foci of the two beams overlap. This configuration is called crossed dipole trap [204] where the trap depth from the two beams add up. In case of a crossed dipole trap, care should be taken to avoid interference effects at the focal overlap of the two beams. This can be achieved by either using orthogonal polarization for both beams or by introducing a frequency difference between the two focused beams. If the frequency difference is much more than the trapping frequencies, any interference pattern will oscillate at very fast rates and the atoms will only see an average potential devoid of interference effects.

Cooling in the dipole trap can be achieved by performing an evaporation in a similar principle like magnetic trap of removing the hottest atoms while letting the remaining atoms rethermalize to a lower temperature. However, the usual method to perform evaporation in a dipole trap is different from a magnetic trap. It is done by lowering the power in the dipole trap beams which lowers the trap depth allowing the hotter atoms to escape while the remaining ones rethermalize [204].

4.6.1 Experimental results

In the experiment, we implement a crossed dipole trap using a 100 W 1070 nm infrared (IR) laser from IPG photonics. Our setup is shown in figure 4.7.

The light from the IR laser is spilt into two paths of equal power. In each path the light is diffracted via an AOM and the first diffracted order is used for generating the crossed dipole trap. This has two benefits : first it allows to regulate trap depth by controlling the RF power of the AOM. Such a control is helpful for both performing evaporative cooling in the dipole trap and for stabilising power in the beams. The second benefit of using the AOM is that it allows us to introduce a relative frequency of about 220 MHz between the two arms of the crossed dipole trap thus preventing interference effects.

We measured the focus spot size of the dipole beam using the knife-edge method. The measured focus waist radius was $45 \mu\text{m}$. The resulting trap depth for sodium

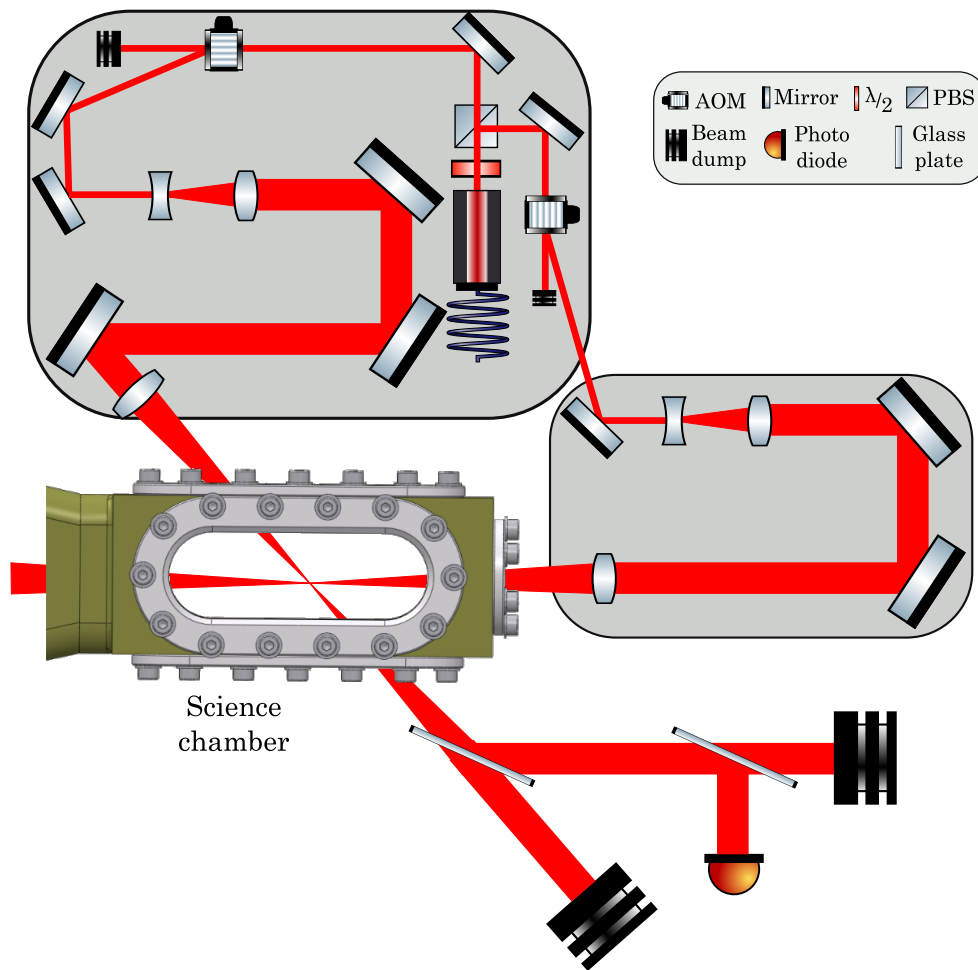


Figure 4.7: Setup of the crossed dipole trap. Two breadboards host the optics required for the two arms. Intensity stabilization is shown for only one arm.

is estimated to be around $160 \mu\text{K}$ per 10 W . Since the intensity of the beam directly determines the potential, any fluctuation in the laser power will cause fluctuation in trapping potential which can cause heating of atoms. So we need to actively stabilize the power in both arms of the crossed dipole trap. For this, after the beams have passed through the science chamber, we sample a small portion of each beam using a two-stage beam sampler. This sampled light is incident on a photo-diode which provides signal to a PID control module.

The IR laser is invisible to human eye and due to its high power, it is very dangerous to operate with. For laser safety, proper care is taken in terms of using laser goggles and using a thick aluminum frame as a laser protection cover around the main table and dipole trap breadboards. The invisibility of the IR light also makes it hard to align. Although one can use IR fluorescent cards, they suffer from issues of saturation and blooming. Instead, we use an IR fluorescent glass detector (from *Precision Laser Scanning*) which makes it very easy to continuously observe the IR beam and is designed to be used at higher laser powers. To achieve a first rough alignment, we

use a resonant yellow light on the MOT such that it destroys the MOT. This beams now marks the path on which we overlap the IR beam using the glass detector for visualization. Once we get a signal in the camera images, fine tuning is done via turning the screws of the mirror closest to the science chamber.

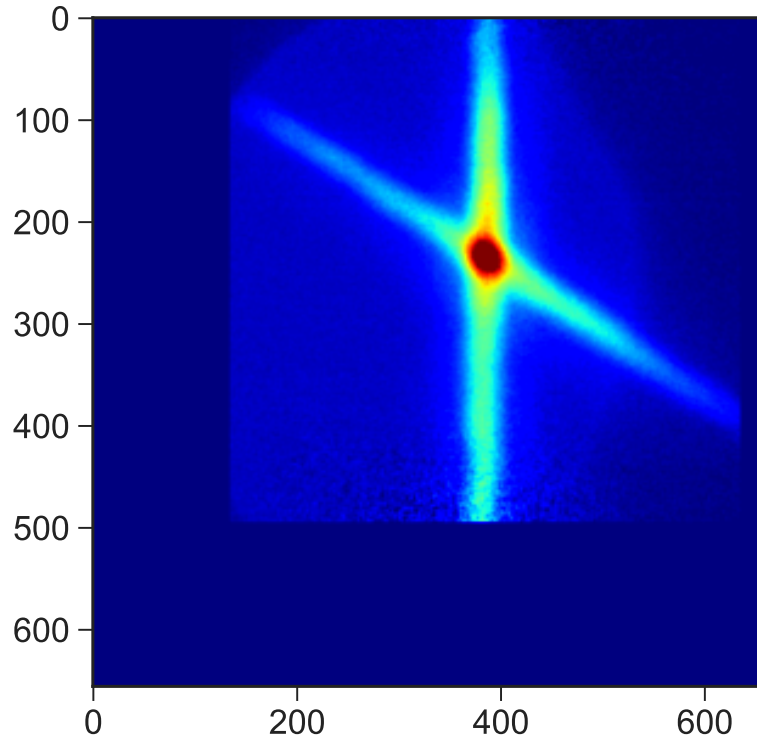


Figure 4.8: Image of atoms captured in the crossed dipole trap. Only a selected region of the full image is shown.

To load atoms into the dipole trap we operated the IR laser at full power of 100 W and have tried two strategies : load directly after the molasses stage and load from the magnetic trap. We observed loading in both cases with maximum number of loaded atoms estimated to be about 1×10^5 . The loaded atoms had a temperature of about $150 \mu\text{K}$. An image of atoms captured in the crossed dipole trap is shown in figure 4.8.

On trying to perform evaporation in the dipole trap, although we did observe some cooling effect, but the loss of atoms was much more than the decrease in temperature thereby giving no gain in phase-space density. The reasoning for the observed behaviour is similar to that for the magnetic trap as in the initial conditions are not favourable for the evaporation to work. To resolve this we plan to first fix the issue of poor cooling performance in the magnetic trap. Once this is done, we will perform some pre-cooling in the magnetic trap and load the pre-cooled atoms into dipole trap. This should substantially improve the efficiency of the subsequent evaporation in the dipole trap and possibly allows observation of a BEC.

4.7 Optical tweezers

Optical tweezers operate on the same principle as optical dipole traps and have been used to trap single atoms [205]. This is achieved by operating in the so-called collisional blockade regime [206] which occurs when the focus waist size of a red-detuned beam is less than few microns. In this regime the number of trapped atoms in the tweezers is either 0 or 1. Using optical tweezers an array of neutral atoms can be prepared for quantum simulation/computation purposes [73, 207]. In a mixture experiment optical tweezers can be used to immerse local impurities of one species into an quantum gas of another species. Such a setup can be used to study a wide range of physics like impurity physics, quantum thermodynamics etc.

4.7.1 Array of tweezers and defect-free sorting

A common method to generate a 1D-array of tweezers, is to drive an acousto-optic deflector (AOD) with multiple frequency tones. Since loading of atoms in the tweezers is a probabilistic process, some tweezers are not loaded causing a defect in the array. To prepare a defect free array, one needs to rearrange the traps for eliminating the defects [73, 207]. In order to rearrange the tweezers, one has to dynamically change the frequencies depending on the loading configuration obtained initially. In our case, we use an arbitrary waveform generator (AWG), Spectrum AWG M4i.6631-x8 to generate the multiple frequency tones for driving the AOD. The AWG is programmed in python, so for assembling defect free arrays we developed our own code which can sort an initially loaded array into a defect free configuration. The code is described in detail in appendix B and is also available publicly at <https://github.com/synqs/sorting-tweezers>.

4.7.2 Problems with potassium vacuum

Our original goal was to trap potassium single atoms in the optical tweezers. However, after about a year of achieving dual species MOT and operating the setup, we started facing vacuum issues in the potassium 2D-MOT chamber. The base pressure in the 2D-MOT chamber gradually increased over few months from 10^{-10} mbar to 10^{-8} mbar for the case when the oven heating was completely turned off. This is contrasting with sodium where the pressure always falls down when we turn off oven heating.

To debug the issue, we tried performing a Helium leak test in the 2D-MOT chamber, but could not detect any leak. Another reason we thought of was the pump being clogged with excess potassium, because in the early weeks of searching for a potassium 2D-MOT, we had operated the oven at a higher temperature of about 100°C . So we baked the potassium 2D-MOT setup including the ion-getter pump, followed by a reactivation of the ion-getter pump. After this the pressure reached below 10^{-9} mbar but started rising again slowly over few months. This might be an indication of a damaged pump to resolve which we might have to replace it.

However, since our setup is modular, we were able to completely isolate the potassium 2D-MOT chamber from rest of the vacuum system, by closing its gate valve. Then we decided to try loading sodium atoms into tweezers until the potassium vacuum issue has been fixed.

4.7.3 Experimental details

A schematic of our optical tweezer setup is shown in figure 4.9. To generate the light for optical tweezers, we use a Titanium-Sapphire (Ti-Sa) laser (*Sirah Matisse CR*). We operate the Ti-Sa at a wavelength of 780 nm where we can get a maximum total laser power of 2 W. The Ti-Sa light is passed via an AOD driven with several frequency tones. This results in several beams in the first diffraction order, one for each driving frequency. These frequencies are usually separated by about a MHz or less. The diffracted beams are expanded and imaged onto the back focal plane of our imaging objective via a magnifying telescope comprising 400 mm and 75 mm lenses. Thereby the objective produces an array of focused spots in its focal plane. The advantage of using the imaging objective for focusing the tweezers is that once an atom is trapped in the tweezers, it is automatically in the correct focus for imaging too. This is because due to color correction, the focus for atom imaging wavelength and tweezers wavelength overlap.

To calculate the tweezer size and separation we begin by noticing that the 75 mm lens and the 400 mm lens form a Keplerian telescope as shown in the figure 4.9. The waist of tweezers in the focal plane inside the science chamber depends only on the beam waist size in the back focal plane of the objective. This size is magnified by a factor of $400/75 = 5.3$ compared to the input beam waist in the AOD. The beam waist of the Ti-Sa output is 2 mm, so it will be magnified to 10.6 mm for each frequency tone. The objective will focus these beams into an array of spots with each spot having a diameter w_0 of

$$2w_0 = \frac{4\lambda}{\pi f d} \quad (4.13)$$

where w_0 is the focus waist radius. Putting in the numbers we get $2w_0 = 2.8 \mu\text{m}$. To calculate the distance between the focused tweezers in the science chamber, we use the fact that 400 mm and the objective also form a telescope. So the distance d_{tweezer} between consecutive tweezers is demagnified by a factor of $30.05/400 = 0.075125$ compared to the distance d_1 between the focused spots in the focal plane of 75 mm lens. Since the AOD is kept at the first-focus of the 75 mm lens, the distance d_1 depends only on the deflection angle θ_d of the AOD. The deflection angle θ_d in turn depends on the AOD RF frequency as :

$$\sin \theta_d = \frac{\lambda f_{\text{RF}}}{v_{\text{sound}}} \quad (4.14)$$

where θ_d is the deflection angle, λ is the sound wavelength in the AOD crystal, f_{RF} is the RF frequency driving the AOD and v_{sound} is the speed of sound in AOD crystal. From

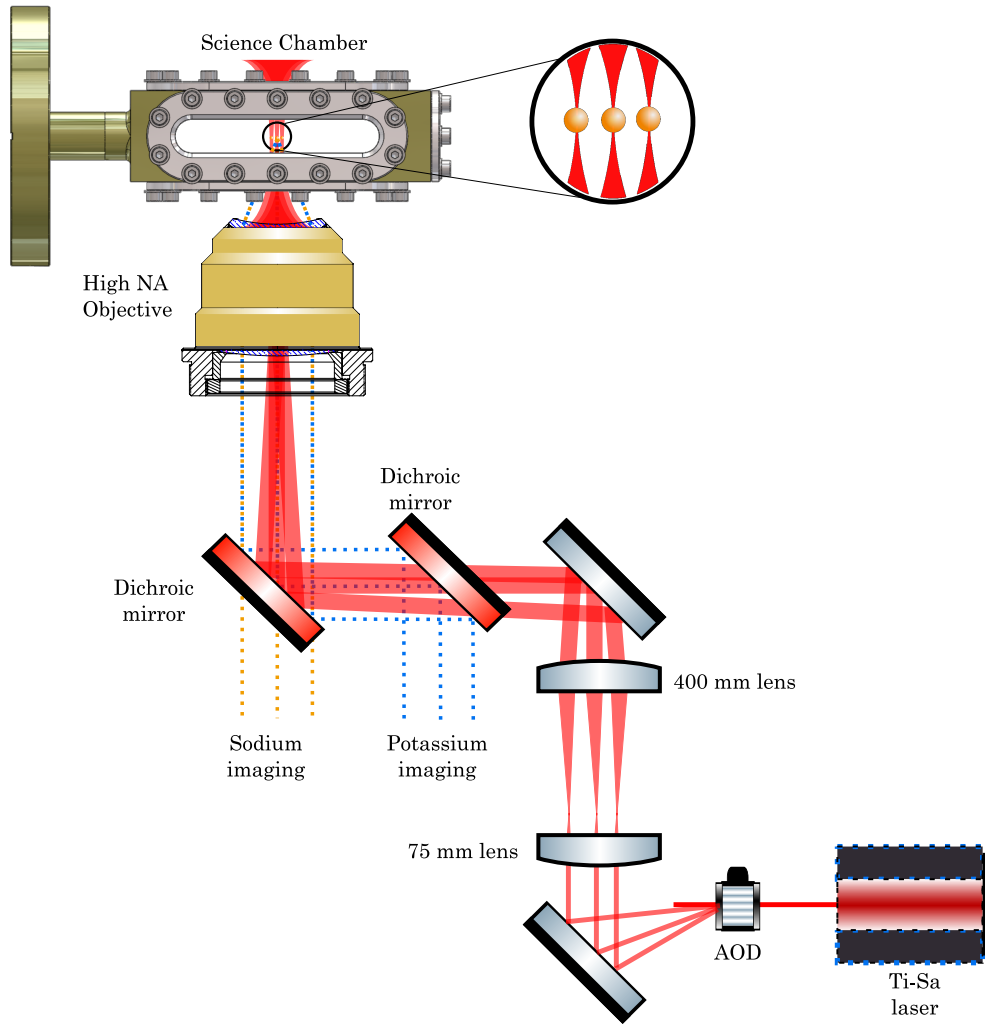


Figure 4.9: Schematic of the tweezer setup.

θ_d , d_1 can be calculated as $d_1 = 75 \text{ mm} * \tan \theta_d$. Putting in the numbers $f_{\text{RF}} = 0.75 \text{ MHz}$ we get $d_1 = 66.88 \text{ }\mu\text{m}$ which results in $d_{\text{tweezer}} = 0.075125 * d_1 = 5.02 \text{ }\mu\text{m}$. These values are also used in appendix section B.2 to explain the sorting protocol. The trap depth for the calculated tweezer size is $270 \text{ }\mu\text{K}$ per 10 mW for sodium.

To align the optical tweezers, we used a process similar to the alignment of optical dipole trap. We use a resonant yellow light such that it destroys the dipole trap and make it pass through the optic axis of the imaging system and the magnifying telescope for the tweezers. After this we overlap the Ti-Sa light with the yellow light path. This gives us a good starting point after which we look for a possible indication of atoms being trapped. For tweaking the alignment, we use the mirrors in the telescope path.

As a first try we have tried to use a single frequency for the AOD for producing one tweezer. We have tried loading this tweezer both from the dark-SPOT MOT and also from the dipole trap. We have observed an indication of the trapping effect of the tweezer in both cases. For this we compare the case with and without the tweezer light.

In case when the tweezer light is on, we see a clear accumulation of atoms around the tweezer focus. Figure 4.10 shows this observation for the case of loading the tweezer from a single arm dipole trap.

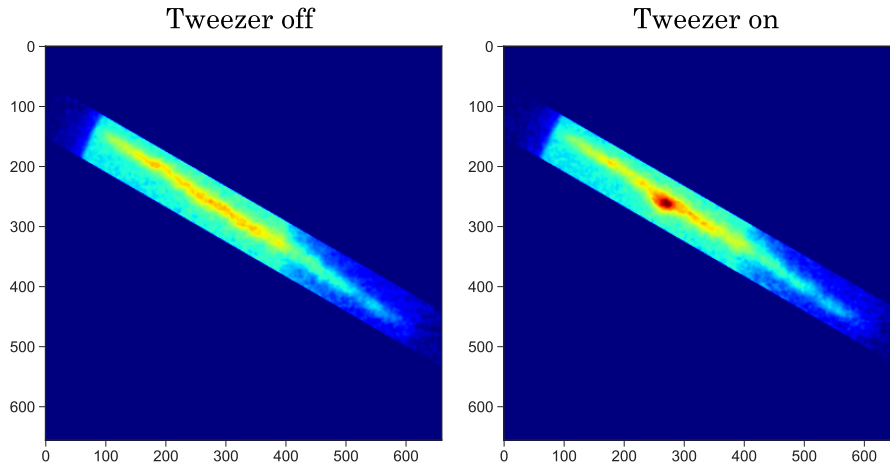


Figure 4.10: Indication of tweezer trapping from a single arm dipole trap: by comparing the cases with and without the tweezer light, we see a clear accumulation of atoms around the tweezer focus. Only a selected region of the full image is shown.

Although we have observed an indication of tweezer trapping, we have not yet been able to sharply image the tweezer spot or resolve different tweezers when driving the AOD with multiple frequencies. This is because our imaging setup is in a demagnifying configuration for sodium. So it does not allow us to resolve individual tweezer sites. The ultimate goal is to trap potassium single atoms in the tweezers and there we plan to use a high magnification in the imaging to resolve individual tweezer sites.

4.8 Summary and future plans

In this part of the thesis, I described our work on developing a new cold atomic sodium-potassium mixture experiment, which has the potential of becoming a versatile platform for quantum simulation. I described the details of our experiment setup, the realization of a dual-species MOT, the first experimental studies, and the efforts in achieving the goal of an ultracold quantum gas mixture. In our setup, we have been able to trap sodium atoms in magnetic trap and optical dipole trap. However, evaporation did not lead to a gain in phase-space density which is due to unfavorable initial conditions in terms of atom number and temperature. So the next steps involve improving these settings and improving the magnetic trap by allowing for the possibility to use higher magnetic gradients. On the potassium side, we have started to debug the 2D-MOT vacuum setup to fix the high pressure issues. Once this is resolved, we will try to simultaneously load sodium and potassium in the magnetic trap and try to cool potassium sympathetically by evaporating sodium. This will allow us to load

4. Towards ultracold quantum mixture

potassium single atoms in optical tweezers and also cooling to the motional ground state.

PART II

Random unitaries with ultracold atoms

Random Unitaries : an overview

In **Part I** of this thesis, I described the development of our ultracold quantum mixture experiment which has the prospect of becoming a general purpose platform capable of quantum simulating a wide range of physics. In the same spirit in this part of the thesis, we develop protocols for realising “random unitaries” in ultracold atom experiments. Random unitaries are very general in their applications i.e. they can be used to extract several properties of quantum systems. In this chapter, I introduce the necessary formalism of random unitaries and mention the examples of using them for probing quantum systems.

The method of using random unitaries to probe a quantum state was first suggested in [56] to extract non-linear functions of density matrices without re-constructing the density matrix itself. For a given system with dimension d , these protocols require sampling unitary matrices uniformly at random from the corresponding unitary group $U(d)$. For this one needs to understand three basic concepts : Haar measure, unitary k -design and frame potential. These three topics are explained in the coming sections.

5.1 Haar measure

In quantum mechanics operations on isolated systems are represented by unitary matrices. These unitary matrices form a unitary group $U(d)$ and Haar measure represents a uniform probability distribution on the unitary group. Mathematically speaking Haar measure [208, 209] is a unitarily invariant probability measure i.e. for any function f defined on $U(d)$ and any unitaries U and V :

$$\int_{U(d)} d\mu f(U) = \int_{U(d)} d\mu f(UV) = \int_{U(d)} d\mu f(VU) \quad (5.1)$$

where $d\mu$ represents the Haar measure on $U(d)$. Being a probability measure, Haar measure is normalised, i.e.

$$\int_{U(d)} d\mu = 1 \quad (5.2)$$

It has been proven that for any Lie group, Haar measure always exists and is unique (upto a scaling constant) [210]. The group $U(d)$ equipped with a Haar measure is also called a circular unitary ensemble (introduced by Dyson [211]).

The ability to engineer Haar random unitaries is an important tool for a diverse set of studies ranging from protocols in quantum information [59–61] and claiming quantum computational advantage [55] to studying fundamental physics like thermalization [212], information scrambling [213] and black hole models [214] etc. Although Haar random unitaries represent quantum randomness, it takes exponential amount of resources to realise them in physical systems [215]. So approximations of the Haar random unitaries are used which produce pseudo-randomness. This approximation is achieved by using a finite set of unitaries called unitary k -design.

5.2 Unitary k -designs

A unitary k -design [216–218] is a set \mathcal{E} of unitaries such that for any polynomial $P_{(k,k)}(U)$ upto order k in matrix elements of U and upto order k in their complex conjugates, the result of averaging $P_{(k,k)}(U)$ over \mathcal{E} is the same as would be obtained by averaging $P_{(k,k)}(U)$ over the Haar measure. Formally a set of unitary matrices \mathcal{E} , is a (exact) unitary k -design (UkD) iff :

$$\frac{1}{|\mathcal{E}|} \sum_{U \in \mathcal{E}} P_{(k,k)}(U) = \int_{U(d)} d\mu P_{(k,k)}(U) \quad (5.3)$$

where $|\mathcal{E}|$ is the cardinality of the set. From this definition, it follows that a unitary k -design reproduces the first k -moments of the Haar measure. Also it is easy to see that a unitary k -design is also a unitary $k - 1$ -design. UkD can be either weighted or unweighted. A weighted UkD is that where not all unitaries have the same weight, while for an unweighted UkD all unitaries have the same weight (i.e. $1/|\mathcal{E}|$). In this thesis we will only consider unweighted UkD .

It has been proven that UkD exist for all orders k in all dimensions d [219–222]. Lower bounds have been obtained on the size of UkD [217, 218]. However, only few constructions for exact UkD are known [216, 223–226]. So the notion of approximate UkD have gained importance. To define an approximate UkD , we use an alternative definition for an exact UkD in terms of distance between the twirling quantum channels [227–229] induced by the UkD unitaries and the Haar measure unitaries. Formally a finite set \mathcal{E} is an ϵ -approximate UkD when [230] :

$$\|\Phi_{Haar}^{(k)} - \Phi_{\mathcal{E}^{(k)}}\|_{\diamond} \leq \epsilon \quad (5.4)$$

where $\|\cdot\|_{\diamond}$ represents the diamond norm [231] and the twirling channels are defined for any operator O as :

$$\Phi_{Haar}^{(k)}(O) = \int d\mu (U^{\dagger})^{\otimes k} O U^{\otimes k} \quad (5.5)$$

and

$$\Phi_{\mathcal{E}}^{(k)}(O) = \frac{1}{|\mathcal{E}|} \sum_{U \in \mathcal{E}} (U^\dagger)^{\otimes k} O U^{\otimes k} \quad (5.6)$$

Several protocols [216–218, 230, 232–244] have been proposed for efficient implementation of approximate UkD . A common principle for generating UkD in these proposals is that of evolution under randomly varying unitaries [238, 245]. Since the dynamics in a complex many-body system results from evolution under a Hamiltonian, it is important to understand the randomness arising from such a unitary evolution [239, 240]. Using the dynamics under a Hamiltonian to generate approximate UkD has also been proposed and experimentally implemented [246–249]. It is also used in our construction of approximate UkD as described in [Chapter 6](#).

5.3 Frame potentials

The definition of UkD given in equation 5.3 or 5.4 is not a very useful way to check if a set of unitary matrices forms a unitary k -design. Instead we can use a quantity called frame potential for this purpose. Given a set of unitary operators \mathcal{E} , the k -th frame potential is defined by [217]:

$$F_{\mathcal{E}}^{(k)} = \frac{1}{|\mathcal{E}|^2} \sum_{U, V \in \mathcal{E}} \left| \text{Tr} \{U^\dagger V\} \right|^{2k} \quad (5.7)$$

where $|\mathcal{E}|$ denotes the cardinality of the ensemble. We denote the k -th frame potential of Haar distributed unitaries by $F_{\text{Haar}}^{(k)}$. A lower bound for frame potential of an arbitrary set \mathcal{E} of unitary matrices is given by [220, 250]:

$$F_{\mathcal{E}}^{(k)} \geq F_{\text{Haar}}^{(k)} \quad (5.8)$$

with equality if and only if \mathcal{E} is k -design. Also if $k \leq d$, $F_{\text{Haar}}^{(k)} = k!$ [250].

Note that for generating UkD , we usually work with a finite set of unitary matrices. So while calculating the sum in equation 5.7, we will have terms for which $U = V$ and the sum will pick up a term $|d|^{2k}/|\mathcal{E}|^2$. The total contribution of such terms for the whole set is $|d|^{2k}/|\mathcal{E}|$. This represents a correction factor which we have to subtract from the calculated frame-potential for the set of unitaries [251].

5.4 Applications of random unitaries

Random unitaries are a key requirement for several protocols in quantum information and quantum computation. Certain types of random states generated using random unitaries have also been proposed to have metrological utility [62]. In this section we focus on using random unitaries for performing randomized measurements in cold atom systems. These measurements provide an alternative to tomography for characterizing general quantum states. They focus on characterization in terms

of measuring interesting properties of the state rather than reconstructing its density matrix. They were first proposed in [56] to extract non-linear functions of the density matrix. An advantage of these methods over tomography is that for most protocols it does not matter which exact unitary is used for the randomized measurement because at the end what counts is the statistical correlation over the entire U^kD . Randomized measurements have also been proposed [252] for improving the efficiency of tomography.

5.4.1 Benchmarking and verifying quantum simulators

In order to trust quantum simulators [253], they must be able to reliably prepare quantum states of interest. Since no device is perfect, a characterization of quantum machines is done to assess the accuracy of their operations. Conventional techniques for this purpose include quantum state/process tomography [254–256], fidelity estimation [257], matrix product state tomography [258] etc. However these protocols suffer from either exponential scaling or not being able to account for state preparation and measurement errors. An alternative technique is that of randomized benchmarking [57, 259–261] which is used to extract an average gate error rate. This technique uses unitary 2-designs (recently higher order version has been proposed and implemented [226]) and is scalable with the system size. It can also account for state preparation and measurement (SPAM) errors. However, a key requirement in this technique is to invert the unitary evolution which might not be so feasible for many body systems.

An alternative approach is to use random unitaries for cross-platform verification [262] of quantum devices. The idea is to use statistical correlations between randomized measurements for estimating the overlap between the state prepared by a quantum simulator and a target state (either from a computer or another quantum simulator). This method although scales exponentially with system size, it still scales better than state tomography protocol for a general case when no assumption can be made about the structure of the state.

5.4.2 Characterizing quantum states

A very important property of quantum states is entanglement [263, 264]. It represents non-local correlations in a system and is an important resource for proving the potential of quantum simulators over classical methods. Characterization of entanglement is usually done via entanglement entropies (like von-Neumann entropy, Renyi entropy etc.). While these entropies serve as entanglement measure for only pure states, they can still detect the presence of entanglement in mixed states [264]. They are also central aspects of laws concerning the evolution of entanglement (say after a quench) [265, 266], its scaling with system size (volume/area law) [267, 268] and dynamical phenomena [269, 270].

Experimentally, entanglement entropies have been extracted by using either tomog-

raphy [271] or by performing interference with multiple copies of the system [272–276]. While tomography protocols suffer from exponential scaling problem, the use of many body interferometry is also not feasible on all experimental platforms. In such cases randomized measurements provide an alternative to extract entanglement entropies [277, 247, 278]. The randomized measurements method still scales exponentially but in a more favorable way compared to tomography. The experimental implementation of using random unitaries to extract entanglement entropies has been already demonstrated [248, 279].

While the concept of entanglement in bi-partite systems is very well understood, the same is not true for the multi-partite case [280, 281]. Protocols based on random unitaries have been proposed to enable detection and certification of multi-partite entanglement detection [282–285]. An important quantity used for multi-partite entanglement characterization is Quantum Fisher information (QFI) [286, 287]. Recently protocols using randomized measurements have been experimentally implemented to extract QFI [288, 289].

5.4.3 Quantum phases and phase transitions

In the Landau-Ginzburg–Wilson formalism [290–292] the ideas of spontaneous symmetry breaking and local order parameter are used for characterization of phases. A phase is usually denoted by the nonzero values of the local order parameter. However, there exist phases like the topological phase [293] which cannot be described in the framework a local order parameter. Such phases are linked to global properties which are not revealed with local observables. In dimensions higher than one, topological entanglement entropy [294, 295] is conjectured to be a distinct feature for topological order. There also exists topological phases which have an intrinsic symmetry [296–298]. For distinguishing different phases in these systems, non-local correlators called as “many-body topological invariants” have been used numerically [299, 300] but have not been measured experimentally.

Using random unitaries, the topological entanglement entropy can be extracted as a correction to the area law of entanglement entropy. For extracting the “many-body topological invariants” protocols based on randomized measurements have been proposed [301, 302].

Quantum phase transitions [303] are driven by quantum fluctuations. They are characterized by non-analyticity of ground state properties as a non-thermal parameter approaches the critical point. The ability of random unitaries to measure overlap between two states can be used to detect quantum phase transitions. The idea is based on proposals [304–306] and experimental demonstration [307] of measuring state overlap between states for slightly different parameters to detect quantum phase transitions. At the phase transition, the state changes rapidly with the changed parameters resulting in a dip in the overlap fidelity and signifying the phase transition.

5.4.4 Quantum chaos and information scrambling

The field of quantum chaos [308, 309] involves understanding quantum dynamics and quantum properties of systems which show chaos in the classical limit. While classical chaos theory and quantum mechanics are separately well understood, formulating a theory of quantum chaos coherent with both ideas is an outstanding challenge. Various indicators for quantum chaos have been proposed like out-of-time-order correlators (OTOC) [310, 311], isospectral twirling [312], spectral form factor (SFF) [313, 314] etc. Random unitaries are closely linked to quantum chaos and it has been shown that to simulate quantum chaos one needs unitary 4-designs or higher order designs [312, 315]. Another recent proposal suggests using randomized measurements to extract SFF as a probe for quantum many-body chaos [316].

A related idea is that of information scrambling [214, 213] wherein locally encoded information spreads across the degrees of freedom of the system. This idea appears in various physical systems, from black holes [317] to quantum many-body systems [318]. Information scrambling can be studied in terms of decay of OTOCs [213, 319]. The time evolution of OTOCs characterizes different types of scrambling regimes : fast scrambling [317] (resembling black hole physics) and slow scrambling [320] (resembling many-body localization).

Previous works have studied information scrambling by extracting OTOCs using sign reversal of Hamiltonian [321–325], many-body interferometry [326, 327], ancillary systems [328] and shuffling operations [329]. But these approaches have some limitations for example they suffer from decoherence issues or are not feasible for other platforms (for example reversing the sign of a many body Hamiltonian is not so easy). There have been theoretical and experimental attempts to resolve these issues [330–335]. Random unitaries provide one such alternative for designing a protocol to extract OTOCs and study their time evolution. The protocol [336] based on randomized measurements does not require implementation of time reversal, many-body interferometry and is also robust to decoherence. This protocol has already been experimentally implemented on different platforms [337, 338].

Generating Random Unitaries

As explained in [Chapter 5](#), random unitaries provide a versatile toolbox to extract several quantities of interest from a physical system. The two types of physical systems we consider are : collective spins of atoms in a BEC and ultracold fermions in a 1D lattice. In this chapter, I describe the protocols we have developed for generating random unitaries on these systems. For collective spins we have developed a protocol based on the experimental results obtained earlier [339] which demonstrated the onset of chaos in collective spin systems subject to periodic coupling. Thus our protocol underlines the connection between chaos (in classical limit) and random unitaries [250]. For fermionic systems, we present a protocol which makes it possible to generate random unitaries for those atomic species (for e.g. fermionic lithium) for whom the requirements of previous proposals [340] might be experimentally very challenging.

6.1 Approximate unitary k -design in collective spins

We consider a collective spin formed by a BEC trapped in a single spatial mode of an optical dipole trap. [Figure 6.1 A](#) shows a schematic of the collective spin. The BEC comprises N indistinguishable two-level atoms, where the two levels are internal $|F, m_F\rangle$ atomic states labelled as 0 and 1. Using the Schwinger representation [341] the atoms can be described as a collective spin with the following operators :

$$L_z = \frac{1}{2} [a_1^\dagger a_1 - a_0^\dagger a_0] , \quad (6.1a)$$

$$L_+ = a_1^\dagger a_0 , \quad (6.1b)$$

$$L_- = a_0^\dagger a_1 , \quad (6.1c)$$

$$L_x = \frac{1}{2} [L_+ + L_-] , \quad (6.1d)$$

$$L_y = \frac{1}{2i} [L_+ - L_-] \quad (6.1e)$$

where the operators a^\dagger and a fulfill the usual bosonic commutation relations. Such systems can be experimentally prepared [342] in two component Bose gases where one measures the occupation $a^\dagger a$ of the two internal atomic states (and thus L_z)

through Stern-Gerlach imaging [109]. The Hamiltonian of each collective spin is :

$$H = \chi L_z^2 + \Delta L_z - \Omega L_x \quad (6.2)$$

The Hamiltonian has three terms : non-linear interaction (χ), detuning (Δ) and linear coupling (Ω). These collective spins have been proposed as viable qudits for universal fault-tolerant quantum computation in ultracold atom systems [52]. The dimension of the qudit can be varied by changing the length of the collective spin (i.e. the number of atoms in the BEC).

The Hilbert space for such a qudit is $\mathcal{H} = \mathbb{C}^d$ with $d = 2\ell + 1 = N + 1$, where ℓ is a positive half-integer, i.e. $\ell \in \{\frac{1}{2}, 1, \frac{3}{2}, \dots\}$ and N is the number of atoms in the qudit. As the computational basis, we choose the eigenstates $|\ell, m\rangle$ of the angular momentum operators L_z and L^2 i.e.

$$L_z |\ell, m\rangle = m |\ell, m\rangle, \quad (6.3a)$$

$$L^2 |\ell, m\rangle = \ell(\ell + 1) |\ell, m\rangle \quad (6.3b)$$

with $m = -\ell, \dots, \ell$.

For the collective spin Hamiltonian, we consider (6.2) with a periodic coupling $\Omega(t)$:

$$H(t) = \chi L_z^2 + \Delta L_z - \Omega(t) L_x \quad (6.4)$$

and

$$\Omega(t) = \Omega_0 (1 + A \sin[\omega(t + t_0)]) \quad (6.5)$$

where A is the modulation amplitude and ω is the modulation frequency. This Hamiltonian realizes the unitary evolution

$$U_{\text{POAT}}(A, t) = \mathcal{T} e^{-i \int_0^t dt H(t)}. \quad (6.6)$$

where \mathcal{T} represents the time-ordered exponential and we call $U_{\text{POAT}}(A, t)$ the periodic-one-axis-twisting (POAT) gate. The Hamiltonian in eq. (6.4) has been used to study the onset of chaos in periodically driven two-component Bose-Einstein condensates [339]. Since the connection between chaos and randomness has been well established [250], eq. (6.4) forms a basis for creating unitary k -designs.

For generating unitary k -designs, we follow an approach motivated from [340]. Each member of the unitary k -design (i.e. a random unitary) is generated by a sequence of η unitary quenches. Each quench is a sequence of two elementary evolutions under (6.4) : the first evolution for time duration t_1 in which $\Omega(t)$ is modulated according to (6.5) and the second evolution for time duration t_2 in which $\Omega(t) = 0$. The total time for each quench is a constant denoted as t_{tot} . For each quench, t_1 is uniformly sampled from the interval $[0, t_{\text{tot}}]$ and t_2 is calculated as $t_2 = t_{\text{tot}} - t_1$. The values of other parameters used for generating unitary k -designs are similar to those in

[339] with $N = 500$, $N\chi = 2\pi * 27\text{Hz}$, $t_{\text{tot}} = 22.2\text{ms}$, $A = 0.85$, $\omega = 848.23\text{Hz}$, $\Omega = 242.35\text{Hz}$, $\Delta = -26.65\text{Hz}$, $t_0 = 4.4\text{ms}$. The choice of these parameters is made so that the collective spin exhibits chaotic dynamics classically. The protocol for generating random unitaries is pictorially depicted in figure 6.1 B.

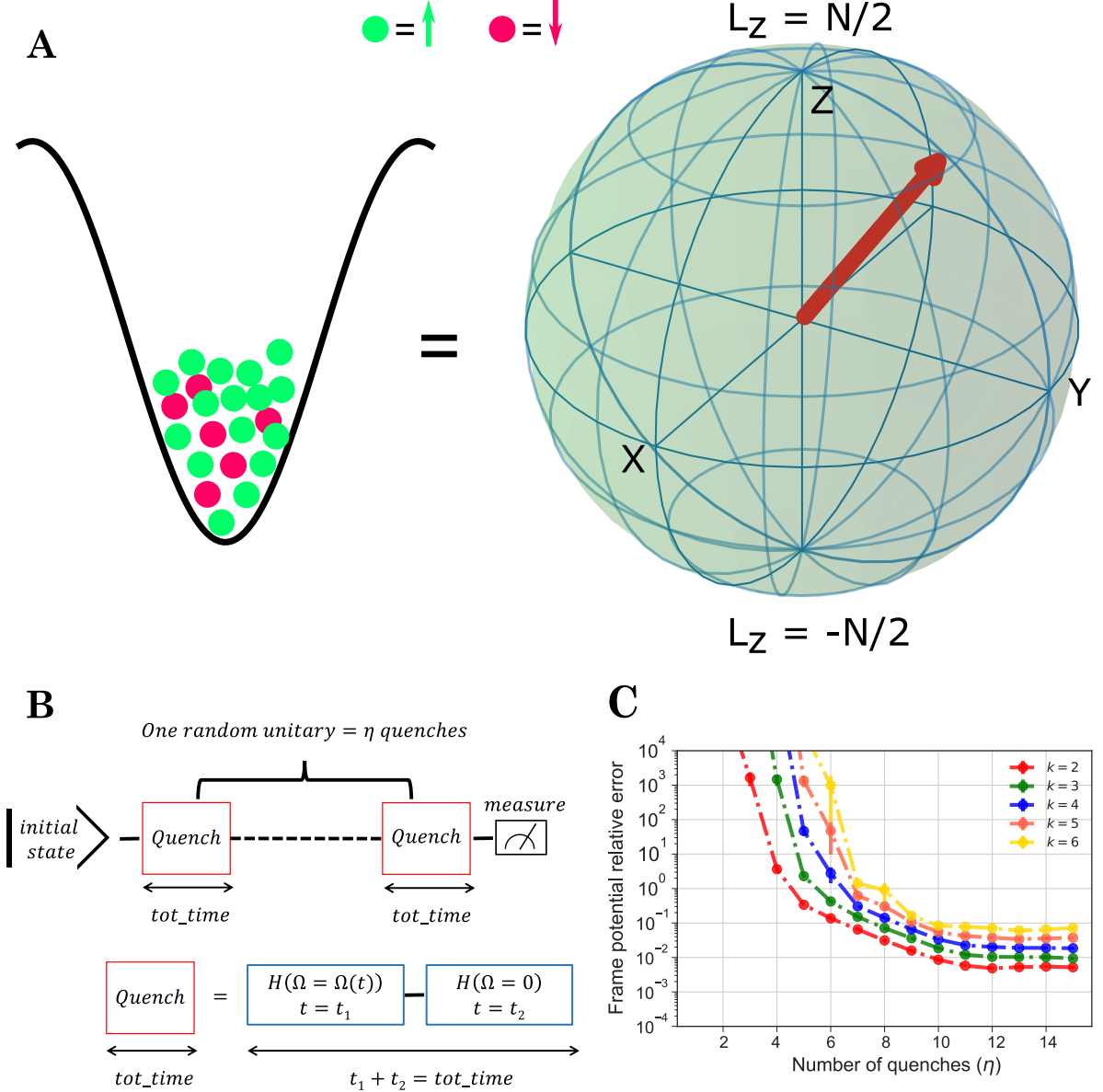


Figure 6.1: Unitary k -designs for a collective spin qudit. **A:** A collective spin qudit is formed by two level atoms in a BEC. **B:** Protocol for generating k -designs for single qudit, t_1 is uniformly sampled from the interval $[0, t_{\text{tot}}]$ and t_2 is calculated as $t_2 = t_{\text{tot}} - t_1$. **C:** Relative error on the k -th frame potential as a function of number of quenches for $N = 500$. The whole process is repeated 100 times to get average values and statistical error bars.

We generate an ensemble of 500 random unitaries for each value of η . For benchmarking our generated k -designs we calculate $F_{\mathcal{E}}^{(k)}$ mentioned in eq. (5.7) for each ensemble taking into account the correction factor of $d^{2k}/500$ arising from finite size of our ensembles. Since $F_{\mathcal{E}}^{(k)} \geq k!$ with equality only for an exact k -design, we calculate the relative error on $F_{\mathcal{E}}^{(k)}$ as $|F_{\mathcal{E}}^{(k)} - k!|/k!$ and plot it as a function of number of quenches η .

As shown in figure 6.1 C, we observe the convergence to unitary k -designs for increasing number of quenches. The whole process is repeated 100 times to get statistical error bars.

From figure 6.1 C, it is clear that for less than 10% relative error on upto 6-th frame potential, a minimum of 10 quenches are required. This means the time required for one random unitary is ≈ 200 ms. This time can be reduced by increasing the slowest energy scale which is the interaction energy $N\chi$. As shown in [342], $N\chi$ increases with increasing atom number. So for longer collective spins we expect the time scales to be smaller.

6.2 Approximate unitary k -design for 1D-fermion system

We consider a 1D system of fermions in a lattice (with L sites) whose dynamics is governed by the Hamiltonian:

$$\begin{aligned}
 H_{FH} = & - \sum_{i=1,\sigma}^{L-1} J_{i,i+1} (c_{i,\sigma}^\dagger c_{i+1,\sigma} + \text{h.c.}) - \sum_{i=1,\sigma}^{L-2} J'_{i,i+2} (c_{i,\sigma}^\dagger c_{i+2,\sigma} + \text{h.c.}) \\
 & + \sum_{i=1}^L U n_{i,\uparrow} n_{i,\downarrow} + \sum_{i=1}^L \mu_i (n_{i,\uparrow} + n_{i,\downarrow})
 \end{aligned} \tag{6.7}$$

where h.c. denotes the hermitian conjugate. The Hamiltonian has four terms : nearest neighbor tunneling (J), next-nearest neighbor tunneling (J'), on-site interaction (U) and chemical potential (μ). For $J' = 0$ eq. (6.7) reduces to the standard Fermi-Hubbard Hamiltonian. The relative strength of J' and J can be tuned by changing the lattice depth, as shown in [343] for the Bose-Hubbard model. A schematic of the Fermi-Hubbard system is shown in figure 6.2 A.

Each of the four terms in equation (6.7) preserves the number of spin up/down particles individually (and therefore the total particle number N and the projection S_z). The existence of these symmetries leads to separation of the full Hilbert space into sectors labelled by the conserved quantum numbers (N, S_z) and results in a block-diagonal Hamiltonian in the Fock basis. For L lattice sites, N_\uparrow spin up particles and N_\downarrow spin down particles the dimension of the (N, S_z) sector is given by $\dim(N, S_z) = \binom{L}{N_\uparrow} \binom{L}{N_\downarrow}$ where $N = N_\uparrow + N_\downarrow$ and $S_z = \frac{1}{2}(N_\uparrow - N_\downarrow)$.

In experiments, one measures the spin resolved occupancy of each site. Consider for example a 1D lattice with $L = 3$, $N = 4$, $N_\uparrow = 2$ and $N_\downarrow = 2$. So we are in the sector $(N, S_z) = (4, 0.0)$. To every lattice site we associate two binary digits i.e. each digit is either 0 or 1. The first digit is 0(1) if $N_\uparrow = 0(1)$ for that site and the second digit is 0(1) if a $N_\downarrow = 0(1)$ for that site. With this representation, all possible outcomes for the $(4, 0)$ sector are :

$$B = \{001111, 011011, 011110, 100111, 101101, 110011, 110110, 111001, 111100\} \tag{6.8}$$

The set B forms the basis for creating the matrices of different terms in equation (6.7). As an explanation of the representation, the first element of B represents the outcome when the first lattice site is empty while both the second and third lattice sites have one particle each of spin up and spin down.

For generating unitary k -designs we follow an approach motivated from [340]. However, unlike [340] we do not consider the chemical potential μ_i to be spin dependent. It makes the generation of unitary k -designs experimentally easier for some fermionic species e.g. lithium-6. To understand this we start by noting that the energy scales for chemical potential is usually a few kHz. Proposals for creating spin-dependent terms in the Fermi-Hubbard model usually involve multiple lasers and Raman transitions [344, 345]. However Raman dressing of atoms leads to a heating rate which depends on the energy scale for chemical potential and the fine-splitting energy (i.e. separation of D1 and D2 transitions). For lithium-6 where the fine-splitting is very small (10 GHz), the ratio of the chemical potential to the heating rate is ~ 100 [346]. For $\mu = 10$ kHz, the heating rate would be 100 Hz which is significantly high. So one would like to design a method without the need for spin-dependent terms in the Fermi-Hubbard Hamiltonian. This is what we have achieved with our protocol.

Each member of the unitary k -design is generated by a sequence of η unitary quenches. Each quench is a sequence of two elementary evolutions under (6.7): the first evolution for time duration t_1 in which μ_i are uniformly sampled from the interval $[-10 \text{ kHz}, 10 \text{ kHz}]$ and the second evolution for time duration t_2 in which $\mu_i = 0, \forall i$. For each quench, t_1 and t_2 are uniformly sampled from the interval $[1 \text{ ms}, 4 \text{ ms}]$. The values of other parameters used for generating k -designs are $J = 1 \text{ kHz}$, $J' = 0.1 \text{ kHz}$ and $U = 0.5 \text{ kHz}$. The protocol for generating random unitaries is pictorially depicted in figure 6.2 B.

We generate an ensemble of 500 random unitaries for each value of η . For benchmarking our generated k -designs, we calculate $F_{\mathcal{E}}^{(k)}$ mentioned in eq. (5.7) for each ensemble taking into account the correction factor of $d^{2k}/500$ arising from finite size of our ensembles. These calculations are performed sector-wise for various (N, S_z) sectors because the dynamics of the (6.7) is constrained to a (N, S_z) sector. Since $F_{\mathcal{E}}^{(k)} \geq k!$ with equality only for an exact k -design, we calculate the relative error on $F_{\mathcal{E}}^{(k)}$ as $|F_{\mathcal{E}}^{(k)} - k!|/k!$ and plot it as a function of number of quenches η . As shown in figure 6.2 C, we observe the convergence to unitary k -designs for increasing number of quenches. The whole process is repeated 100 times to get statistical error bars.

The performance of our method in various sectors is shown in figure 6.3. We observe that our method works well in all sectors except $(N, S_z = 0), \forall N > 1$ i.e. when $N_{\uparrow} = N_{\downarrow}$. However, we do not understand what causes the observed failure. A possible reason could be existence of some symmetry in these sectors which prevents the dynamics from going chaotic and thus does not cause a decrease in frame potential.

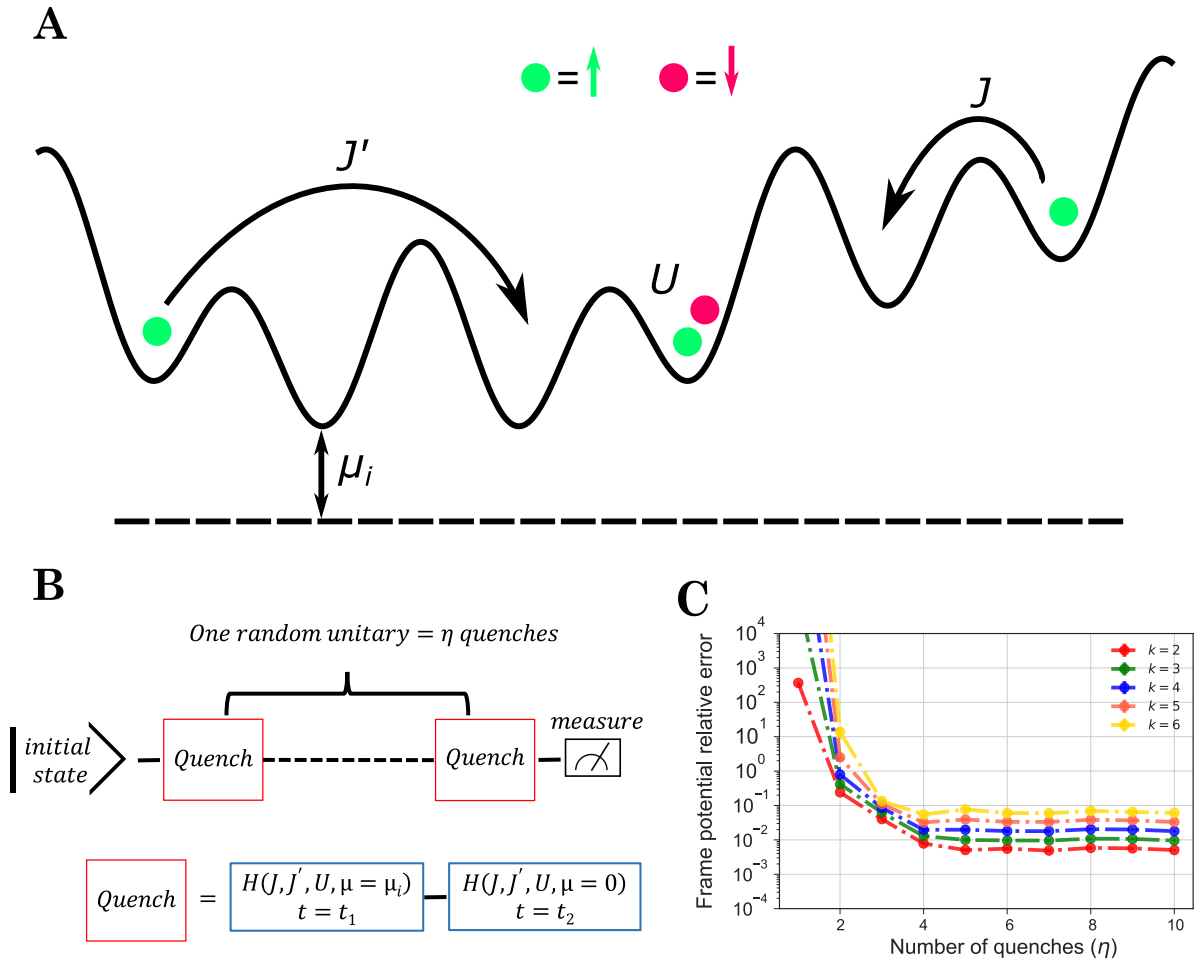


Figure 6.2: Unitary k -designs for fermions. **A**: Schematic of a Fermi-Hubbard system. **B**: Protocol for generating unitary k -designs for fermions. **C**: Relative error on the k -th frame potential as a function of number of quenches for the configuration $L = 6$, $(N, S_z) = (4, 1.0)$. The whole process is repeated 100 times to get average values and statistical error bars.

6.3 Summary

In this part of the thesis, I described our work on developing protocols for implementation of random unitaries in bosonic and fermionic cold atom systems. For both cases our protocols focus on experimental feasibility. In the bosonic case, this is achieved by realizing random unitaries using the experimental protocol already implemented for studying onset of classically chaotic dynamics in a collective spin system. As next steps, apart from improving timings, we can try to extend the protocol to the case of multiple collective spins. This can be done by alternate application of the already proposed random unitaries for a single collective spin and entangling operation between different collective spins. This is a work in progress and we have already seen preliminary results that the method works for multiple collective spins.

For the fermionic case, our protocol can help to relax the experimental requirements on spin-dependent lattices thus making it useful for fermionic species like lithium-

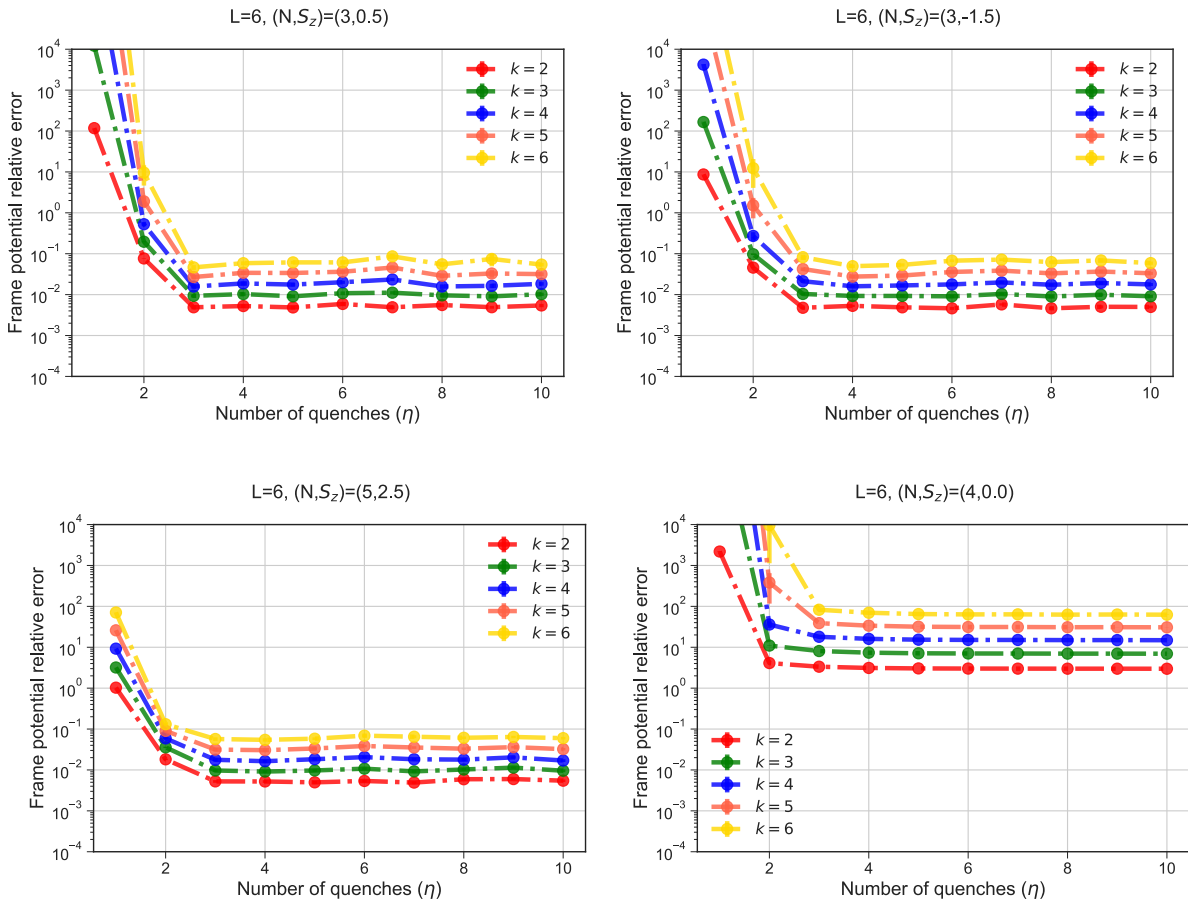


Figure 6.3: Unitary k -designs for fermions in various sectors.

6 where it is hard to engineer a spin-dependent lattice. Our protocols works on all sectors of the Hilbert space except those with $S_z = 0$, and we have to develop a better understanding on what causes the observed behaviour in these sectors. Finally we can try to extend our protocol to two dimensional lattices which will increase its usefulness for state-of-the-art experimental setups.

PART III

**Qlue: a web interface for cold atom
backends**

Web technology for cold atoms

As mentioned in section 2.6.1, there exist use cases where *labscript* misses architectural requirements and one would like to include *labscript* into a larger framework. In this chapter, I will explain in detail those scenarios and how we developed a general solution for the problem¹. The idea stems from the desire for unifying experiment execution on different platforms for quantum simulation/computation. These platforms can be based on different technologies (e.g. neutral atoms, ions, superconducting qubits, photonic systems etc.) and have different software for experiment control.

In this diversity there is a unifying feature for different experiments, they essentially act as a backend for executing some instructions. On adopting this picture, we realise that we become totally agnostic to the details of how the instructions are executed on a hardware if we can provide a general higher level description of the experiments. A very suitable high-level description for experiments on different platforms is to think of these experiments as quantum circuits. In this picture any experiment essentially has three steps : state preparation, unitary evolution under given Hamiltonian(s) and measurement. In fact we are not just limited to experiments anymore and can even include theoretical simulator backends which execute the quantum circuit instructions on a computer.

For providing software support for a given backend to use the language of quantum circuits, two types of libraries are required : one which allows to write quantum circuits for the backends using some kind of quantum circuit framework and a second one for parsing the quantum circuits into a format which can be run on the backends. For the first one, there already exist quantum circuit frameworks like *Qiskit*, *Pennylane* etc. which provide a comprehensive support for writing quantum circuits. For the second one, custom code has to be written which will depend on the details of the particular backend.

Another important point is that we assume that the user who writes the quantum circuits does not necessarily have local access to the backends. So the user submits

¹Some parts of this chapter overlap with our online documentation at <https://synqs.github.io/qlue/> and <https://labscript-qc.readthedocs.io/>

quantum circuits to the backend remotely over the internet. We use the word *job* for the submission by the remote user. We immediately realise that we also need a service which can act as the central point for various remote users to submit their jobs or fetch the result/status of their previously submitted jobs. For security reasons this service should provide robust user management and authentication.

A framework which can provide all the features described in the paragraphs above has several advantages :

- Using the circuit language for quantum simulation experiments, the same circuit can be executed on different backends. This can allow to benchmark the performance of quantum simulators against each other.
- Describing experiments as backends can be used to build a simulator backend for the experiment which mimics the experiment upto certain Hilbert space dimension as allowed by the computing resources. This can be helpful for an experiment-theory collaboration where theory colleagues can try new ideas for the experiment on the simulator backend. After testing a new research idea on the simulator it can be easily implemented on the experiment using the same circuit language.
- Setting up a simulator backend can be really helpful in encouraging student participation in cold atom research. The ability for a remote user to submit jobs and get results back over the internet has partly been the reason for the huge popularity of quantum computing. A similar infrastructure can be implemented for research and teaching too.

We have developed a web interface called *qlue*² which allows remote users to submit their jobs to remote cold atom backends. In the following sections, I explain the architecture of *qlue* in detail and give examples of several backends we have connected to this interface. Then I present few examples wherein we have used the architecture to run some jobs on a real experimental backend and on simulator backends to reproduce the experimental observations for different systems.

7.1 Qlue architecture

The architecture of *qlue* has three components : the remote client, the server and the backends. The overall architecture is shown in figure 7.1. These three components use various software libraries/framework to communicate with each other. The code for all components of *qlue* is organised in various GitHub repositories as shown in table 7.1. Most of these repositories are public and can be downloaded freely. In the following parts of this chapter these repositories will be referred to by their names

²<https://github.com/synqs/qlue>

mentioned in table 7.1. In order to use *qlue* architecture for a new backend, code has to be written on all three components. The existing code for already added backends should provide enough examples on how to structure the new code.

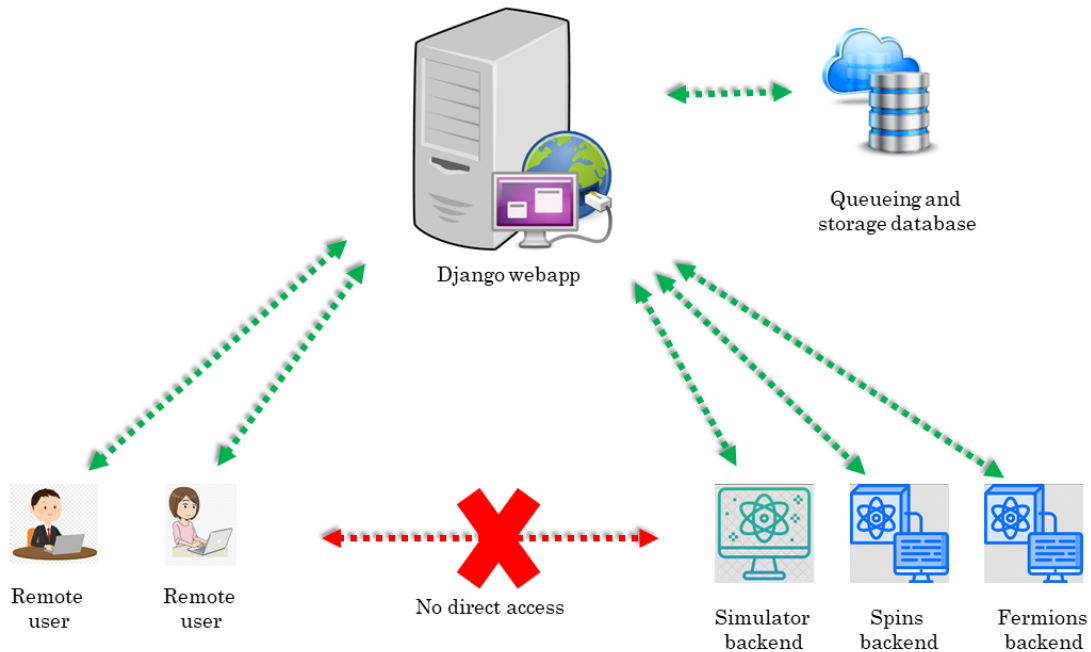


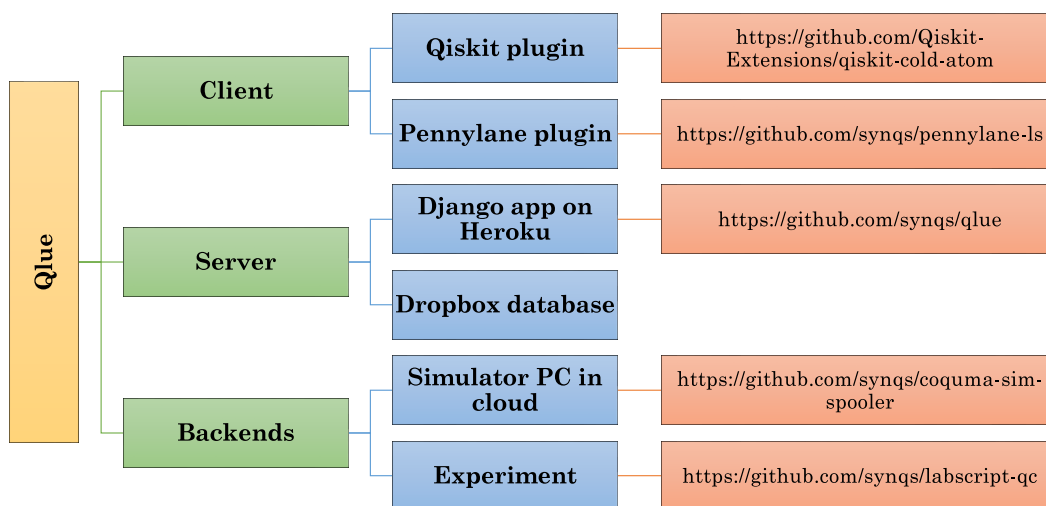
Figure 7.1: Qlue architecture.

In order to establish the communication between different components of *qlue*, a standard file format has to be decided. We have decided to use the *JSON* (JavaScript Object Notation) format due to convenience in transferring it over the internet. Many programming languages provide support for the *JSON* format which also makes the architecture of *qlue* more language agnostic. *JSON* is a data format which is very similar to a python dictionary and consists of key-value pairs.

In our case for unambiguous communication, it is very important that all *JSON* files follow a well defined layout/schema. Therefore we have decided the schema for all *JSON* files that are transferred in the workflow of *qlue* architecture. These are described in the section 7.1.1. We enforce the practice of following the decided *JSON* schemas by sanity checking the `job_JSON` submitted by the client. It reduces the chance for malicious code execution by an adversary.

7.1.1 The JSON schemas

There are three important JSON schemas. The first one is the schema for the backend to describe its capabilities, operations supported etc. The second one is for the JSON generated for each job submitted by the client. Finally the last one is for the result generated by the backend. All three of these are listed below.



GitHub repository	URL
qiskit-cold-atom	https://github.com/Qiskit-Extensions/qiskit-cold-atom
pennylane-ls	https://github.com/synqs/pennylane-ls
qlue	https://github.com/synqs/qlue
coquma-sim-spooler	https://github.com/synqs/coquma-sim-spooler
labscript-qc	https://github.com/synqs/labscript-qc

Table 7.1: Qlue code organization in various GitHub repositories and their URLs. Note that `coquma-sim-spooler` is currently a private repository.

7.2 The client

A client is a remote user who writes quantum circuits in the his/her favorite quantum circuit framework (*QisKit*, *PennyLane*, something else). These circuits are then compiled into *JSON* files using plugins for the quantum circuit framework. The *JSON* files are sent over the internet to a remote server which queues them for parsing/execution on the backend. The backend can be a real cold atom machine or a simulator running on a computer. Communication with the server is only possible if the client has authorized credentials. To get the credentials, the client has to create an account at <https://qsimsim.synqs.org>.

7.2.1 QisKit and PennyLane plugins

The plugin for compiling Qiskit circuits to *JSON* files is available at `qiskit-cold-atom` repository. Similarly the plugin for compiling a pennylane circuit into *JSON* is available at `pennylane-ls` repository. Note that these plugins offer several backends. Each backend is a device with its own operations. For each device, the plugin converts a quantum circuit into a *JSON* which respects the schema expected by the corresponding backend. The plugin also sends these *JSON* files to the server via HTTP requests. The GitHub repositories for these plugins also have some examples on how to write circuits for our already existing backends.

```

1 {'backend_name': 'atomic_mixtures',
2   'backend_version': '0.0.1',
3   'n_qubits': 2,      # number of wires
4   'atomic_species': ['Na', 'Li'],
5   'basis_gates': ['delay', 'rx'],
6   'gates': [
7     {'name': 'delay',
8      'parameters': ['tau', 'delta'],
9      'qasm_def': 'gate delay(tau, delta) {}',
10     'coupling_map': [[0, 1]], # on which components/wires can the instruction
                                apply
11     'description': 'evolution under SCC Hamiltonian for time tau'},
12     {'name': 'rx',
13      'parameters': ['theta'],
14      'qasm_def': 'gate rx(theta) {}',
15      'coupling_map': [[0]], # on which components/wires can the instruction apply
16      'description': 'Rotation of the sodium spin'}],
17   'supported_instructions': ['delay', 'rx', 'measure', 'barrier'],
18   'local': False,        # backend is local or remote (as seen from user)
19   'simulator': False,   # backend is a simulator
20   'conditional': False, # backend supports conditional operations
21   'open_pulse': False,  # backend supports open pulse
22   'memory': True,       # backend supports memory
23   'max_shots': 60,
24   'coupling_map': [[0, 1]],
25   'max_experiments': 3,
26   'description': 'Setup of an atomic mixtures experiment with one trapping site and
                                two atomic species, namely Na and Li.',
27   'url': 'http://url_of_the_remote_server',
28   'credits_required': False,
29   'online_date': datetime.pyi = hopefully soon,
30   'display_name': str = None}

```

Table 7.2: Example JSON for backend capabilities

```

1 {
2   experiment_id(str): {
3     'instructions': [
4       (inst_name(str), wires(List[int]), params(List[float])),
5     ],
6     'shots': int,
7     'num_wires': int,
8     'seed': int,
9     'wire_order': str,
10  }
11 }

```

Table 7.3: JSON schema for user submitted jobs.

That's all what is required on the client side. Basically choose one of these frameworks and start submitting the jobs. If the client wants to use a different quantum circuit framework, then appropriate code has to be written for compiling quantum circuits of that framework into *JSON* files which follow the schema expected by the backend (for e.g. 7.3). This is also the case when a new backend is added as a device to either Qiskit or PennyLane plugins. In any case the existing code can be used a guide to write

```
1 {
2   "backend_name": str,
3   "backend_version": str,
4   "job_id": str,
5   "qobj_id": str,
6   "success": bool,
7   "header": dict, # must be JSON serializable
8   "results": list[
9     {
10      "header": dict, # must be JSON serializable
11      "shots": int,
12      "success": bool,
13      "meas_return": str,
14      "meas_level": int, # most likely always 1 or perhaps 0
15      "data": {
16        "memory": list
17      }
18    }
19 ],
20 "experiments": list[],
21 }
```

Table 7.4: JSON schema for backend results

new code.

7.3 The server

The server provides an access point to the remote client where the client can submit new jobs or fetch the result/status of previously submitted jobs. For this the client needs to make appropriate HTTP requests to the appropriate URLs. When the server receives these HTTP requests, depending on which URL they were sent to, a particular action is taken and then the reply is sent back to the client. It is clear that the server needs to run some kind of code which does the following tasks : listen for HTTP requests from the clients, provide URLs such that each URL is associated with a particular action and respond back to the client. We achieve these tasks by a *Django* web app running on a *Gunicorn* server. To make the web app available across the internet, it is hosted on the *Heroku* platform.

In the following subsections the server is explained in three parts : Django web app with URLs, the view functions and the Dropbox database.

7.3.1 Django web app with URLs

Django is a Python-based free and open-source framework for making web apps. It uses the Model-View-Template architecture:

- Model : Build databases from classes with the help of Object Relational Mapper (ORM).

- View : The function which is executed whenever a particular URL receives an HTTP request. For every URL, Django associates a corresponding view function. A view function is similar to a usual python function.
- Template : HTML/CSS code for inserting web elements in a HTML document.

As mentioned earlier, for the client, communicating with the server essentially boils down to sending correct HTTP request to the correct URL. On the server side Django will execute the view function corresponding to that URL. The structure of a general URL looks like :

```
server_domain/requested_backend/appropriate_view_name
```

So for example if server_domain=https://qsimsim.synqs.org/api/, and we want to use get_job_status view for the fermions backend i.e. requested_backend=fermions and appropriate_view_name=get_job_status. The complete URL will look like :

```
https://qsimsim.synqs.org/api/fermions/get_job_status/
```

In our case server_domain=https://qsimsim.synqs.org/api/ always. The names of all views are given in subsection 7.3.2. The backends provided at the moment are : singlequdit, multiqudit, fermions . As described above, this information can be combined to figure out all currently valid URLs.

7.3.2 The view functions

A view function executes a specific code for each HTTP request coming at the URL linked to that view function. A description of various view functions is given below:

- The get_config view : This function returns a *JSON* dictionary containing the backend capabilities and details. At the moment this is relevant only for the *Qiskit* plugin as the *pennylane* plugin does not make use of it.
- The post_job view : This function extracts the *JSON* dictionary describing a potential experiment from a HTTP request. The extracted job_JSON is dumped onto *Dropbox* for further processing. It then responds with another *JSON* dictionary which has a job_id key. This job_id is important to query the server for results of the job later on. A typical *JSON* response from the server has the following schema: `{'job_id': 'some string', 'status': 'some string', 'detail': 'some string', 'error_message': 'some string'}`
- The get_job_status view : This function extracts the job_id of a previously submitted job from a HTTP request. It responds with a *JSON* dictionary describing the status.

- The `get_job_result` view : This function extracts the `job_id` of a previously submitted job from a HTTP request. If the job has not finished running and results are unavailable, it responds with a *JSON* dictionary describing the status. Otherwise it responds with a *JSON* dictionary describing the result. The formatting of the result dictionary is described in table 7.4.
- The `get_user_jobs` view : This function returns a *JSON* dictionary containing all the finished jobs for a user at a particular backend.
- The `get_next_job_in_queue` view : This function is not available to regular users. It is a very special function and is reserved for use only by the backends. When backends ask the server for the next job it should work on, the server replies with a *JSON* dictionary with details of the next job to be executed. From this, the backend knows exactly where the `job_JSON` file is stored on Dropbox. It fetches the `job_JSON` and starts to process it.

We will frequently use the term status dictionary and result dictionary. These are just *JSON* files which are stored on the Dropbox like `job_JSON`. When a view reads these files, it converts the data in them to a python dictionary and sends it to the client or modifies the dictionary further before saving it back to the *JSON* file.

7.3.3 The Dropbox database

We use Dropbox to store all our *JSON* files. For reading from and writing to Dropbox, the Dropbox python API ³ functions are used. Using Dropbox as a storage has several benefits:

- We immediately implement asynchronous job management wherein Dropbox acts like a queue. It is required because the rate at which the remote clients submit jobs is very different from the rate at which the backends can process them. So the server dumps the job coming from the remote client onto Dropbox and whenever the appropriate backend is free, it will process the job.
- Dropbox also serves as a database storage for various *JSON*, like `job_JSON`, `result_JSON`, `status_JSON` etc. It gives us 2 GB free storage which is a lot considering price of commercial database alternatives.
- It allows us to unify the workflow for both simulator and real machine. It does not matter what the backend is, the workflow is the identical. Different backends only differ in how exactly they process a job which is determined by the code in their corresponding `spooler.py` files.

³<https://github.com/dropbox/dropbox-sdk-python>

Note that Dropbox can be replaced with any other storage service (like Amazon S3, Microsoft azure storage, Google cloud storage etc.) which allows a user to read and write content using a python API. For this one would need to implement four basic functions for accessing the cloud storage provider. The details of these four functions are given in `storage_providers.py` file with an example implementation for Dropbox. A new storage service can have its own class inheriting from the base class (just like the Dropbox class) and override the base functions.

An important feature for properly using Dropbox as a storage and queue is the folder structure for storing *JSON* files on Dropbox. The Dropbox folder structure is shown in figure 7.2. If one plans to replace Dropbox with other storage service, then one needs to implement a folder/naming structure similar to figure 7.2. This structure is also connected to the file name of a given *JSON*.

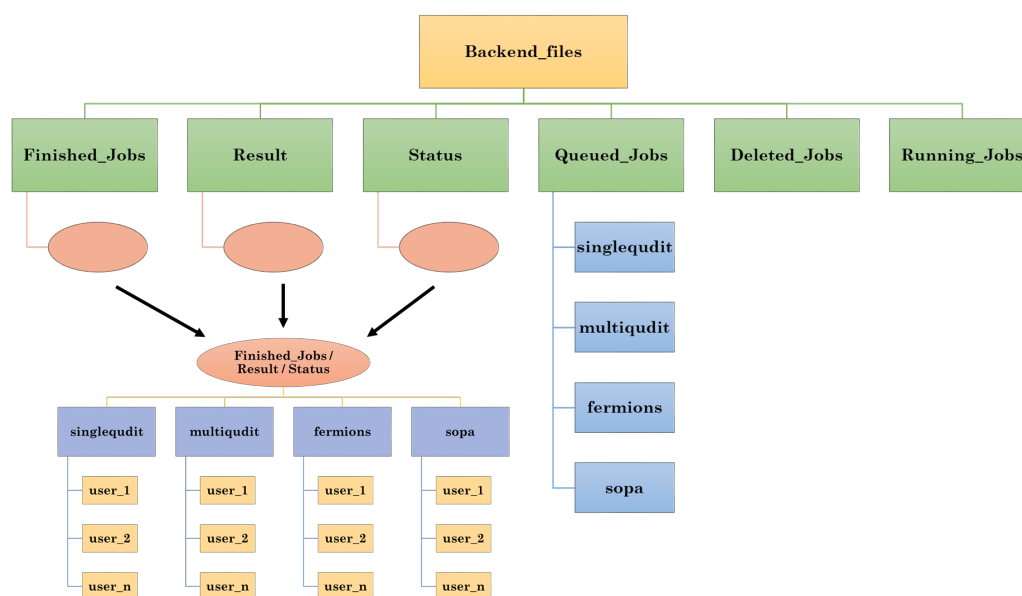


Figure 7.2: Dropbox folder structure. Each box is a folder. The pink ovals show that the sub-folder structure is identical for some folders. This structure is shown separately and marked with arrows.

7.3.4 An example workflow

To understand how various modules described so far are used, we consider an example workflow. In summary this subsection describes the journey of a `job_JSON` file submitted by the client and other *JSON* files created as a consequence. Lets say a remote client named `user_1` submits a job to the `singlequdit` backend. This means `user_1` sent a POST request to the following URL :

```
https://qsimsim.synqs.org/api/singlequdit/post_job/
```

The server will immediately save the *JSON* to Dropbox at `Backend_files/Queued_Jobs/singlequdit/job-20210906_203730-singlequdit-user_1-1088f.json`. The *JSON* file name has special meaning as will be explained later. The file

is stored here temporarily before backend processes it. At the same time the server will reply to the user with a `job_id` response which will look something like :

```
{'job_id': '20210906_203730-singlequdit-user_1-1088f', 'status': 'INITIALIZING', 'detail': 'Got your json.', 'error_message': 'Got your json.'}
```

This dictionary is also saved as a `status_JSON` at `Backend_files/Status/singlequdit/user_1/status-20210906_203730-singlequdit-test-1088f.json`. Note the name is quite similar to the `job_JSON` except a `status-` prefix.

The `job_id` key-value has lot of information. It has the UTC date and time of creation of the job `20210906_203730` which means it was created on 6 September 2021 at 20:37:30 PM UTC time. The `job_id` also has the user name who created this job i.e. `user_1` and the backend where this job is supposed to be executed i.e. `singlequdit`. At the end the job has some random alpha-numeric string of 5 characters.

On the backend side, it will query the server for the next job it should work on. Also let us suppose the backend is querying about the next job for `singlequdit` backend. The server looks for the file list in the directory `Backend_files/Queued_Jobs/singlequdit/`. It will choose the first created file from that list. Lets say this is the file `job-20210906_203730-singlequdit-user_1-1088f.json`. Now the server will move this file from `Backend_files/Queued_Jobs/singlequdit/job-20210906_203730-singlequdit-user_1-1088f.json` to `Backend_files/Running_Jobs/job-20210906_203730-singlequdit-user_1-1088f.json` and respond to the backend with a `job_msg_dict` which looks like :

```
{"job_id": "20210906_203730-singlequdit-user_1-1088f", "job_json": "Backend_files/Running_Jobs/job-20210906_203730-singlequdit-user_1-1088f.json"}
```

From this, the backend knows exactly where the `job_JSON` file is stored on Dropbox. It fetches the `job_JSON` and starts to process it. For processing the job, the spooler begins by sanity-checking the *JSON* for correct schema. If the `job_JSON` fails this check the file is moved to `Backend_files/Deleted_Jobs/job-20210906_203730-singlequdit-user_1-1088f.json`. The `status_JSON` is also updated by the spooler to:

```
{'job_id': '20210906_203730-singlequdit-user_1-1088f', 'status': 'ERROR', 'detail': 'Got your json.; Failed json sanity check. File will be deleted. More details.' 'error_message': 'Got your json.; Failed json sanity check. File will be deleted. More details.'}
```

From the `status_JSON` dictionary, the user is automatically informed about the details of why the error happened.

If however, the `job_JSON` passes sanity checking, then it is executed. The backend

goes through the instruction list and creates the appropriate circuit and calculates the end result. Then it generates the given number of shots and formats everything into the result dictionary. The schema of the result dictionary is given in table 7.4. Then the backend will upload the `result_JSON` to `Backend_files/Result/singlequdit/user_1/result-20210906_203730-singlequdit-user_1-1088f.json`. It will also move the `job_JSON` from `Backend_files/Running_Jobs/job-20210906_203730-singlequdit-user_1-1088f.json` to `Backend_files/Finished_Jobs/singlequdit/user_1/job-20210906_203730-singlequdit-user_1-1088f.json`. Finally the backend will update the `status_JSON` to:

```
{'job_id': '20210906_203730-singlequdit-user_1-1088f', 'status': 'DONE', 'detail': 'Got your json.; Passed json sanity check; Compilation done. Shots sent to solver.', 'error_message': 'Got your json.; Passed json sanity check; Compilation done. Shots sent to solver.'}
```

This completes the execution of the job and the results are now available. To fetch the results the client has to make an appropriate HTTP request (using the plugins) to the `get_job_result` view.

7.4 The backends

This section describes how different backends execute the jobs submitted by the clients. The backend can be a simulator running on a computer or a real cold atom machine. It runs files which are responsible for executing `job_JSON` and updating `status_JSON` and `result_JSON`. Currently, the organization of these files is different for the simulator backend and the experiment backend. This is because they were developed at different times. The simulator backend has been tested a lot more than the experiment backend and in future its file structure will also be used for the experiment backend. We first describe the simulator backend and then the experiment backend.

7.4.1 The simulator backend

This is a Ubuntu virtual machine running a file called `maintainer.py`. The `maintainer.py` file in turn runs other python files called `spooler(s)`. The directory structure of the simulator backend is shown in figure 7.3 :

The `maintainer.py` runs an endless while loop. Every 2 seconds, the `maintainer.py` selects a backend from a list and asks the server for the next job to be executed for that particular backend. If the server replies with a job, the `maintainer` will execute the appropriate `spooler.py` file for that backend. If the server replied with empty response then the loop continues and `maintainer` will ask for job for the next backend defined in the list `backends_list`.

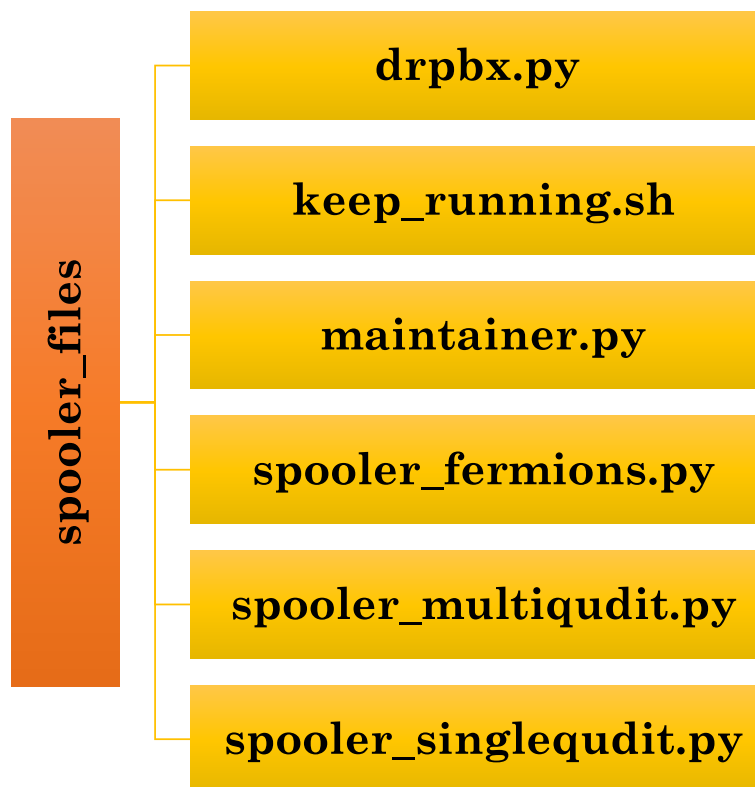


Figure 7.3: Simulator backend files.

Lets consider the case where the server replied with a job name. The maintainer will download the contents of this `job_JSON` and also the `status_JSON` from Dropbox. Both the job dictionary and status dictionary are passed as an argument to the `add_job` function of the appropriate spooler file for that backend. The `add_job` function will perform sanity check on the dictionary using predefined *JSON* schema and execute the circuit. Then it returns the status and result as python dictionaries. These dictionaries are used by the maintainer to update `status_JSON` and `result_JSON` on Dropbox. The Dropbox python API is used for reading from and writing to Dropbox. The corresponding code is in the file `drpbx.py`.

For details on how exactly the circuits are executed, one can have a look at the function `gen_circuit` in each spooler file. Essentially it boils down to creating the correct unitary matrix for each instruction in the instruction list.

There are some important points to note here:

- The `maintainer.py` is a python program. It might very well crash and then our simulator is no longer executing circuits. Although the server is still fine and storing jobs properly but the jobs will be in waiting queue as long as the simulator is dead. As a quick fix to this problem, we do not run the `maintainer.py` file directly. Instead we have a bash script `keep_running.sh` which runs the `maintainer.py` in a loop. If python file crashes the bash script will automatically

restart it.

One might ask and what if the bash script also crashes? Now we enter the realm of keeping a script running on a Linux computer. There are several packages (like `supervisord`, `monit`, `runit` etc.) that offer a solution to this problem. We have a simpler approach. We use a cron job to restart the simulator computer every 6 hours. On restart another cron job launches the bash script `keep_running.sh`. This way the simulator gets revived every 6 hours even if it got stuck.

- The spooler is a virtual machine (VM) living in the cloud. The code for the simulator backends is stored on GitHub repository `coquma-sim-spooler`. In order to deploy the latest GitHub code on the VM, we use a GitHub action that uses the SCP protocol to copy the latest files to the VM on each push to the main branch of the GitHub repository.

7.4.2 The experiment backend

A first draft for implementing the experiment backends is highlighted in the `labscript-qc` GitHub repository. The code was tested to perform MOT loading in a cold atom machine in Heidelberg by submitting circuits from Zurich. Although we still need to polish this part of the code, it already highlights important ideas on how to connect a real cold atom machine to `qlue`. There are two files : `Spooler.py` and `Result.py`. Their functionality is explained below.

The Spooler.py

After the `post_job` view dumps the *JSON* files on the storage database they have to be processed further to execute experiments on the cold atom machine. In order to run experiments on our cold atom setup, we use the *labscript suite* as explained in section 2.6. This means we have to convert the received *JSON* files into python code understandable by *labscript*. This is done by `Spooler.py` file. We will mention three parts of the *labscript suite* : *Runmanager*, *BLACS* and *Lyse*. Documentation about them is available at `labscript suite` repositories⁴.

The `Spooler.py` checks for a `job_JSON` every 3 seconds in the storage location on the database. It then picks up the `job_JSON` that was created first and starts to process it. After all the steps of processing are done the `job_JSON` file is moved to a different location.

During processing, the `spooler` first checks if the *JSON* satisfies a predefined schema. This is also a safety check which allows to see if the user submitted anything in the *JSON* which is inappropriate. After this it updates the status dictionary. Then it continues processing by using the *JSON* to extract an experiment python file for

⁴<https://github.com/labscript-suite>

labscript and set the value of parameters in *Runmanager*. After generating the files and setting parameters, the spooler triggers the execution of the experiment via *runmanager remote API* commands. The status dictionary for that job is appropriately updated.

In *labscript* all the shots generated from a given python code are stored as HDF files. Lets say for a given experiment python file we have 10 shots i.e. 10 HDF files. These shots are passed from *Runmanager* to *BLACS* for actual execution. *BLACS* queues all the shots and executes them sequentially. All data pertaining to a shot (e.g. value of parameters, camera images etc.) is stored in its HDF file. Further data analysis is now run on these HDF files.

The Result.py

After the shots have been executed, we use *Lyse* to run analysis routines on the HDF files. There are two types of analysis routines: single shot and multi shot. This was explained in section 2.6. Given the location of a shot folder *Lyse* can generate a pandas dataframe by reading all the values, be it *Runmanager* globals or analysis routine results. It is this pandas dataframe we are interested in.

Currently by using *QisKit* plugin if we had to scan a parameter across shots, it is done by creating a new experiment dictionary for each value of the parameter. So for e.g. if we want to scan a parameter across 5 values, our *job_JSON* dictionary will have 5 nested experiment dictionaries in it, one each for a particular value of the parameter. And each of the 5 experiment dictionaries will also have a *shots* key which will tell how many times that particular experiment is repeated.

When the *Spooler.py* starts execution of this *JSON* in *labscript*, it will create a job folder with *job_id* name. Inside this job folder there will be 5 sub-folders one for each experiment. Inside each experiment's folder will be the HDF files for the shots of that particular experiment.

As the individual shots get executed they dump their complete HDF path in a separate text file (call it *address_file*) for each shot. This way we know which shots have finished executing. The *Result.py* keeps checking this dump location for these *address_files* and selects the first created *address_file*. It gets the shot path in it and starts with processing of that particular shot. First it will run all single shot analysis routines on this files as defined in the *store_result()* function. The results of the single shot analysis are stored inside the HDF file by creating appropriate groups. After this it calls the *move_to_sds()* function which will move the HDF file from this location to a network storage i.e. SDS. Also it will move the *address_file* of this shot from the original dump location to a new one.

After moving to SDS, the *Result.py* updates a *job_dictionary* which is initially read from a text file. This dictionary keeps track of all running jobs. If the *job_id* of the

shot just moved to SDS is not in this dictionary, it is included along-with its folder location in SDS. This dictionary is also useful to determine on which job a multi-shot analysis can be run. The `result.py` checks the first key in this dictionary and figures out if that job is done or not. If yes then it proceeds with multi-shot analysis for that job by using Lyse to generate a CSV from pandas dataframe for each sub-folder of the experiments in a job. After generating CSVs it generates the result JSON for this job in a specific format given by the schema given in 7.4. Then it updates the status of this job to 'DONE'. Finally the finished job is removed from the dictionary of running jobs.

7.5 Usage examples

In this section few examples are described wherein we have used the *qlue* architecture to run some jobs on a real experimental backend and on simulator backends to reproduce the experimental observations for different systems.

7.5.1 MOT loading on a experiment

This example presents a proof-of-principle demonstration that the architecture of *qlue* can be used to submit jobs to a cold atom experiment backend. In this case the remote client submitted a MOT loading experiment circuit written in *Qiskit*. The circuit was executed on our SoPa experiment. The client submitted circuits from Zurich to a cold atom machine in Heidelberg. The result is shown in figure 7.4. The circuit implementation is available in *Qiskit* (in `qiskit-cold-atom` GitHub repository).

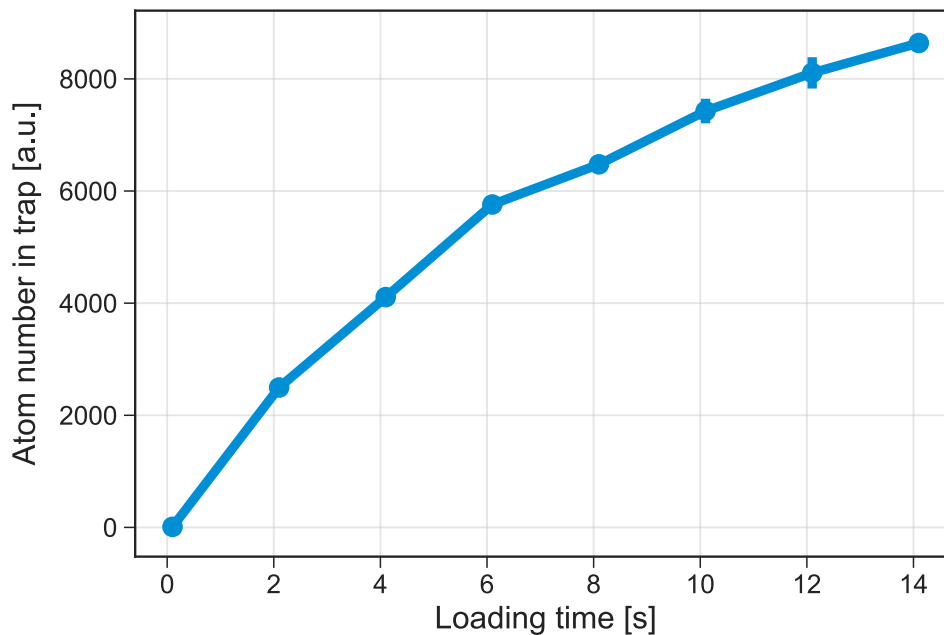


Figure 7.4: MOT loading in SoPa experiment via *qlue*.

7.5.2 Spin-squeezing in a BEC

This example reproduces results of the experimental observation of spin-squeezing mentioned in [147]. This is done by using the single qudit simulator backend. The experiment involves evolution under the Hamiltonian in equation 6.2 which produces a squeezed state quantified by reduction in variance of the measurement observable L_z . This evolution is first expressed as a circuit built out of gates supported by the single qudit backend. There after the circuit can be submitted to the server and results can be analysed later when available. The circuit implementation is available for both *pennylane* (in `pennylane-ls` GitHub repo) and *Qiskit* (in `qiskit-cold-atom` GitHub repo). We compare the results from the cloud simulator with the experimental data [147] as shown in figure 7.5. The comparison is done for two circuits with two different evolution times under the Hamiltonian in equation 6.2. We see that the simulator results agree well with the experimental observation.

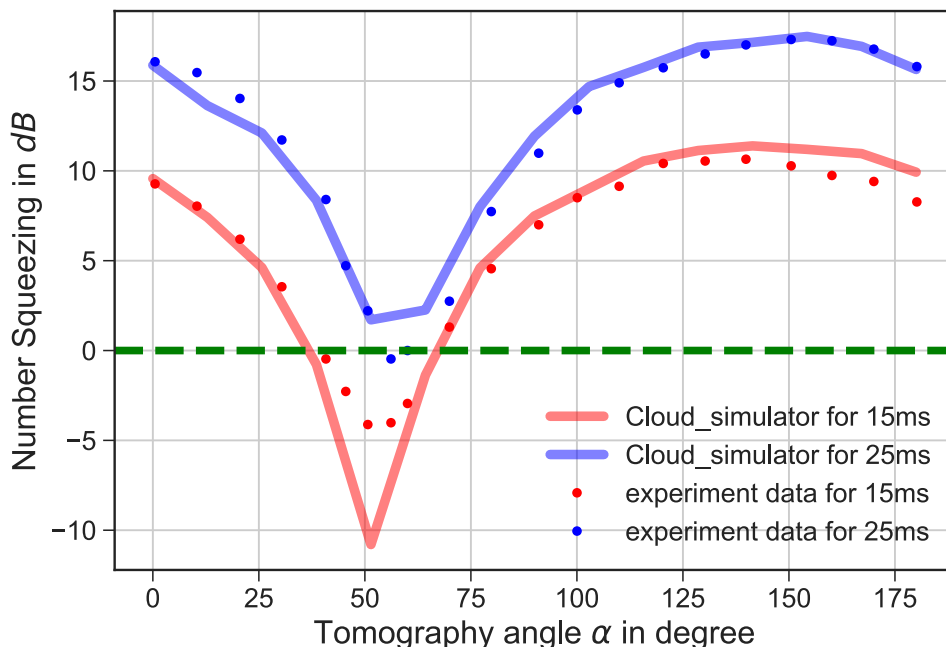


Figure 7.5: Simulating spin-squeezing in a BEC via qlue. The comparison with the experiment is done for two circuits with two different evolution times.

7.5.3 Fermi Hubbard model dynamics

This example reproduces results of the experimental observation of Fermi-Hubbard model dynamics in a double well potential mentioned in [347]. This is done by using the fermion simulator backend. The experiment involves loading the left well with two fermions of opposite spin and observing the atom number in the right well. This evolution is first expressed as a circuit built out of gates supported by the fermion backend. There after the circuit can be submitted to the server and results can be analysed later when available. The circuit implementation is available for both *pennylane* (in `pennylane-ls` GitHub repo) and *Qiskit* (in `qiskit-cold-atom` GitHub

repo). We compare the results from the cloud simulator with the experimental data [347] as shown in figure 7.6. The comparison is done for two circuits with two different interaction strengths under the standard Fermi-Hubbard Hamiltonian. We see that the simulator results agree well with the experimental observation.

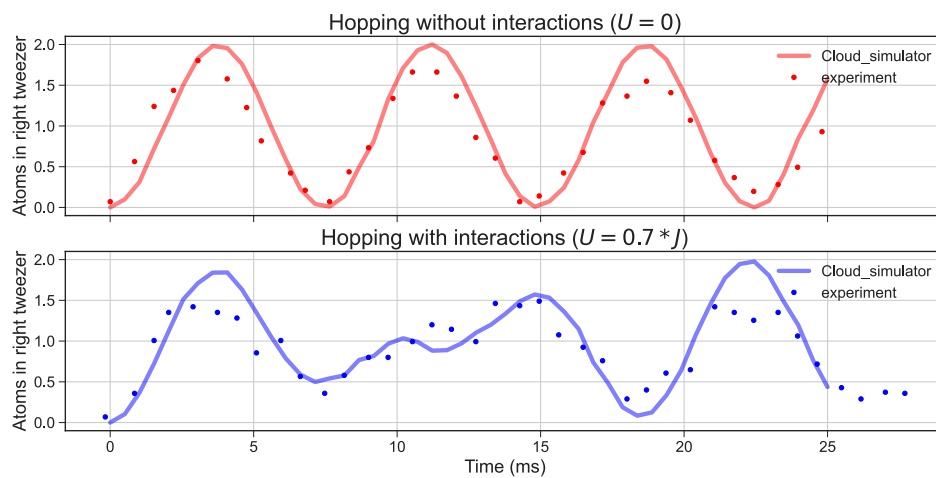


Figure 7.6: Simulating Fermi Hubbard model dynamics via qlue. The comparison with the experiment is done for two circuits with two different interaction strengths under the standard Fermi-Hubbard Hamiltonian.

7.6 Summary

In this chapter, I described our work on developing a general-purpose architecture to access a remote quantum research platform. The platform can be an actual experiment or a simulator running on a computer. Using this architecture we have demonstrated proof-of-principle control of our SoPa experiment and have also used the simulator backends for reproducing experimental observations from our and other groups in Heidelberg.

Conclusion and outlook

In this thesis, I described the work on building up of various components in a toolkit for research in quantum simulation. This included progress on three fronts: experimental platform, theoretical methods, and technological framework. Each of these was inspired by existing work by other research groups while trying to improve upon the limitations. The unifying theme for all the three cases was the versatility and general purpose applicability of the tools being developed.

On the experiment, we have already achieved laser cooling in a dual species MOT of sodium and potassium. We established the experimental capabilities of our newly setup machine by performing single atom counting in a sodium MOT. Moving on the route to obtaining a sodium BEC, we upgraded to a compact electromagnet coil design, optimized the performance of the MOT and have observed loading of sodium atoms in magnetic and optical dipole traps. We also tried to perform evaporative cooling in the magnetic and optical dipole traps but did not observe a gain in phase-space density in the atom cloud. We were unable to try evaporative cooling at higher magnetic field gradients due to limited water cooling capabilities of our compact coil design. On the potassium side, we encountered issues with the vacuum level in the 2D-MOT chamber. Thanks to our modular experimental setup, we were able to isolate the potassium 2D-MOT chamber to continue debugging its vacuum problems while simultaneously progressing the experiment on the sodium part.

To realize the goal of immersing potassium atoms in a sodium BEC, the next steps on the experiment include further improvement of the initial atom number and density in the magnetic trap. This is crucial to gain phase space density by performing evaporative cooling. For this, we are trying to improve the atom flux of our 2D MOT by using additional laser power from an independent laser and also trying to optimize the Zeeman slower performance. Alternatively, we can use the additional laser for gray molasses cooling of sodium to achieve lower temperatures before loading into the magnetic trap. The implementation of an upgraded magnetic trap with higher currents and field gradients is in progress, using new electromagnetic coils with better water cooling. This will also make subsequent cooling in the dipole trap more efficient. For potassium, the 2D-MOT vacuum issues have to be resolved possibly by

replacing the existing ion-getter pump. Then we will implement simultaneous capture of sodium and potassium in the magnetic trap and sympathetically cool potassium by evaporating only sodium. This should allow loading single potassium atoms in optical tweezers and possibly cooling to its motional ground state. For observing the single atoms, the imaging has to be modified by increasing the magnification to resolve different tweezers sites. Finally, the tweezers sorting algorithm, which we have developed, will be implemented to assemble a defect free optical tweezers array.

On the theory side, we have proposed methods for realizing random unitaries with cold atoms in bosonic collective spin systems and fermionic systems in a lattice. For the bosonic system, we tried to base our protocol on the experiments already performed previously in our research group. Our protocol also highlights the connection between chaos (in classical limit) and random unitaries. For the fermionic system, we provided an alternative protocol, which relaxes the experimental requirements of the other existing protocol and can be particularly beneficial for certain atomic species (like lithium-6).

As a next step, we need to look further into optimizing the time required to realize a random unitary. For the bosonic case, one easy way is to increase the spin length because the slowest time scale (from the non-linear interaction energy) increases with increasing atom number. For the fermionic case, the time scales are already shorter than the bosonic case. However, we need to understand why our method of generating random unitaries for fermions does not work for the $S_z = 0$ sector. An interesting follow-up could be extending the fermion protocol to two dimensional lattices for enabling its application in fermionic quantum gas microscope experiments.

For the web interface, we have already demonstrated a proof of principle example where a remote user submitted circuits to our experiment. They were executed on our machine and the results were made available to the user in a secure way. We have also demonstrated the workflow in other examples where we could reproduce some previous experimental results with our theory simulator backends. At present, the infrastructure is ready to be adopted by any experiment or theory group.

An additional feature we could implement in the infrastructure is to show a simple example of scalability by having more than one server which accepts the circuits and more than one virtual machine (for theory backends for example) which process them. The architecture supports this by reorganizing folder names in the database and some server URLs. Although the infrastructure is fully open source and free to be implemented by anyone on their own, some research groups might not want to do this but use our setup as a service. In that case, we have to draft usage and sharing policies for collaboration and sharing secrets like API keys.

Appendices

Technical details

This appendix presents some technical details omitted in [Chapter 2](#). In the following sections, a description of the electronics and the analysis code is given.

A.1 Electronics

In our experiment, we have a lot of electronic components which include both commercial and home-built devices. To mention a few items, the list includes control electronics for lasers and vacuum pumps, RF drivers and amplifiers for AOMs and EOMs, power supplies for various circuits, and signal generation/acquisition modules for controlling/interfacing various hardware in the experiment etc. For efficient space utilization, our electronics are housed in two 19-inch racks which allows us to save space by stacking them.

Given below is a list of some electronics solutions we have self-built for various needs in the lab :

- **Temperature stabilization** : We use a common strategy for temperature stabilization in our experiment. Be it temperature of the room, spectroscopy vapor cell, viewports of the vacuum system or the oven for the atomic species. The circuit consists of a thermocouple sensor which produces a calibrated voltage at a given temperature. This voltage is then amplified (using Adafruit amplifier board *AD8495*) and fed into an analog input pin of an Arduino Yun. The Arduino runs a PID stabilization code and gives a correction voltage. The correction voltage goes to a controllable heating source which regulates the current through that particular heater. For temperature stabilization of the vapor cell, viewports and ovens, we use a phase dimmer (NS-80 from *FG Elektronik*) as the controllable heating source. For regulating the room temperature we have an air conditioner which is always running and a controllable heater which receives the correction from the Arduino and adjusts the heating to stabilize the room temperature.
- **Photo diode** : Photo-diodes output a photo-current proportional to the light

intensity falling on the sensor. However, for most applications this current needs to be converted into a voltage. While a simple resistor could be used to achieve this, it will suffer from bandwidth issues due to parasitic stray capacitance of the photo-diode. This can be critical for AC applications where the light signal is modulating. To resolve this, a trans-impedance amplifier can be used which allows one to select both the gain and the bandwidth for the operation. Our photo-diode design based on such a trans-impedance circuits is shown in figure A.1.

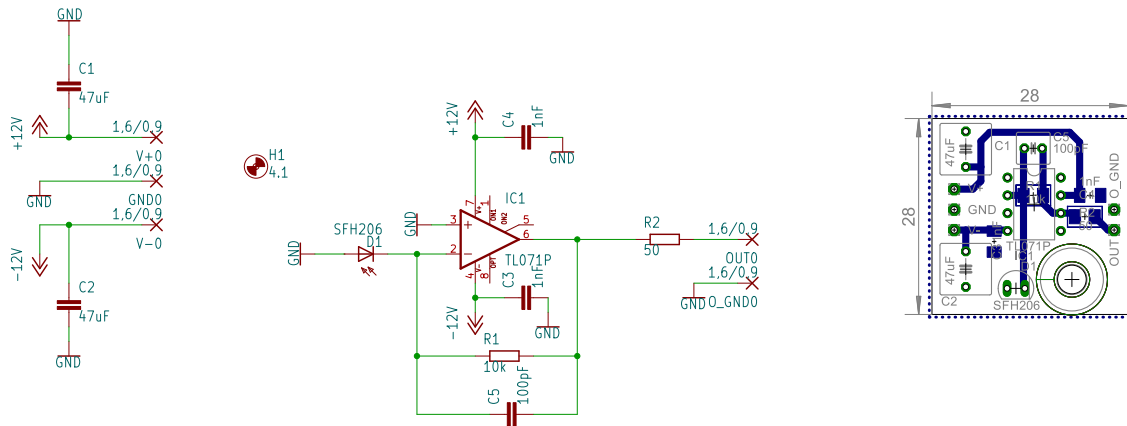


Figure A.1: Photo-diode circuit schematic and PCB layout.

- Laser beam shutters** : In our experiments we need to generate pulses of lasers beams during the sequence. Usually AOMs are capable of ultrafast switching with high repeatability. But they do not offer complete extinction of the beams. For this we use a simple mechanical shutter which consists of a razor blade glued onto a electromagnetic relay for moving the blade in and out of the beam. The circuit for controlling the shutter relay is shown in figure A.2.

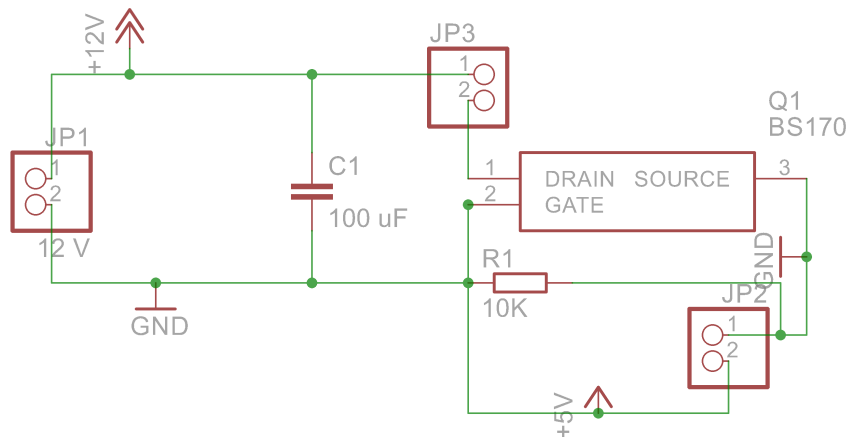


Figure A.2: Shutter controller circuit schematic.

- **Water leak detector** : For our new MOT coils which are water-cooled (see section 4.2), we made a simple circuit for detecting water leaks. It uses a soldering board with strips and we connect all alternate strips together. So if a water drop falls between any two strips, the circuit gets completed and a buzzer in the circuit turns on.
- **MOT coil switching circuit** : The circuit for turning off and on the current in the MOT coils is shown in figure A.3. A driver circuit receives the TTL trigger from NI PXIe 6535 card and produces the necessary gate voltage to control the IGBT.

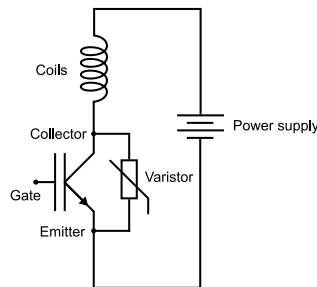


Figure A.3: IGBT switching circuit schematic.

A.2 Analysis code structure

As explained in section 2.6, our analysis code is divided into two parts : the python file with all analysis functions and the jupyter notebook where these functions are called. A simplified layout of the python file is given in the table A.1.

```

1  import numpy as np
2  from scipy.ndimage import gaussian_filter
3  import random
4  from copy import deepcopy
5  from scipy.optimize import curve_fit
6
7  class analysis_functions():
8      def gaussian(self, x, A, mu, sig, c):
9          return A*np.exp(-np.power(x - mu, 2.) / (2 * np.power(sig, 2.))) + c
10
11     def get_bare_subtracted_images(self, atom_images, background_images):
12         subtracted_images_bare=atom_images-background_images
13         return subtracted_images_bare

```

Table A.1: A layout of the analysis file. Only two functions are shown for demonstration.

The functions from this file are used in the jupyter notebook. To demonstrate this, first few cells of the jupyter notebook are given below.

A. Technical details

```
from lyse import *
import os
import sys
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.patches as patches
import math
from scipy.optimize import curve_fit
from scipy import constants
from scipy import ndimage
```

Specify the scan folder path and get a list of paths for all shots in there.

```
scan_folder=R'C:\Users\...\Experiment\2021\10\19\0004'
file_list=list(fn for fn in next(os.walk(scan_folder))[2])
shot_paths = [os.path.join(scan_folder,fn) for fn in file_list]
```

Get the LYSE info dataframe for all these shots. Then you can filter out the shots later based on what particular case you want to analyse.

```
df=dataframe_utilities.get_dataframe_from_shots(shot_paths)
```

Lets check what does the summary tell us about this scan.

```
df['Summary'][0]
```

Lets find out which globals were scanned.

```
# Get the List of globals
r = Run(shot_paths[0], no_write = False)
globals_list=list(r.get_globals())
# Make dataframe subset with the list of globals
df1=df[globals_list]
# Using the dataframe subset, the following list produces globals that had a non_unique value during the entire scan.
changing_list=list(set(list(df1.columns)) - set(list(df1.columns[df1.nunique() <= 1])))
# Now make a list of scanned globals
scanned_globals_list=[x[0] for x in changing_list]
# Remove the global 'Run_number' from the list
scanned_globals_list.remove('Run_number')
#scanned_globals_list.remove('some_other_global')
# Optionally if you have a scan of boolean variables you might want to remove them too from this list.
```

```
print(scanned_globals_list)
```

```
scanned_globals_dict={x:sorted(list(df[x].unique())) for x in scanned_globals_list}
```

```
print(scanned_globals_dict)
```

Import the functions for our analysis

```
#analysis_file_path=R"path_to_analysis_file"
#sys.path.append(os.path.abspath(analysis_file_path))
#We have all analysis functions in a file called 'No_lyse_GUI_analysis_functions' which lives in analysis_file_path
from No_lyse_GUI_analysis_functions import *
%load_ext autoreload
%autoreload 2
```

Data loading, shaping and masks declaration

```
Ny == Camera_Frame_Rows
```

```
Nx == Camera_Frame_Columns
```

```
def load_pics(file_path=None, fluo_images_number=0, orientation = 'NaZyla'):
    Ny, Nx = Run(file_path[0]).get_image(orientation, 'NaReference').shape

    atom_images = np.zeros((fluo_images_number, Ny, Nx)).astype('float32')
    background_images = np.zeros((fluo_images_number, Ny, Nx)).astype('float32')

    for i in range(fluo_images_number):
        r = Run(file_path[i])
        background_images[i,:,:]=np.asarray(r.get_image(orientation, 'fluorescence', 'NaReference'))
        atom_images[i,:,:]=np.asarray(r.get_image(orientation, 'fluorescence', 'NaAtoms'))

    return atom_images, background_images, Ny, Nx, fluo_images_number
```

```
def load_pics_mako(file_path=None, fluo_images_number=0, orientation = 'mako'):
    Ny, Nx = Run(file_path[0]).get_image(orientation, 'NaReference', 'frame').shape

    atom_images = np.zeros((fluo_images_number, Ny, Nx)).astype('float32')
    background_images = np.zeros((fluo_images_number, Ny, Nx)).astype('float32')

    for i in range(fluo_images_number):
        r = Run(file_path[i])
        background_images[i,:,:]=np.asarray(r.get_image(orientation, 'NaReference', 'frame'))
        atom_images[i,:,:]=np.asarray(r.get_image(orientation, 'NaAtoms', 'frame'))

    return atom_images, background_images, Ny, Nx, fluo_images_number
```

Flatten images. So it would be cool to get the same shape as in `transform_intensities` here.
****It actually does that.****

```
def transform_intensities(atom_images, background_images, Ny, Nx, fluo_images_number=0):
    atom_images=atom_images.reshape(fluo_images_number, Ny*Nx)
    background_images=background_images.reshape(fluo_images_number, Ny*Nx)
    return atom_images, background_images, Ny, Nx
```

Define ROI and ROS

```
left_ROS=60 #min value 0
right_ROS=left_ROS+20 #max value 2560 for Zyla
up_ROS=70 #in value 0
down_ROS=up_ROS+20 #max value 2160 for Zyla

x0=350#63
y0=250#70
height=250
width=250
degree=0#15
rotation_dt =0
theta=np.pi*(degree/180)

orientation = 'mako'
```

```
analyse=analysis_functions()
```

```
def get_ratio_corrected_results():
    once=True
    MOT_resultant_histogram_ratio_corrected = pd.DataFrame(columns=[scanned_globals_list[0], 'Counts'])
    for scanned_global in scanned_globals_list:
        for i in range(len(scanned_globals_dict[scanned_global])):
            scanned_global_value=scanned_globals_dict[scanned_global][i]
            #print(scanned_global_value)
            filter_condition=(df[scanned_global]==scanned_global_value)&&(df['some_other_condition']==False)
            file_path=list(df.loc[filter_condition]['filepath'])
            fluo_images_number=len(file_path)#df['Number_of_atom_images'][0]#
            if fluo_images_number ==0:
                continue
            atom_images, background_images, Ny, Nx = transform_intensities(
                *load_pics_mako(file_path, fluo_images_number, orientation))
            if once:
                analyse.set_regions(Ny, Nx, left_ROS, right_ROS, up_ROS, down_ROS, x0, y0, height, width, theta)
                analyse.set_flattened_masks()
                once=False
            subtracted_images = analyse.get_bare_subtracted_images(atom_images, background_images)

    return
```


Tweezers sorting algorithm

In this appendix, the details of our algorithm for generating a 1D tweezer array and for sorting the tweezer array into a defect-free configuration are given.

B.1 The scheme

To generate a 1D array of tweezers, an AOD is driven with multiple frequencies produced using an AWG. These frequencies are usually few tens to hundred MHz and the AWG produces samples at the rate of several hundred MHz. After the array is generated, the loading of atoms in the tweezers takes place in a stochastic fashion i.e. only some tweezers are loaded. To get a defect free assembly, the array needs to be rearranged by updating the frequencies of tweezers loaded with atoms. For this reason, we need to operate the AWG in FIFO mode where it continuously streams new data from a computer which is responsible for calculating updated frequencies.

In the FIFO mode, the AWG continuously outputs data from its own on-board memory. As a result of this the free space in the on-board memory increases as the AWG continues to output data. Once the free space in the on-board memory exceeds a threshold, the AWG fetches fresh data from the computer. Since the AWG outputs data at several hundred MHz rate, the computer would also need to calculate the next data points very fast. Performing such calculations on-the-fly might not always be possible with a given computer hardware.

The method described here allows to use very high sample rates from the AWG even when the computer cannot calculate data very fast. The overall scheme involves using two data arrays in computer memory which are used alternately to stream their contents to the AWG. The selection of which array will stream to the AWG is done using a Boolean variable. When one array is being used for streaming to the AWG, the other array is used for calculating and storing the data of the next move. After this calculation, the Boolean variable is flipped, and the role of the two arrays interchange (i.e. the first one is now used for calculation and the second one for streaming). Since the array used for streaming to the AWG is not used for calculating, this scheme

guarantees that there will never be a shortage of data when the AWG is outputting at very high sample rates. A diagrammatic representation of this scheme is given in figure B.1.

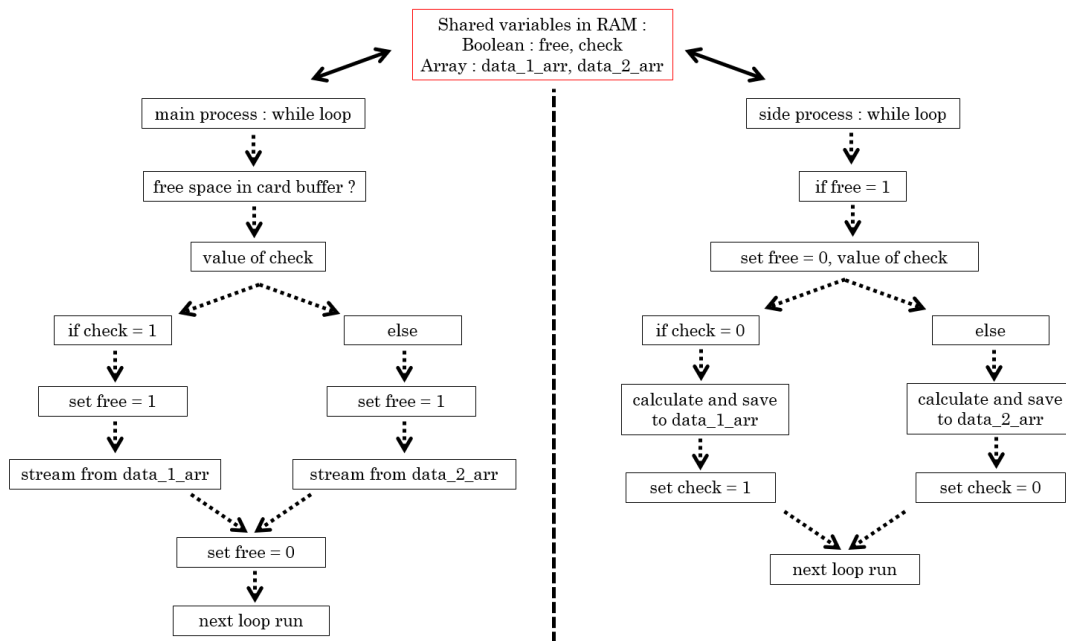


Figure B.1: Sorting algorithm : The main process and side process run independently and share four variables in the computer memory.

The scheme described above is realized by using multiprocessing in python. We run two independent python processes which share four variables in the computer memory. The different parts are described below :

- **Shared variables** : These variables are synchronized (using Rlock) according to python language multiprocessing rules¹. They are of two types : Boolean and Array. There are two Boolean variables : *free*, *check* and two Array variables : *data_1_arr*, *data_2_arr*. The variable *free* is *True* (or 1) if there is some free space in AWG on-board memory. Only then the side process calculates the next move. The variable *check* is *True* (or 1) if *data_1_arr* is being used by the main process for streaming to the AWG. In that case the variable *data_2_arr* is used by the side process for calculating the next move. For the case when *check* is *False* (or 0), the role of *data_1_arr* and *data_2_arr* is reversed.
- **Main process** : This is the part of the code (see B.4) below the line *if __name__ == '__main__':*. It is responsible for checking whether there is some free space (above a threshold) available in the AWG on-board memory. If yes, then it streams data from the computer memory to the AWG. Depending on the value of the Boolean variable *check*, it selects one of the two arrays for streaming.

¹See python documentation at <https://docs.python.org/3/library/multiprocessing.html>

- **Side process** : It runs the function `jump_between()` which calculates and stores the data for the next streaming by the main process. It uses two arrays alternately for calculation. Using the Boolean variable `check`, the scheme is synchronized such that only the array not being used by the main process for streaming, is used for calculating the next move. This way there is no problem of data update speed even when the next move is calculated slowly. Once the new data for the next move is calculated and stored, the value of `check` is flipped. So in the next streaming event the main process will stream the new data to the AWG.

B.2 Calculation methods

To calculate the signal for all tweezers, we start by first selecting a frequency resolution. This tells us how long of a signal we need to generate in time to be able to resolve two frequencies separated by the desired frequency resolution. After that we define the frequencies which we need to generate. These are an array of equidistant numbers. For each frequency we also define an amplitude. Then we have to calculate the sinusoidal signal for each frequency upto the calculated time duration (reciprocal of the frequency resolution) and then sum up each frequency component into one array. This has to be done in a efficient way using vectorization.

Lets say we want to generate a tweezers array with 5 tweezers. Lets consider the same parameters we calculated in section 4.7.3 where for a separation of $5\ \mu\text{m}$ between consecutive tweezers, their frequencies need to be separated by 0.75 MHz. However, when we update the frequencies, we want to move the tweezers in a step size which is fine enough such that atoms do not get lost while moving. For this, the tweezer can only move by a fraction of its waist at one update step. Lets say we fix this step size to be at one-fourth of the tweezer waist. This also gives us the frequency resolution we must have. Since we calculated the tweezer waist to be $2.8\ \mu\text{m}$, one-fourth of this is $0.7\ \mu\text{m}$. We also know that $5\ \mu\text{m}$ is equal to 0.75 MHz in frequency. This means $0.7\ \mu\text{m}$ is equal to 0.105 MHz in frequency. So our required frequency resolution is 0.105 MHz.

For being able to resolve two signals which are 0.105 MHz apart, we need to produce the signal for a long enough time for which the lower limit is the reciprocal of the frequency resolution. In our case this boils down to about $10\ \mu\text{s}$. So we define a time array from 0 to $10\ \mu\text{s}$. The number of samples required in this array is equal to the product of sample rate of the AWG and $10\ \mu\text{s}$. With this array we calculate the sine wave for each frequency (with corresponding amplitude). Finally, for each time point index, we sum the value of each of the five arrays to get a 1D array which is ready to be streamed to the AWG. This process can be continued till the target frequency configuration is reached.

B.3 Performance

Since atoms trapped in the tweezers have a finite lifetime, the sorting process should be done as quickly as possible. In this section the timings required for sorting some configurations into a defect free target are given. Starting with a given number of tweezers, a stochastic loading of 50% is assumed and 5 configurations corresponding to each such setting are considered. Different configurations require different sorting timings and we report the mean value and error bar for each timing. The timings were measured using builtin timing functions of python.

B.3.1 Setting 1 :

We consider 10 tweezers and 5 filled with atoms. The frequencies for tweezers are 48 MHz to 58 MHz in steps of 1 MHz. Then we consider 5 configurations for which we measure timings. In the table below a 1 represents an occupied tweezer while 0 represents an empty tweezer.

Initial configuration	Target configuration	Sorting time (in ms)
0100111100	0001111100	25.5(5)
1001111000	0001111100	34(2)
1010101001	0001111100	37(2)
0101100110	0001111100	26(2)
1000100111	0001111100	37(1.5)

Table B.1: Sorting performance for 1D array of 10 tweezers.

B.3.2 Setting 2 :

We consider 20 tweezers and 10 filled with atoms. The frequencies are 42 MHz to 62 MHz in steps of 1 MHz. Then we consider 5 configurations for which we measure timings. In the table below a 1 represents an occupied tweezer while 0 represents an empty tweezer.

Initial configuration	Target configuration	Sorting time (in ms)
00101111010110100010	00000111111111100000	89(6)
00110110000110101011	00000111111111100000	108(6)
11001010001110001101	00000111111111100000	104(8)
10100101000011110011	00000111111111100000	107(5)
11101100100111010000	00000111111111100000	113(9)

Table B.2: Sorting performance for 1D array of 20 tweezers.

B.4 Code

The following cell reproduces the sorting code. The code is also publicly available at <https://github.com/synqs/sorting-tweezers>.

```

1  # Importing modules
2  import sys
3  import os
4  sys.path.append(os.path.abspath(R"C:\Users\SoPa\Documents\GitHub\spectrum_awg\RPB"))
5  from pyspcm import *
6  from spcm_tools import *
7  import ctypes
8  import numpy as np
9  import numexpr as ne
10 from multiprocessing import Process, Value, Array
11 from timeit import default_timer as timer
12
13 def get_freq_and_amp(sign, frequencies, amplitudes):
14     """
15     This function calculates the frequencies for the immediately next step.
16     """
17     freq_update_step = 50e3 # in Hz. frequency step to take while moving the
18                             tweezers.
19     freq = frequencies+(sign*freq_update_step)
20     amp = amplitudes
21     return freq, amp
22
23 def calculate_target_frequency(start_frequencies, tweezer_occu):
24     """
25     This function calculates the target frequency arrangement. Target is
26     same for all until configs with same number of tweezers and loaded
27     atoms. The target arrangement is assembled around the center of full array.
28     """
29     inds,=np.asarray(tweezer_occu==1).nonzero() #
30     center=np.take(inds, inds.size//2)
31     z,=np.asarray(inds==center).nonzero() #
32     new_inds=np.arange(center-z[0],center-z[0]+inds.size)
33     new_inds=new_inds+(tweezer_occu.size//2)-center
34     return start_frequencies[new_inds]
35
36 def calculate_signal(t, frequencies, amplitudes):
37     """
38     This function calculates the total data using numpy
39     vectorization and broadcasting.
40     """
41     return np.sum(amplitudes[:,None]*np.sin(2*np.pi*frequencies[:,None]*t[None,:]),
42                  axis=0).astype('int16')
43
44 def calculate_signal_fast(t, frequencies, amplitudes):
45     """
46     This function calculates the total data using numpy

```

B. Tweezers sorting algorithm

```
45     and numexpr. Performance is better than numpy alone.
46     """
47     return np.sum(amplitudes[:,None]*ne.evaluate('sin(arg)', {'arg':2*np.pi*
                                                    frequencies[:,None]*t[None,:]}), axis=
48                                     0).astype('int16')
49
50 def jump_between(check, free, data_1_arr, data_2_arr, t):
51     """
52     This function runs in an independent side process.
53     It manipulates the shared arrays and booleans to stream properly to AWG.
54     """
55     start_frequencies = np.arange(42., 62.0, 1.0)*1e6 # frequencies for all tweezers
56                                                         produced intially
57     start_amplitudes = np.linspace(10000//start_frequencies.size, 10000//
58                                                         start_frequencies.size,
59                                                         start_frequencies.size)
60     target_frequencies = np.ones(start_frequencies.size) # frequencies of the defect
61                                                         free array for a particular load
62                                                         configuration
63     frequencies = np.ones(start_frequencies.size) # actual frequencies that are to
64                                                         be streamed to the AWG
65     amplitudes = np.ones(start_frequencies.size)
66     sign = np.ones(start_frequencies.size)
67     file_read=False # local boolean variable for tracking the occupancy text file
68     no_atoms=False # local boolean variable for representing an empty array
69     time_once=True # local boolean variable for measuring sorting performance
70     time_diff=0.0 # local float variable for measuring sorting performance
71     textfile_path=R'C:\Users\SoPa\Documents\tweezer_occupancy.txt'
72     while True:
73         if file_read==False:
74             try:
75                 with open(textfile_path, "r") as text_file:
76                     # time_diff = timer()
77                     tweezer_occu=text_file.read()
78                 file_read=True
79                 tweezer_occu=np.asarray(list(tweezer_occu), dtype=int) # read the
80                                                         tweezer occupancy text
81                                                         file as a numpy array
82                 assert tweezer_occu.size == start_frequencies.size, 'Tweezer
83                                                         occupancy file invalid'
84                 if np.any(tweezer_occu):
85                     frequencies,amplitudes = start_frequencies[tweezer_occu==1],
86                                                         start_amplitudes[
87                                                         tweezer_occu==1]
88                     target_frequencies = calculate_target_frequency(
89                                                         start_frequencies,
90                                                         tweezer_occu)
91             else:
92                 no_atoms=True
93     except Exception as e:
94         print(e)
```

```

81         pass
82     continue
83     if no_atoms:
84         continue
85     if free.value==1:
86         free.value=0
87         freq_diff=target_frequencies-frequencies
88         sign=np.sign(freq_diff) # array to store whether each frequency os above
89                                 # or below the target
90     if np.any(sign): # this executes until target frequencies are reached
91         frequencies,amplitudes = get_freq_and_amp(sign,frequencies,
92                                                     amplitudes)
93     else: # this executes when target frequencies are reached
94         if time_once: # this executes once, when target frequencies are
95                       # reached
96             # time_diff=timer()-time_diff
97             # print(time_diff)
98             time_once=False
99             pass
100        if check.value == 0:
101            try:
102                temp_arr_1=calculate_signal_fast(t, frequencies,amplitudes)
103                ctypes.memmove(ctypes.byref(data_1_arr.get_obj()), temp_arr_1.
104                                ctypes.data,
105                                temp_arr_1.nbytes)
106            except Exception as e:
107                print(e)
108                check.value = 1
109        else:
110            try:
111                temp_arr_2=calculate_signal_fast(t, frequencies,amplitudes)
112                ctypes.memmove(ctypes.byref(data_2_arr.get_obj()), temp_arr_2.
113                                ctypes.data,
114                                temp_arr_2.nbytes)
115            except Exception as e:
116                print(e)
117                check.value = 0
118
119 # Buffer data transfer function
120 def vCalcNewData (pnBuffer, lNumCh, llSamplePos, llNumSamples,final_data):
121     lStartPosInBuffer_bytes = (llSamplePos % lPreCalcLen) * 2 * lNumCh #why is the
122                                 # factor 2 needed here? int16 2 byte
123                                 # long
124
125     lToCopy_bytes = llNumSamples * 2 * lNumCh
126     lPreCalcLen_bytes = lPreCalcLen * 2 * lNumCh
127     lAlreadyCopied_bytes = 0
128     while lAlreadyCopied_bytes < lToCopy_bytes:
129         # copy at most the pre-calculated data
130         lCopy_bytes = lToCopy_bytes - lAlreadyCopied_bytes
131         if lCopy_bytes > lPreCalcLen_bytes - lStartPosInBuffer_bytes:
132             lCopy_bytes = lPreCalcLen_bytes - lStartPosInBuffer_bytes

```

B. Tweezers sorting algorithm

```
123
124     # copy data from pre-calculated buffer to DMA buffer. The get_obj() function
                                     is for shared Array variable.
125     ctypes.memmove (cast (pnBuffer, c_void_p).value + lAlreadyCopied_bytes, cast
                                     (final_data.get_obj(), c_void_p).
                                     value + lStartPosInBuffer_bytes,
                                     lCopy_bytes)
126     lAlreadyCopied_bytes += lCopy_bytes
127     lStartPosInBuffer_bytes = 0
128
129 if __name__ == '__main__':
130     # Initiate the card
131     spcm_vClose (-1) #this line closes the card first for the case that an error
                                     occurred which stopped the compiling
                                     before the card is closed.
132
133     # open card
134     # uncomment the second line and replace the IP address to use remote cards like
                                     in a generatorNETBOX
135     hCard = spcm_hOpen (create_string_buffer (b'/dev/spcm0'))
136     #hCard = spcm_hOpen (create_string_buffer (b'TCPIP::192.168.1.10::inst0::INSTR')
                                     )
137
138     if hCard == None:
139         sys.stdout.write("no card found...\n")
140         exit ()
141     # read type, function and SN and check for D/A card
142     lCardType = int32 (0)
143     spcm_dwGetParam_i32 (hCard, SPC_PCITYP, byref (lCardType))
144     lSerialNumber = int32 (0)
145     spcm_dwGetParam_i32 (hCard, SPC_PCISERIALNO, byref (lSerialNumber))
146     lFuncType = int32 (0)
147     spcm_dwGetParam_i32 (hCard, SPC_FNCTYPE, byref (lFuncType))
148     sCardName = szTypeToName (lCardType.value)
149     if lFuncType.value == SPCM_TYPE_A0:
150         sys.stdout.write("Found: {0} sn {1:05d}\n".format(sCardName,lSerialNumber.
                                     value))
151     else:
152         sys.stdout.write("This is an example for D/A cards.\nCard: {0} sn {1:05d}
                                     not supported by example\n".format
                                     (sCardName,lSerialNumber.value))
153
154     spcm_vClose (hCard);
155     exit ()
156
157     # Setup the card
158     lPreCalcLen = int(0) # in samples
159     Rate = 500*(10**6) #in Hz, 1250MHz is max for this card
160     # set samplerate to Rate, no clock output
161     spcm_dwSetParam_i64 (hCard, SPC_SAMPLERATE, Rate)
162     spcm_dwSetParam_i32 (hCard, SPC_CLOCKOUT, 0)
163     # driver might have adjusted the sampling rate to the best-matching value, so we
                                     work with that value
164
165     SRate = int64 (0)
```

```

162     spcm_dwGetParam_i64 (hCard, SPC_SAMPLERATE, byref (SRate))
163
164     # setup the mode
165     qwChEnable = uint64 (CHANNEL0)
166     spcm_dwSetParam_i32 (hCard, SPC_CARDMODE,      SPC_REP_FIFO_SINGLE)#
                                     SPC_REP_FIFO_SINGLE)
167     spcm_dwSetParam_i64 (hCard, SPC_CHENABLE,     qwChEnable)
168     spcm_dwSetParam_i64 (hCard, SPC_SEGMENTSIZE, 4096) # used to limit amount of
                                     replayed data if SPC_LOOPS != 0
169     spcm_dwSetParam_i64 (hCard, SPC_LOOPS,       0) # continuous replay
170     lSetChannels = int32 (0)
171     spcm_dwGetParam_i32 (hCard, SPC_CHCOUNT, byref (lSetChannels))
172     lBytesPerSample = int32 (0)
173     spcm_dwGetParam_i32 (hCard, SPC_MIINST_BYTESPERSAMPLE, byref (lBytesPerSample))
174
175     # setup the trigger mode (SW trigger, no output)
176     spcm_dwSetParam_i32 (hCard, SPC_TRIG_ORMASK,   SPC_TMASK_SOFTWARE)
177     #spcm_dwSetParam_i32 (hCard, SPC_TRIG_ORMASK,   SPC_TMASK_NONE)
178     #spcm_dwSetParam_i32 (hCard, SPC_TRIG_ORMASK, SPC_TMASK_EXT0)
179     spcm_dwSetParam_i32 (hCard, SPC_TRIG_ANDMASK, 0)
180     #spcm_dwSetParam_i32 (hCard, SPC_TRIG_EXT0_LEVEL0, 1000)
181     #spcm_dwSetParam_i32 (hCard, SPC_TRIG_EXT0_MODE, SPC_TM_HIGH)
182     #spcm_dwSetParam_i32 (hCard, SPC_CH0_STOPLEVEL, SPCM_STOPLVL_HIGH)
183     #####
184     spcm_dwSetParam_i32 (hCard, SPC_TRIG_CH_ORMASK0, 0)
185     spcm_dwSetParam_i32 (hCard, SPC_TRIG_CH_ORMASK1, 0)
186     spcm_dwSetParam_i32 (hCard, SPC_TRIG_CH_ANDMASK0, 0)
187     spcm_dwSetParam_i32 (hCard, SPC_TRIG_CH_ANDMASK1, 0)
188     spcm_dwSetParam_i32 (hCard, SPC_TRIGGEROUT,    0)
189
190     # setup all channels
191     for i in range (0, lSetChannels.value):
192         spcm_dwSetParam_i32 (hCard, SPC_AMP0 + i * (SPC_AMP1 - SPC_AMP0), int32 (
                                     1000))
193         spcm_dwSetParam_i32 (hCard, SPC_ENABLEOUT0 + i * (SPC_ENABLEOUT1 -
                                     SPC_ENABLEOUT0), int32(1))
194
195     # Data calculation
196     ne.set_num_threads(8) #number of threads for numexpr
197     frequencies = np.arange(42., 62.0, 1.0)*1e6 # frequencies for all tweezers
                                     produced intially
198     amplitudes = np.linspace(10000//frequencies.size, 10000//frequencies.size,
                                     frequencies.size)
199     freq_resolution=50e3 # frequency resolution desired
200     end_time = 1/freq_resolution # time required to achieve desired frequency
                                     resolution
201     number_of_samples = int(SRate.value*end_time)
202     t=np.linspace(0.0, end_time, num=number_of_samples, endpoint=False) # time
                                     points array
203     data_1=calculate_signal_fast(t, frequencies,amplitudes) # data points array
204     data_2=calculate_signal_fast(t, frequencies,amplitudes) # data points array

```

B. Tweezers sorting algorithm

```
205     lPreCalcLen = len(data_1)
206     print('data size in MB', lPreCalcLen*2/1024/1024)
207
208     # setup hardware buffer (card memory)
209     llHWBufSize= uint64 (32*1024*1024) # Do not make too big to reduce latency
210     spcm_dwSetParam_i64 (hCard, SPC_DATA_OUTBUFSIZE, llHWBufSize);
211     spcm_dwSetParam_i32 (hCard, SPC_M2CMD, M2CMD_CARD_WRITESTATUS);
212
213     # setup software buffer or PC memory
214     lNotifySize_bytes = int32(512*1024) # 1 MByte
215     qwBufferSize = uint64 (30*1024*1024) # For simplicity qwBufferSize should be a
                                         multiple of lNotifySize_bytes
216
217     # we try to use continuous memory if available and big enough
218     pvBuffer = c_void_p ()
219     qwContBufLen = uint64 (0)
220     spcm_dwGetContBuf_i64 (hCard, SPCM_BUF_DATA, byref(pvBuffer), byref(qwContBufLen
                                         ))
221     sys.stdout.write ("ContBuf length: {0:d}\n".format(qwContBufLen.value))
222     if qwContBufLen.value >= qwBufferSize.value:
223         sys.stdout.write("Using continuous buffer\n")
224     else:
225         pvBuffer = pvAllocMemPageAligned (qwBufferSize.value)
226         sys.stdout.write("Using buffer allocated by user program\n")
227
228     # we calculate data for all enabled channels, starting at sample position 0, and
                                         fill the complete DMA buffer
229     check = Value('b', True) # synchronized shared Boolean variable
230     free = Value('b', False) # synchronized shared Boolean variable
231     data_1_arr = Array('i', range(lPreCalcLen)) # synchronized shared Array variable
232     data_2_arr = Array('i', range(lPreCalcLen)) # synchronized shared Array variable
233     ctypes.memmove(ctypes.byref(data_1_arr.get_obj()), data_1.ctypes.data, data_1.
                                         nbytes)
234     ctypes.memmove(ctypes.byref(data_2_arr.get_obj()), data_2.ctypes.data, data_2.
                                         nbytes)
235
236     qwSamplePos = 0
237     lNumAvailSamples = (qwBufferSize.value // lSetChannels.value) // lBytesPerSample
                                         .value
238     vCalcNewData (pvBuffer, lSetChannels.value, qwSamplePos, lNumAvailSamples,
                                         data_1_arr)
239     qwSamplePos += lNumAvailSamples
240
241     # we define the buffer for transfer and start the DMA transfer
242     sys.stdout.write("Starting the DMA transfer and waiting until data is in board
                                         memory\n")
243     spcm_dwDefTransfer_i64 (hCard, SPCM_BUF_DATA, SPCM_DIR_PCTOCARD,
                                         lNotifySize_bytes, pvBuffer, uint64 (0
                                         ), qwBufferSize)
244     spcm_dwSetParam_i32 (hCard, SPC_DATA_AVAIL_CARD_LEN, qwBufferSize)
245     spcm_dwSetParam_i32 (hCard, SPC_M2CMD, M2CMD_DATA_STARTDMA)
```



```

246  # we'll start and wait until the card has finished or until a timeout occurs
247  lStatus = int32(0)
248  lAvailUser_bytes = int32(0)
249  LPCPos = int32(0)
250  lFillsize = int32(0)
251  bStarted = False
252  # start the side proces to run the function jump_between()
253  p = Process(target=jump_between, args=(check, free, data_1_arr, data_2_arr, t))
254  p.start()
255  # continuously stream to AWG
256  while True:
257      dwError = spcm_dwSetParam_i32 (hCard, SPC_M2CMD, M2CMD_DATA_WAITDMA)
258      if dwError != ERR_OK:
259          if dwError == ERR_TIMEOUT:
260              sys.stdout.write ("... Timeout\n")
261          else:
262              sys.stdout.write ("... Error: {0:d}\n".format(dwError))
263              break;
264      else:
265          # start the card if the onboard buffer has been filled completely
266          spcm_dwGetParam_i32 (hCard, SPC_FILLSIZEPROMILLE, byref (lFillsize));
267          if lFillsize.value == 1000 and bStarted == False:
268              sys.stdout.write("... data has been transferred to board memory\n")
269              sys.stdout.write("\nStarting the card...\n")
270              dwError = spcm_dwSetParam_i32 (hCard, SPC_M2CMD, M2CMD_CARD_START |
                M2CMD_CARD_ENABLETRIGGER)
271
272              if dwError == ERR_TIMEOUT:
273                  spcm_dwSetParam_i32 (hCard, SPC_M2CMD, M2CMD_CARD_STOP)
274                  sys.stdout.write ("... Timeout at start\n")
275                  break;
276              bStarted = True
277          else:
278              sys.stdout.write ("... Fillsize: {0:d}/1000\n".format(lFillsize.
                value))
279
280          pass
281          spcm_dwGetParam_i32 (hCard, SPC_M2STATUS, byref (lStatus))
282          spcm_dwGetParam_i32 (hCard, SPC_DATA_AVAIL_USER_LEN, byref (
                lAvailUser_bytes))
283          spcm_dwGetParam_i32 (hCard, SPC_DATA_AVAIL_USER_POS, byref (LPCPos))
284          # calculate new data
285          if lAvailUser_bytes.value >= lNotifySize_bytes.value:
286              if check.value == 1:
287                  free.value=1
288                  # time_diff = timer()
289                  pData = (c_char * (qwBufferSize.value - LPCPos.value)).
                from_buffer (pvBuffer,
                LPCPos.value)
290
291                  lNumAvailSamples = (lNotifySize_bytes.value // lSetChannels.
                value) //
                lBytesPerSample.value
                # to avoid problems

```

B. Tweezers sorting algorithm

```
289         vCalcNewData (pnData, lSetChannels.value, qwSamplePos,
290                       with buffer wrap-
291                       arounds we fill only
292                       one notify size
293                       lNumAvailSamples,
294                       data_1_arr)
295
296     else:
297         free.value=1
298         # time_diff = timer()
299         pnData = (c_char * (qwBufferSize.value - lPCPos.value)).
300                 from_buffer (pvBuffer,
301                               lPCPos.value)
302         lNumAvailSamples = (lNotifySize_bytes.value // lSetChannels.
303                             value) //
304                             lBytesPerSample.value
305         # to avoid problems
306         # with buffer wrap-
307         # arounds we fill only
308         # one notify size
309         vCalcNewData (pnData, lSetChannels.value, qwSamplePos,
310                       lNumAvailSamples,
311                       data_2_arr)
312
313     spcm_dwSetParam_i32 (hCard, SPC_DATA_AVAIL_CARD_LEN,
314                         lNotifySize_bytes)
315
316     qwSamplePos += lNumAvailSamples
317     # time_diff=timer()-time_diff
318     # print(time_diff)
319     free.value=0
320
321     # Stop the card
322     # send the stop command
323     dwError = spcm_dwSetParam_i32 (hCard, SPC_M2CMD, M2CMD_CARD_STOP |
324                                   M2CMD_DATA_STOPDMA)
325
326     spcm_vClose (hCard);
327     # End the side process
328     p.terminate()
329     p.join()
```

Bibliography

- [1] King, H. and Millburn, J. *Geared to the Stars: The Evolution of Planetariums, Orreries, and Astronomical Clocks*. University of Toronto Press, 1978. [URL](#).
- [2] Johnson, T. H., Clark, S. R., and Jaksch, D. “What is a quantum simulator?” In: *EPJ Quantum Technology* vol. 1, no. 1 (July 2014). [URL](#).
- [3] Turing, A. M. “On Computable Numbers, with an Application to the Entscheidungsproblem.” In: *Proceedings of the London Mathematical Society* vol. s2-42, no. 1 (1937), pp. 230–265. [URL](#).
- [4] Foulkes, W. M. C. et al. “Quantum Monte Carlo simulations of solids.” In: *Rev. Mod. Phys.* vol. 73 (1 Jan. 2001), pp. 33–83. [URL](#).
- [5] Pollet, L. “Recent developments in quantum Monte Carlo simulations with applications for cold gases.” In: *Reports on Progress in Physics* vol. 75, no. 9 (Aug. 2012), p. 094501. [URL](#).
- [6] Orús, R. “Tensor networks for complex quantum systems.” In: *Nature Reviews Physics* vol. 1, no. 9 (Aug. 2019), pp. 538–550. [URL](#).
- [7] Carleo, G. and Troyer, M. “Solving the quantum many-body problem with artificial neural networks.” In: *Science* vol. 355, no. 6325 (Feb. 2017), pp. 602–606. [URL](#).
- [8] Carrasquilla, J. and Torlai, G. “How To Use Neural Networks To Investigate Quantum Many-Body Physics.” In: *PRX Quantum* vol. 2 (4 Nov. 2021), p. 040201. [URL](#).
- [9] Manin, Y. “Vychislimoe i nevychislimoe (Computable and Uncomputable in Russian).” In: *Soviet Radio* (1980), pp. 13–15.
- [10] Feynman, R. P. et al. “Simulating physics with computers.” In: *Int. j. Theor. phys* vol. 21, no. 6/7 (1982). [URL](#).
- [11] Lloyd, S. “Universal Quantum Simulators.” In: *Science* vol. 273, no. 5278 (1996), pp. 1073–1078. [URL](#).
- [12] Somaroo, S. et al. “Quantum Simulations on a Quantum Computer.” In: *Phys. Rev. Lett.* vol. 82 (26 June 1999), pp. 5381–5384. [URL](#).
- [13] Georgescu, I. M., Ashhab, S., and Nori, F. “Quantum simulation.” In: *Rev. Mod. Phys.* vol. 86 (1 Mar. 2014), pp. 153–185. [URL](#).
- [14] Buluta, I. and Nori, F. “Quantum Simulators.” In: *Science* vol. 326, no. 5949 (2009), pp. 108–111. [URL](#).

- [15] Altman, E. et al. “Quantum Simulators: Architectures and Opportunities.” In: *PRX Quantum* vol. 2 (1 Feb. 2021), p. 017003. [URL](#).
- [16] Anderson, M. H. et al. “Observation of Bose-Einstein Condensation in a Dilute Atomic Vapor.” In: *Science* vol. 269, no. 5221 (1995), pp. 198–201. [URL](#).
- [17] Bradley, C. C. et al. “Evidence of Bose-Einstein Condensation in an Atomic Gas with Attractive Interactions.” In: *Phys. Rev. Lett.* vol. 75 (9 Aug. 1995), pp. 1687–1690. [URL](#).
- [18] Davis, K. B. et al. “Bose-Einstein Condensation in a Gas of Sodium Atoms.” In: *Phys. Rev. Lett.* vol. 75 (22 Nov. 1995), pp. 3969–3973. [URL](#).
- [19] DeMarco, B. and Jin, D. S. “Onset of Fermi Degeneracy in a Trapped Atomic Gas.” In: *Science* vol. 285, no. 5434 (1999), pp. 1703–1706. [URL](#).
- [20] Proukakis, N. et al. *Quantum Gases : Finite Temperature and Non-Equilibrium Dynamics*. IMPERIAL COLLEGE PRESS, 2013. [URL](#).
- [21] Bloch, I. and Zoller, P. “Focus on Cold Atoms in Optical Lattices.” In: *New Journal of Physics* vol. 8, no. 8 (Aug. 2006). [URL](#).
- [22] Lewenstein, M. et al. “Ultracold atomic gases in optical lattices: mimicking condensed matter physics and beyond.” In: *Advances in Physics* vol. 56, no. 2 (2007), pp. 243–379. [URL](#).
- [23] Bloch, I., Dalibard, J., and Zwerger, W. “Many-body physics with ultracold gases.” In: *Rev. Mod. Phys.* vol. 80 (3 July 2008), pp. 885–964. [URL](#).
- [24] Bloch, I., Dalibard, J., and Nascimbène, S. “Quantum simulations with ultracold quantum gases.” In: *Nature Physics* vol. 8, no. 4 (Apr. 2012), pp. 267–276. [URL](#).
- [25] Gross, C. and Bloch, I. “Quantum simulations with ultracold atoms in optical lattices.” In: *Science* vol. 357, no. 6355 (2017), pp. 995–1001. [URL](#).
- [26] Muldoon, C. et al. “Control and manipulation of cold atoms in optical tweezers.” In: *New Journal of Physics* vol. 14, no. 7 (July 2012), p. 073051. [URL](#).
- [27] Kaufman, A. M. and Ni, K.-K. “Quantum science with optical tweezer arrays of ultracold atoms and molecules.” In: *Nature Physics* vol. 17, no. 12 (Nov. 2021), pp. 1324–1333. [URL](#).
- [28] Jaksch, D. et al. “Cold Bosonic Atoms in Optical Lattices.” In: *Phys. Rev. Lett.* vol. 81 (15 Oct. 1998), pp. 3108–3111. [URL](#).
- [29] Guidoni, L. and Verkerk, P. “Optical lattices: cold atoms ordered by light.” In: *Journal of Optics B: Quantum and Semiclassical Optics* vol. 1, no. 5 (Sept. 1999), R23–R45. [URL](#).
- [30] Chin, C. et al. “Feshbach resonances in ultracold gases.” In: *Rev. Mod. Phys.* vol. 82 (2 Apr. 2010), pp. 1225–1286. [URL](#).

- [31] Bakr, W. S. et al. “A quantum gas microscope for detecting single atoms in a Hubbard-regime optical lattice.” In: *Nature* vol. 462, no. 7269 (Nov. 2009), pp. 74–77. [URL](#).
- [32] Sherson, J. F. et al. “Single-atom-resolved fluorescence imaging of an atomic Mott insulator.” In: *Nature* vol. 467, no. 7311 (Aug. 2010), pp. 68–72. [URL](#).
- [33] Haller, E. et al. “Single-atom imaging of fermions in a quantum-gas microscope.” In: *Nature Physics* vol. 11, no. 9 (July 2015), pp. 738–742. [URL](#).
- [34] Omran, A. et al. “Microscopic Observation of Pauli Blocking in Degenerate Fermionic Lattice Gases.” In: *Phys. Rev. Lett.* vol. 115 (26 Dec. 2015), p. 263001. [URL](#).
- [35] Greiner, M. et al. “Quantum phase transition from a superfluid to a Mott insulator in a gas of ultracold atoms.” In: *Nature* vol. 415, no. 6867 (Jan. 2002), pp. 39–44. [URL](#).
- [36] Regal, C. A., Greiner, M., and Jin, D. S. “Observation of Resonance Condensation of Fermionic Atom Pairs.” In: *Phys. Rev. Lett.* vol. 92 (4 Jan. 2004), p. 040403. [URL](#).
- [37] Strinati, G. C. et al. “The BCS–BEC crossover: From ultra-cold Fermi gases to nuclear systems.” In: *Physics Reports* vol. 738 (Apr. 2018), pp. 1–76. [URL](#).
- [38] Schneider, U. et al. “Metallic and Insulating Phases of Repulsively Interacting Fermions in a 3D Optical Lattice.” In: *Science* vol. 322, no. 5907 (Dec. 2008), pp. 1520–1525. [URL](#).
- [39] Jördens, R. et al. “A Mott insulator of fermionic atoms in an optical lattice.” In: *Nature* vol. 455, no. 7210 (Sept. 2008), pp. 204–207. [URL](#).
- [40] Simon, J. et al. “Quantum simulation of antiferromagnetic spin chains in an optical lattice.” In: *Nature* vol. 472, no. 7343 (Apr. 2011), pp. 307–312. [URL](#).
- [41] Labuhn, H. et al. “Tunable two-dimensional arrays of single Rydberg atoms for realizing quantum Ising models.” In: *Nature* vol. 534, no. 7609 (June 2016), pp. 667–670. [URL](#).
- [42] Mazurenko, A. et al. “A cold-atom Fermi–Hubbard antiferromagnet.” In: *Nature* vol. 545, no. 7655 (May 2017), pp. 462–466. [URL](#).
- [43] Jepsen, P. N. et al. “Spin transport in a tunable Heisenberg model realized with ultracold atoms.” In: *Nature* vol. 588, no. 7838 (Dec. 2020), pp. 403–407. [URL](#).
- [44] Barontini, G. and Paternostro, M. “Ultra-cold single-atom quantum heat engines.” In: *New Journal of Physics* vol. 21, no. 6 (June 2019), p. 063019. [URL](#).
- [45] Niedenzu, W. et al. “Quantized refrigerator for an atomic cloud.” In: *Quantum* vol. 3 (June 2019), p. 155. [URL](#).
- [46] Bauer, J., Salomon, C., and Demler, E. “Realizing a Kondo-Correlated State with Ultracold Atoms.” In: *Phys. Rev. Lett.* vol. 111 (21 Nov. 2013), p. 215304. [URL](#).

- [47] Sundar, B. and Mueller, E. J. “Proposal to directly observe the Kondo effect through enhanced photoinduced scattering of cold fermionic and bosonic atoms.” In: *Phys. Rev. A* vol. 93 (2 Feb. 2016), p. 023635. [URL](#).
- [48] Rentrop, T. et al. “Observation of the Phononic Lamb Shift with a Synthetic Vacuum.” In: *Phys. Rev. X* vol. 6 (4 Nov. 2016), p. 041041. [URL](#).
- [49] Scazza, F. et al. *Repulsive Fermi and Bose Polarons in Quantum Gases*. 2022. [URL](#).
- [50] Mil, A. et al. “A scalable realization of local U(1) gauge invariance in cold atomic mixtures.” In: *Science* vol. 367, no. 6482 (Mar. 2020), pp. 1128–1130.
- [51] Aidelsburger, M. et al. “Cold atoms meet lattice gauge theory.” In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* vol. 380, no. 2216 (2022), p. 20210064. [URL](#).
- [52] Kasper, V. et al. “Universal quantum computation and quantum error correction with ultracold atomic mixtures.” In: *Quantum Science and Technology* vol. 7, no. 1 (Nov. 2021), p. 015008. [URL](#).
- [53] Schulze, T. A. et al. “Feshbach spectroscopy and dual-species Bose-Einstein condensation of ^{23}Na – ^{39}K mixtures.” In: *Phys. Rev. A* vol. 97 (2 Feb. 2018), p. 023623. [URL](#).
- [54] Hartmann, T. et al. “Feshbach resonances in $^{23}\text{Na} + ^{39}\text{K}$ mixtures and refined molecular potentials for the NaK molecule.” In: *Phys. Rev. A* vol. 99 (3 Mar. 2019), p. 032711. [URL](#).
- [55] Zhong, H.-S. et al. “Quantum computational advantage using photons.” In: *Science* vol. 370, no. 6523 (Dec. 2020), pp. 1460–1463. [URL](#).
- [56] Enk, S. J. van and Beenakker, C. W. J. “Measuring $\text{Tr}\rho^n$ on Single Copies of ρ Using Random Measurements.” In: *Phys. Rev. Lett.* vol. 108 (11 Mar. 2012), p. 110503. [URL](#).
- [57] Emerson, J., Alicki, R., and Życzkowski, K. “Scalable noise estimation with random unitary operators.” In: *Journal of Optics B: Quantum and Semiclassical Optics* vol. 7, no. 10 (Sept. 2005), S347–S352. [URL](#).
- [58] Cross, A. W. et al. “Validating quantum computers using randomized model circuits.” In: *Phys. Rev. A* vol. 100 (3 Sept. 2019), p. 032328. [URL](#).
- [59] Abeyesinghe, A. et al. “The mother of all protocols: restructuring quantum information’s family tree.” In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* vol. 465, no. 2108 (June 2009), pp. 2537–2563. [URL](#).
- [60] Hayden, P. et al. “Randomizing Quantum States: Constructions and Applications.” In: *Communications in Mathematical Physics* vol. 250, no. 2 (July 2004), pp. 371–391. [URL](#).

- [61] Sen, P. “Random measurement bases, quantum state distinction and applications to the hidden subgroup problem.” In: *21st Annual IEEE Conference on Computational Complexity (CCC’06)*. 2006, 14 pp.–287. [URL](#).
- [62] Oszmaniec, M. et al. “Random Bosonic States for Robust Quantum Metrology.” In: *Phys. Rev. X* vol. 6 (4 Dec. 2016), p. 041044. [URL](#).
- [63] Knips, L. “A Moment for Random Measurements.” In: *Quantum Views* vol. 4 (Nov. 2020), p. 47. [URL](#).
- [64] Notarnicola, S. et al. *A randomized measurement toolbox for Rydberg quantum technologies*. 2021. [URL](#).
- [65] Elben, A. et al. *The randomized measurement toolbox*. 2022. [URL](#).
- [66] Bhatt, R. P. et al. “Stochastic dynamics of a few sodium atoms in presence of a cold potassium cloud.” In: *Scientific Reports* vol. 12, no. 1 (Feb. 2022). [URL](#).
- [67] Stan, C. A. and Ketterle, W. “Multiple species atom source for laser-cooling experiments.” In: *Review of Scientific Instruments* vol. 76, no. 6 (2005), pp. 6–11.
- [68] Kraft, S. et al. “Formation of ultracold LiCs molecules.” In: *Journal of Physics B: Atomic, Molecular and Optical Physics* vol. 39 (Sept. 2006), S993.
- [69] Martí, G. et al. “Two-element Zeeman Slower for Rubidium and Lithium.” In: *Physical Review A* vol. 81 (Feb. 2010).
- [70] Kemp, S. L. et al. “Production and characterization of a dual species magneto-optical trap of cesium and ytterbium.” In: *Review of Scientific Instruments* vol. 87, no. 2 (2016), p. 023105.
- [71] Lamporesi, G. et al. “Compact high-flux source of cold sodium atoms.” In: *Review of Scientific Instruments* vol. 84, no. 6 (June 2013), p. 063102. [URL](#).
- [72] Tiecke, T. G. et al. “High-flux two-dimensional magneto-optical-trap source for cold lithium atoms.” In: *Physical Review A* vol. 80, no. July (2009), p. 013409.
- [73] Endres, M. et al. “Atom-by-atom assembly of defect-free one-dimensional cold atom arrays.” In: *Science* vol. 354, no. 6315 (2016), pp. 1024–1027. [URL](#).
- [74] Höcker, L. “Building up a modular Na-K quantum gas experiment.” Masters thesis. Universität Heidelberg, 2019.
- [75] Knudsen, M. “Die Gesetze der Molekularströmung und der inneren Reibungsströmung der Gase durch Röhren.” In: *Annalen der Physik* vol. 333, no. 1 (1909), pp. 75–130. [URL](#).
- [76] Roth, A. *Vacuum technology*. 3rd, updated, and enl. ed. Amsterdam ; New York : New York, NY, USA: North-Holland ; Distributors for the United States and Canada, Elsevier Science Pub. Co, 1990.
- [77] Riza, N. A. “Scanning heterodyne optical interferometers.” In: *Review of Scientific Instruments* vol. 67, no. 7 (1996), pp. 2466–2476. [URL](#).

- [78] Donley, E. A. et al. “Double-pass acousto-optic modulator system.” In: *Review of Scientific Instruments* vol. 76, no. 6 (2005), p. 063112. [URL](#).
- [79] Hänsch, T. W., Levenson, M. D., and Schawlow, A. L. “Complete Hyperfine Structure of a Molecular Iodine Line.” In: *Phys. Rev. Lett.* vol. 26 (16 Apr. 1971), pp. 946–949. [URL](#).
- [80] Preston, D. W. “Doppler-free saturated absorption: Laser spectroscopy.” In: *American Journal of Physics* vol. 64, no. 11 (1996), pp. 1432–1436. [URL](#).
- [81] Kilinc, J. “Starting a Na-K experiment for simulating quantum many-body phenomena.” Masters thesis. Universität Heidelberg, 2019.
- [82] Weis, A. and Derler, S. “Doppler modulation and Zeeman modulation: laser frequency stabilization without direct frequency modulation.” In: *Applied Optics* vol. 27, no. 13 (July 1988), p. 2662. [URL](#).
- [83] Impertro, A. F. “Laser system for magneto-optical cooling and trapping of potassium.” Bachelor thesis. Universität Heidelberg, 2017.
- [84] Neuhaus, L. et al. “PyRPL (Python Red Pitaya Lockbox) — An open-source software package for FPGA-controlled quantum optics experiments.” In: *2017 Conference on Lasers and Electro-Optics Europe European Quantum Electronics Conference (CLEO/Europe-EQEC)*. 2017, pp. 1–1. [URL](#).
- [85] Hänsch, T. and Schawlow, A. “Cooling of gases by laser radiation.” In: *Optics Communications* vol. 13, no. 1 (1975), pp. 68–69. [URL](#).
- [86] Cook, R. J. “Atomic motion in resonant radiation: An application of Ehrenfest’s theorem.” In: *Phys. Rev. A* vol. 20 (1 July 1979), pp. 224–228. [URL](#).
- [87] Podlecki, L. et al. “Radiation pressure on a two-level atom: an exact analytical approach.” In: *J. Opt. Soc. Am. B* vol. 35, no. 1 (Jan. 2018), pp. 127–132. [URL](#).
- [88] Lett, P. D. et al. “Optical molasses.” In: *J. Opt. Soc. Am. B* vol. 6, no. 11 (Nov. 1989), pp. 2084–2107. [URL](#).
- [89] Chu, S. et al. “Three-dimensional viscous confinement and cooling of atoms by resonance radiation pressure.” In: *Phys. Rev. Lett.* vol. 55 (1 July 1985), pp. 48–51. [URL](#).
- [90] Phillips, W. D. and Metcalf, H. “Laser Deceleration of an Atomic Beam.” In: *Phys. Rev. Lett.* vol. 48 (9 Mar. 1982), pp. 596–599. [URL](#).
- [91] Bagnato, V. S. et al. “Continuous Stopping and Trapping of Neutral Atoms.” In: *Phys. Rev. Lett.* vol. 58 (21 May 1987), pp. 2194–2197. [URL](#).
- [92] Stenholm, S. “The semiclassical theory of laser cooling.” In: *Rev. Mod. Phys.* vol. 58 (3 July 1986), pp. 699–739. [URL](#).
- [93] Wineland, D. J. and Itano, W. M. “Laser cooling of atoms.” In: *Phys. Rev. A* vol. 20 (4 Oct. 1979), pp. 1521–1540. [URL](#).

- [94] Castin, Y., Wallis, H., and Dalibard, J. “Limit of Doppler cooling.” In: *J. Opt. Soc. Am. B* vol. 6, no. 11 (Nov. 1989), pp. 2046–2057. [URL](#).
- [95] Sesko, D., Fan, C. G., and Wieman, C. E. “Production of a cold atomic vapor using diode-laser cooling.” In: *J. Opt. Soc. Am. B* vol. 5, no. 6 (June 1988), pp. 1225–1227. [URL](#).
- [96] Lett, P. D. et al. “Observation of Atoms Laser Cooled below the Doppler Limit.” In: *Phys. Rev. Lett.* vol. 61 (2 July 1988), pp. 169–172. [URL](#).
- [97] Shevy, Y. et al. “Bimodal speed distributions in laser-cooled atoms.” In: *Phys. Rev. Lett.* vol. 62 (10 Mar. 1989), pp. 1118–1121. [URL](#).
- [98] Ungar, P. J. et al. “Optical molasses and multilevel atoms: theory.” In: *J. Opt. Soc. Am. B* vol. 6, no. 11 (Nov. 1989), pp. 2058–2071. [URL](#).
- [99] Dalibard, J. and Cohen-Tannoudji, C. “Laser cooling below the Doppler limit by polarization gradients: simple theoretical models.” In: *J. Opt. Soc. Am. B* vol. 6, no. 11 (Nov. 1989), pp. 2023–2045. [URL](#).
- [100] Hopkins, S. A. and Durrant, A. V. “Parameters for polarization gradients in three-dimensional electromagnetic standing waves.” In: *Phys. Rev. A* vol. 56 (5 Nov. 1997), pp. 4012–4022. [URL](#).
- [101] Salomon, C. et al. “Laser Cooling of Cesium Atoms Below $3 \mu\text{K}$.” In: *Europhysics Letters (EPL)* vol. 12, no. 8 (Aug. 1990), pp. 683–688. [URL](#).
- [102] Landini, M. et al. “Sub-Doppler laser cooling of potassium atoms.” In: *Phys. Rev. A* vol. 84 (4 Oct. 2011), p. 043432. [URL](#).
- [103] Raab, E. L. et al. “Trapping of Neutral Sodium Atoms with Radiation Pressure.” In: *Phys. Rev. Lett.* vol. 59 (23 Dec. 1987), pp. 2631–2634. [URL](#).
- [104] Monroe, C. et al. “Very cold trapped atoms in a vapor cell.” In: *Phys. Rev. Lett.* vol. 65 (13 Sept. 1990), pp. 1571–1574. [URL](#).
- [105] Foot, C. and Foot, D. *Atomic Physics*. Oxford Master Series in Physics. OUP Oxford, 2005. [URL](#).
- [106] Dieckmann, K. et al. “Two-dimensional magneto-optical trap as a source of slow atoms.” In: *Phys. Rev. A* vol. 58 (5 Nov. 1998), pp. 3891–3895. [URL](#).
- [107] Landini, M. et al. “Direct evaporative cooling of ^{39}K atoms to Bose-Einstein condensation.” In: *Phys. Rev. A* vol. 86 (3 Sept. 2012), p. 033421. [URL](#).
- [108] Roux, K. et al. “Compact bulk-machined electromagnets for quantum gas experiments.” In: *SciPost Phys.* vol. 6 (4 2019), p. 48. [URL](#).
- [109] Ketterle, W., Durfee, D. S., and Stamper-Kurn, D. M. *Making, probing and understanding Bose-Einstein condensates*. 1999. [URL](#).
- [110] Bradley, C. C., Sackett, C. A., and Hulet, R. G. “Bose-Einstein Condensation of Lithium: Observation of Limited Condensate Number.” In: *Phys. Rev. Lett.* vol. 78 (6 Feb. 1997), pp. 985–989. [URL](#).

- [111] Andrews, M. R. et al. “Propagation of Sound in a Bose-Einstein Condensate.” In: *Phys. Rev. Lett.* vol. 79 (4 July 1997), pp. 553–556. [URL](#).
- [112] Andrews, M. R. et al. “Direct, Nondestructive Observation of a Bose Condensate.” In: *Science* vol. 273, no. 5271 (1996), pp. 84–87. [URL](#).
- [113] Kadlecik, S. et al. “Nondestructive spatial heterodyne imaging of cold atoms.” In: *Optics Letters* vol. 26, no. 3 (Feb. 2001), p. 137. [URL](#).
- [114] Turner, L. D. et al. “Off-resonant defocus-contrast imaging of cold atoms.” In: *Optics Letters* vol. 29, no. 3 (Feb. 2004), p. 232. [URL](#).
- [115] Turner, L. D., Domen, K. F. E. M., and Scholten, R. E. “Diffraction-contrast imaging of cold atoms.” In: *Phys. Rev. A* vol. 72 (3 Sept. 2005), p. 031403. [URL](#).
- [116] Pappa, M. et al. “Ultra-sensitive atom imaging for matter-wave optics.” In: *New Journal of Physics* vol. 13, no. 11 (Nov. 2011), p. 115012. [URL](#).
- [117] Reinhard, A. et al. “Dark-ground imaging of high optical thickness atom clouds.” In: *Optics Communications* vol. 324 (2014), pp. 30–33. [URL](#).
- [118] Streed, E. W. et al. “Absorption imaging of a single atom.” In: *Nature Communications* vol. 3, no. 1 (Jan. 2012). [URL](#).
- [119] Picken, C. J., Legaie, R., and Pritchard, J. D. “Single atom imaging with an sCMOS camera.” In: *Applied Physics Letters* vol. 111, no. 16 (2017), p. 164102. [URL](#).
- [120] Bergschneider, A. et al. “Spin-resolved single-atom imaging of ${}^6\text{Li}$ in free space.” In: *Phys. Rev. A* vol. 97 (6 June 2018), p. 063613. [URL](#).
- [121] Covey, J. P. et al. “2000-Times Repeated Imaging of Strontium Atoms in Clock-Magic Tweezer Arrays.” In: *Phys. Rev. Lett.* vol. 122 (17 May 2019), p. 173201. [URL](#).
- [122] Krieger, J. “Zeeman-Slower und Experimentsteuerung für das NaLi-Experiment.” Diploma thesis. Universität Heidelberg, 2008.
- [123] Starkey, P. T. et al. “A scripted control system for autonomous hardware-timed experiments.” In: *Review of Scientific Instruments* vol. 84, no. 8 (2013), p. 085111. [URL](#).
- [124] STARKEY, P. T. “A software framework for control and automation of precisely timed experiments.” PhD thesis. Monash University, 2019. [URL](#).
- [125] BILLINGTON, C. J. “State-dependent forces in cold quantum gases.” PhD thesis. Monash University, 2019. [URL](#).
- [126] Hu, Z. and Kimble, H. J. “Observation of a single atom in a magneto-optical trap.” In: *Optics Letters* vol. 19, no. 22 (Nov. 1994), p. 1888. [URL](#).
- [127] Ott, H. “Single atom detection in ultracold quantum gases: a review of current progress.” In: *Reports on Progress in Physics* vol. 79, no. 5 (Apr. 2016), p. 054401. [URL](#).

- [128] Pezzè, L. et al. “Quantum metrology with nonclassical states of atomic ensembles.” In: *Rev. Mod. Phys.* vol. 90 (3 Sept. 2018), p. 035005. [URL](#).
- [129] Stroescu, I., Hume, D. B., and Oberthaler, M. K. “Double-well atom trap for fluorescence detection at the Heisenberg limit.” In: *Phys. Rev. A* vol. 91 (1 Jan. 2015), p. 013412. [URL](#).
- [130] Hüper, A. et al. “Number-resolved preparation of mesoscopic atomic ensembles.” In: *New Journal of Physics* vol. 23, no. 11 (Nov. 2021), p. 113046. [URL](#).
- [131] Serwane, F. et al. “Deterministic Preparation of a Tunable Few-Fermion System.” In: *Science* vol. 332, no. 6027 (Apr. 2011), pp. 336–338. [URL](#).
- [132] Zürn, G. et al. “Fermionization of Two Distinguishable Fermions.” In: *Phys. Rev. Lett.* vol. 108 (7 Feb. 2012), p. 075303. [URL](#).
- [133] Zürn, G. et al. “Pairing in Few-Fermion Systems with Attractive Interactions.” In: *Phys. Rev. Lett.* vol. 111 (17 Oct. 2013), p. 175302. [URL](#).
- [134] Weber, C. et al. “Single Cs atoms as collisional probes in a large Rb magneto-optical trap.” In: *Phys. Rev. A* vol. 82 (4 Oct. 2010), p. 042722. [URL](#).
- [135] Feng, Z. et al. “Dating glacier ice of the last millennium by quantum technology.” In: *Proceedings of the National Academy of Sciences* vol. 116, no. 18 (Apr. 2019), pp. 8781–8786. [URL](#).
- [136] Chen, C. Y. et al. “Ultrasensitive Isotope Trace Analyses with a Magneto-Optical Trap.” In: *Science* vol. 286, no. 5442 (Nov. 1999), pp. 1139–1141. [URL](#).
- [137] Jiang, W. et al. “ ^{39}Ar Detection at the 10^{-16} Isotopic Abundance Level with Atom Trap Trace Analysis.” In: *Phys. Rev. Lett.* vol. 106 (10 Mar. 2011), p. 103001. [URL](#).
- [138] Ebser, S. et al. “ ^{39}Ar dating with small samples provides new key constraints on ocean ventilation.” In: *Nature Communications* vol. 9, no. 1 (Nov. 2018). [URL](#).
- [139] Ruschewitz, F. et al. “Statistical investigations on single trapped neutral atoms.” In: *Europhysics Letters (EPL)* vol. 34, no. 9 (June 1996), pp. 651–656. [URL](#).
- [140] Haubrich, D. et al. “Observation of individual neutral atoms in magnetic and magneto-optical traps.” In: *Europhysics Letters (EPL)* vol. 34, no. 9 (June 1996), pp. 663–668. [URL](#).
- [141] Hill, S. B. and McClelland, J. J. “Atoms on demand: Fast, deterministic production of single Cr atoms.” In: *Applied Physics Letters* vol. 82, no. 18 (2003), pp. 3128–3130. eprint: <https://doi.org/10.1063/1.1572539>. [URL](#).
- [142] Yoon, S. et al. “Definitive number of atoms on demand: Controlling the number of atoms in a few-atom magneto-optical trap.” In: *Applied Physics Letters* vol. 88, no. 21 (2006), p. 211104. eprint: <https://doi.org/10.1063/1.2206118>. [URL](#).
- [143] Hume, D. B. et al. “Accurate Atom Counting in Mesoscopic Ensembles.” In: *Phys. Rev. Lett.* vol. 111 (25 Dec. 2013), p. 253001. [URL](#).

- [144] Schliederer, M. et al. “Single-atom counting in a two-color magneto-optical trap.” In: *Phys. Rev. A* vol. 103 (3 Mar. 2021), p. 033308. [URL](#).
- [145] Hoth, G. W., Donley, E. A., and Kitching, J. “Atom number in magneto-optic traps with millimeter scale laser beams.” In: *Opt. Lett.* vol. 38, no. 5 (Mar. 2013), pp. 661–663. [URL](#).
- [146] Fort, C. et al. “Cooling mechanisms in potassium magneto-optical traps.” In: *The European Physical Journal D - Atomic, Molecular and Optical Physics* vol. 3, no. 2 (Aug. 1998), pp. 113–118. [URL](#).
- [147] Strobel, H. et al. “Fisher information and entanglement of non-Gaussian spin states.” In: *Science* vol. 345, no. 6195 (2014), pp. 424–427. [URL](#).
- [148] Muessel, W. et al. “Scalable Spin Squeezing for Quantum-Enhanced Magnetometry with Bose-Einstein Condensates.” In: *Physical Review Letters* vol. 113, no. 10 (2014), p. 103004. [URL](#).
- [149] Cont, R. and Bouchaud, J.-P. “Herd behavior and aggregate fluctuations in financial markets.” In: *Macroeconomic Dynamics* vol. 4, no. 2 (June 2000), pp. 170–196. [URL](#).
- [150] Kucsko, G. et al. “Nanometre-scale thermometry in a living cell.” In: *Nature* vol. 500, no. 7460 (2013), pp. 54–58. [URL](#).
- [151] Grusdt, F. and Demler, E. “New theoretical approaches to Bose polarons.” In: *Quantum matter at ultralow temperatures*. 2016, p. 325.
- [152] Gluza, M. et al. “Quantum Field Thermal Machines.” In: *PRX Quantum* vol. 2 (3 July 2021), p. 030310. [URL](#).
- [153] Gröblacher, S. et al. “Observation of strong coupling between a micromechanical resonator and an optical cavity field.” In: *Nature* vol. 460, no. 7256 (2009), pp. 724–727. [URL](#).
- [154] Gleyzes, S. et al. “Quantum jumps of light recording the birth and death of a photon in a cavity.” In: *Nature* vol. 446, no. 7133 (2007), pp. 297–300. [URL](#).
- [155] Saira, O.-P. et al. “Test of the Jarzynski and Crooks Fluctuation Relations in an Electronic System.” In: *Phys. Rev. Lett.* vol. 109 (18 Oct. 2012), p. 180601.
- [156] Maier, C. et al. “Environment-Assisted Quantum Transport in a 10-qubit Network.” In: *Phys. Rev. Lett.* vol. 122 (5 Feb. 2019), p. 050501. [URL](#).
- [157] Chiu, C. S. et al. “String patterns in the doped Hubbard model.” In: *Science* vol. 365, no. 6450 (2019), pp. 251–256.
- [158] Scelle, R. et al. “Motional Coherence of Fermions Immersed in a Bose Gas.” In: *Physical Review Letters* vol. 111, no. 7 (Aug. 2013), p. 070401.
- [159] Jørgensen, N. B. et al. “Observation of attractive and repulsive polarons in a Bose-Einstein condensate.” In: *Physical Review Letters* vol. 117 (2016), p. 055302.

- [160] Yan, Z. Z. et al. “Bose polarons near quantum criticality.” In: *Science* vol. 368, no. 6487 (2020), pp. 190–194.
- [161] Zohar, E., Cirac, J. I., and Reznik, B. “Quantum simulations of lattice gauge theories using ultracold atoms in optical lattices.” In: *Reports on Progress in Physics* vol. 79, no. 1 (2015), p. 014401. [URL](#).
- [162] Kasper, V. et al. “Implementing quantum electrodynamics with ultracold atomic systems.” In: *New Journal of Physics* vol. 19, no. 2 (Feb. 2017), p. 023030.
- [163] Hohmann, M. et al. “Individual Tracer Atoms in an Ultracold Dilute Gas.” In: *Physical Review Letters* vol. 118, no. 26 (2017), pp. 1–5.
- [164] Bouton, Q. et al. “Single-Atom Quantum Probes for Ultracold Gases Boosted by Nonequilibrium Spin Dynamics.” In: *Physical Review X* vol. 10, no. 1 (2020), p. 11018. [URL](#).
- [165] Santos, M. S. et al. “Simultaneous trapping of two different atomic species in a vapor-cell magneto-optical trap.” In: *Phys. Rev. A* vol. 52 (6 Dec. 1995), R4340–R4343. [URL](#).
- [166] Einstein, A. *Quantentheorie des einatomigen idealen Gases*. Dec. 2005. [URL](#).
- [167] Bose. “Plancks Gesetz und Lichtquantenhypothese.” In: *Zeitschrift fur Physik* vol. 26, no. 1 (Dec. 1924), pp. 178–181. [URL](#).
- [168] Robinson, J. E. “Note on the Bose-Einstein Integral Functions.” In: *Phys. Rev.* vol. 83 (3 Aug. 1951), pp. 678–679. [URL](#).
- [169] Hohenberg, P. C. and Halperin, B. I. “Theory of dynamic critical phenomena.” In: *Rev. Mod. Phys.* vol. 49 (3 July 1977), pp. 435–479. [URL](#).
- [170] Ensher, J. R. et al. “Bose-Einstein Condensation in a Dilute Gas: Measurement of Energy and Ground-State Occupation.” In: *Phys. Rev. Lett.* vol. 77 (25 Dec. 1996), pp. 4984–4987. [URL](#).
- [171] Shiozaki, R. F. et al. “Measuring the heat capacity in a Bose-Einstein condensation using global variables.” In: *Phys. Rev. A* vol. 90 (4 Oct. 2014), p. 043640. [URL](#).
- [172] Yukalov, V. I. “Basics of Bose-Einstein condensation.” In: *Physics of Particles and Nuclei* vol. 42, no. 3 (May 2011), pp. 460–513. [URL](#).
- [173] Penrose, O. and Onsager, L. “Bose-Einstein Condensation and Liquid Helium.” In: *Phys. Rev.* vol. 104 (3 Nov. 1956), pp. 576–584. [URL](#).
- [174] Girardeau, M. D. “Off-Diagonal Long-Range Order and Generalized Bose Condensation.” In: *Journal of Mathematical Physics* vol. 6, no. 7 (1965), pp. 1083–1098. [URL](#).
- [175] Ueda, M. *Fundamentals and New Frontiers of Bose-Einstein Condensation*. WORLD SCIENTIFIC, 2010. [URL](#).

- [176] Pethick, C. J. and Smith, H. *Bose–Einstein Condensation in Dilute Gases*. 2nd ed. Cambridge University Press, 2008. [URL](#).
- [177] Pitaevskii, L. et al. *Bose-Einstein Condensation*. International Series of Monographs on Physics. Clarendon Press, 2003. [URL](#).
- [178] Dalfovo, F. et al. “Theory of Bose-Einstein condensation in trapped gases.” In: *Rev. Mod. Phys.* vol. 71 (3 Apr. 1999), pp. 463–512. [URL](#).
- [179] Courteille, P. W., Bagnato, V. S., and Yukalov, V. I. “Bose-Einstein condensation of trapped atomic gases.” In: *Laser Physics* vol. 11 (2001), pp. 659–800. [URL](#).
- [180] Leggett, A. J. “Bose-Einstein condensation in the alkali gases: Some fundamental concepts.” In: *Rev. Mod. Phys.* vol. 73 (2 Apr. 2001), pp. 307–356. [URL](#).
- [181] Ricci, L. et al. “A current-carrying coil design with improved liquid cooling arrangement.” In: *Review of Scientific Instruments* vol. 84, no. 6 (2013), p. 065115. [URL](#).
- [182] Castilho, P. C. M. et al. “A compact experimental machine for studying tunable Bose–Bose superfluid mixtures.” In: *Laser Physics Letters* vol. 16, no. 3 (Feb. 2019), p. 035501. [URL](#).
- [183] Walker, T., Sesko, D., and Wieman, C. “Collective behavior of optically trapped neutral atoms.” In: *Phys. Rev. Lett.* vol. 64 (4 Jan. 1990), pp. 408–411. [URL](#).
- [184] Prentiss, M. et al. “Atomic-density-dependent losses in an optical trap.” In: *Optics Letters* vol. 13, no. 6 (June 1988), p. 452. [URL](#).
- [185] Ketterle, W. et al. “High densities of cold atoms in a dark spontaneous-force optical trap.” In: *Phys. Rev. Lett.* vol. 70 (15 Apr. 1993), pp. 2253–2256. [URL](#).
- [186] Townsend, C. G. et al. “Phase-space density in the magneto-optical trap.” In: *Phys. Rev. A* vol. 52 (2 Aug. 1995), pp. 1423–1440. [URL](#).
- [187] Petrich, W. et al. “Behavior of atoms in a compressed magneto-optical trap.” In: *Journal of the Optical Society of America B* vol. 11, no. 8 (Aug. 1994), p. 1332. [URL](#).
- [188] DePue, M. T. et al. “Transient compression of a MOT and high intensity fluorescent imaging of optically thick clouds of atoms.” In: *Optics Communications* vol. 180, no. 1 (2000), pp. 73–79. [URL](#).
- [189] Migdall, A. L. et al. “First Observation of Magnetically Trapped Neutral Atoms.” In: *Phys. Rev. Lett.* vol. 54 (24 June 1985), pp. 2596–2599. [URL](#).
- [190] Majorana, E. “Atomi orientati in campo magnetico variabile.” In: *Il Nuovo Cimento* vol. 9, no. 2 (Feb. 1932), pp. 43–50. [URL](#).
- [191] Majorana, E. “Oriented Atoms in a Variable Magnetic Field.” In: *Scientific Papers of Ettore Majorana: A New Expanded Edition*. Ed. by Cifarelli, L. Cham: Springer International Publishing, 2020, pp. 77–88. [URL](#).

- [192] Lin, Y.-J. et al. “Rapid production of ^{87}Rb Bose-Einstein condensates in a combined magnetic and optical potential.” In: *Phys. Rev. A* vol. 79 (6 June 2009), p. 063631. [URL](#).
- [193] Ketterle, W. and Druten, N. V. “Evaporative Cooling of Trapped Atoms.” In: *Advances In Atomic, Molecular, and Optical Physics*. Elsevier, 1996, pp. 181–236. [URL](#).
- [194] Modugno, G. et al. “Bose-Einstein Condensation of Potassium Atoms by Sympathetic Cooling.” In: *Science* vol. 294, no. 5545 (2001), pp. 1320–1322. [URL](#).
- [195] Daley, A. J., Fedichev, P. O., and Zoller, P. “Single-atom cooling by superfluid immersion: A nondestructive method for qubits.” In: *Phys. Rev. A* vol. 69 (2 Feb. 2004), p. 022306. [URL](#).
- [196] Hartmann, T. “An experiment apparatus for the production of ultracold bosonic dipolar ground state $^{23}\text{Na}^{39}\text{K}$ molecules and Feshbach spectroscopy in a cold mixture of ^{23}Na and ^{39}K .” PhD thesis. Universität Hannover, 2018. [URL](#).
- [197] Serafini, S. “Realization of a magnetic trap for the production of sodium Bose-Einstein Condensates.” Masters thesis. Universit Trento, 2013.
- [198] Deng, L. et al. “Bose Einstein Condensation of Sodium Atoms in a TOP Trap.” In: May 1998.
- [199] Stam, K. M. R. van der et al. “Large atom number Bose-Einstein condensate of sodium.” In: *Review of Scientific Instruments* vol. 78, no. 1 (Jan. 2007), p. 013102. [URL](#).
- [200] Colzi, G. et al. “Sub-Doppler cooling of sodium atoms in gray molasses.” In: *Phys. Rev. A* vol. 93 (2 Feb. 2016), p. 023421. [URL](#).
- [201] Chu, S. et al. “Experimental Observation of Optically Trapped Atoms.” In: *Phys. Rev. Lett.* vol. 57 (3 July 1986), pp. 314–317. [URL](#).
- [202] Grimm, R., Weidemüller, M., and Ovchinnikov, Y. B. “Optical Dipole Traps for Neutral Atoms.” In: *Advances In Atomic, Molecular, and Optical Physics*. Elsevier, 2000, pp. 95–170. [URL](#).
- [203] Stamper-Kurn, D. M. et al. “Optical Confinement of a Bose-Einstein Condensate.” In: *Phys. Rev. Lett.* vol. 80 (10 Mar. 1998), pp. 2027–2030. [URL](#).
- [204] Adams, C. S. et al. “Evaporative Cooling in a Crossed Dipole Trap.” In: *Phys. Rev. Lett.* vol. 74 (18 May 1995), pp. 3577–3580. [URL](#).
- [205] Schlosser, N. et al. “Sub-poissonian loading of single atoms in a microscopic dipole trap.” In: *Nature* vol. 411, no. 6841 (June 2001), pp. 1024–1027. [URL](#).
- [206] Schlosser, N., Reymond, G., and Grangier, P. “Collisional Blockade in Microscopic Optical Dipole Traps.” In: *Phys. Rev. Lett.* vol. 89 (2 June 2002), p. 023005. [URL](#).

- [207] Barredo, D. et al. “An atom-by-atom assembler of defect-free arbitrary two-dimensional atomic arrays.” In: *Science* vol. 354, no. 6315 (2016), pp. 1021–1023. [URL](#).
- [208] Haar, A. “Der Massbegriff in der Theorie der Kontinuierlichen Gruppen.” In: *Annals of Mathematics* vol. 34, no. 1 (1933), pp. 147–169. [URL](#) (visited on 04/25/2022).
- [209] Hurwitz, A. “Über die Erzeugung der Invarianten durch Integration.” In: *Mathematische Werke: Zweiter Band Zahlentheorie Algebra und Geometrie*. Basel: Springer Basel, 1963, pp. 546–564. [URL](#).
- [210] Alfsen, E. M. “A simplified constructive proof of the existence and uniqueness of Haar measure.” In: *Mathematica Scandinavica* vol. 12, no. 1 (1963), pp. 106–116. [URL](#) (visited on 04/25/2022).
- [211] Dyson, F. J. “Statistical Theory of the Energy Levels of Complex Systems. I.” In: *Journal of Mathematical Physics* vol. 3, no. 1 (1962), pp. 140–156. [URL](#).
- [212] Gogolin, C. and Eisert, J. “Equilibration, thermalisation, and the emergence of statistical mechanics in closed quantum systems.” In: *Reports on Progress in Physics* vol. 79, no. 5 (Apr. 2016), p. 056001. [URL](#).
- [213] Hosur, P. et al. “Chaos in quantum channels.” In: *Journal of High Energy Physics* vol. 2016, no. 2 (Feb. 2016). [URL](#).
- [214] Hayden, P. and Preskill, J. “Black holes as mirrors: quantum information in random subsystems.” In: *Journal of High Energy Physics* vol. 2007, no. 09 (Sept. 2007), pp. 120–120. [URL](#).
- [215] Knill, E. *Approximation by Quantum Circuits*. 1995. [URL](#).
- [216] Dankert, C. et al. “Exact and approximate unitary 2-designs and their application to fidelity estimation.” In: *Phys. Rev. A* vol. 80 (1 July 2009), p. 012304. [URL](#).
- [217] Gross, D., Audenaert, K., and Eisert, J. “Evenly distributed unitaries: On the structure of unitary designs.” In: *Journal of Mathematical Physics* vol. 48, no. 5 (2007), p. 052104. [URL](#).
- [218] Roy, A. and Scott, A. J. “Unitary designs and codes.” In: *Designs, Codes and Cryptography* vol. 53, no. 1 (Apr. 2009), pp. 13–31. [URL](#).
- [219] Seymour, P. and Zaslavsky, T. “Averaging sets: A generalization of mean values and spherical designs.” In: *Advances in Mathematics* vol. 52, no. 3 (1984), pp. 213–240. [URL](#).
- [220] Scott, A. J. “Optimizing quantum process tomography with unitary 2 -designs.” In: *Journal of Physics A: Mathematical and Theoretical* vol. 41, no. 5 (Feb. 2008), p. 055308. [URL](#).
- [221] Kane, D. M. “Small designs for path-connected spaces and path-connected homogeneous spaces.” In: *Transactions of the American Mathematical Society* vol. 367, no. 9 (Apr. 2015), pp. 6387–6414. [URL](#).

- [222] Alagic, G., Majenz, C., and Russell, A. “Efficient Simulation of Random States and Random Unitaries.” In: *Advances in Cryptology – EUROCRYPT 2020*. Springer International Publishing, 2020, pp. 759–787. [URL](#).
- [223] Cleve, R. et al. “Near-linear constructions of exact unitary 2-designs.” In: *Quantum Information and Computation* vol. 16, no. 9&10 (July 2016), pp. 721–756. [URL](#).
- [224] Bannai, E. et al. “On the explicit constructions of certain unitary t -designs.” In: *Journal of Physics A: Mathematical and Theoretical* vol. 52, no. 49 (Nov. 2019), p. 495301. [URL](#).
- [225] Bannai, E. et al. *Explicit construction of exact unitary designs*. 2020. [URL](#).
- [226] Nakata, Y. et al. “Quantum Circuits for Exact Unitary t -Designs and Applications to Higher-Order Randomized Benchmarking.” In: *PRX Quantum* vol. 2 (3 Sept. 2021), p. 030339. [URL](#).
- [227] Bennett, C. H. et al. “Mixed-state entanglement and quantum error correction.” In: *Phys. Rev. A* vol. 54 (5 Nov. 1996), pp. 3824–3851. [URL](#).
- [228] DiVincenzo, D., Leung, D., and Terhal, B. “Quantum data hiding.” In: *IEEE Transactions on Information Theory* vol. 48, no. 3 (Mar. 2002), pp. 580–598. [URL](#).
- [229] Bombin, H. and Martin-Delgado, M. A. “Entanglement distillation protocols and number theory.” In: *Phys. Rev. A* vol. 72 (3 Sept. 2005), p. 032313. [URL](#).
- [230] Low, R. A. “Pseudo-randomness and Learning in Quantum Computation.” In: (2010). [URL](#).
- [231] Aharonov, D., Kitaev, A., and Nisan, N. “Quantum Circuits with Mixed States.” In: *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*. STOC ’98. Dallas, Texas, USA: Association for Computing Machinery, 1998, pp. 20–30. [URL](#).
- [232] Harrow, A. W. and Low, R. A. “Efficient Quantum Tensor Product Expanders and k -Designs.” In: Springer Berlin Heidelberg, 2009, pp. 548–561. [URL](#).
- [233] Harrow, A. W. and Low, R. A. “Random Quantum Circuits are Approximate 2-designs.” In: *Communications in Mathematical Physics* vol. 291, no. 1 (July 2009), pp. 257–302. [URL](#).
- [234] Diniz, I. T. and Jonathan, D. “Comment on “Random Quantum Circuits are Approximate 2-designs” by A.W. Harrow and R.A. Low (Commun. Math. Phys. 291, 257–302 (2009)).” In: *Communications in Mathematical Physics* vol. 304, no. 1 (Mar. 2011), pp. 281–293. [URL](#).
- [235] Brandão, F. G. S. L., Harrow, A. W., and Horodecki, M. “Local Random Quantum Circuits are Approximate Polynomial-Designs.” In: *Communications in Mathematical Physics* vol. 346, no. 2 (Aug. 2016), pp. 397–434. [URL](#).

- [236] Ówikiński, P. et al. “Local random quantum circuits are approximate polynomial-designs: numerical results.” In: *Journal of Physics A: Mathematical and Theoretical* vol. 46, no. 30 (July 2013), p. 305301. [URL](#).
- [237] Turner, P. S. and Markham, D. “Derandomizing Quantum Circuits with Measurement-Based Unitary Designs.” In: *Phys. Rev. Lett.* vol. 116 (20 May 2016), p. 200501. [URL](#).
- [238] Brandão, F. G. S. L., Harrow, A. W., and Horodecki, M. “Efficient Quantum Pseudorandomness.” In: *Phys. Rev. Lett.* vol. 116 (17 Apr. 2016), p. 170502. [URL](#).
- [239] Nakata, Y. et al. “Unitary 2-designs from random X- and Z-diagonal unitaries.” In: *Journal of Mathematical Physics* vol. 58, no. 5 (2017), p. 052203. [URL](#).
- [240] Nakata, Y. et al. “Efficient Quantum Pseudorandomness with Nearly Time-Independent Hamiltonian Dynamics.” In: *Phys. Rev. X* vol. 7 (2 Apr. 2017), p. 021006. [URL](#).
- [241] Mezher, R. et al. “Efficient quantum pseudorandomness with simple graph states.” In: *Phys. Rev. A* vol. 97 (2 Feb. 2018), p. 022333. [URL](#).
- [242] Harrow, A. W. and Mehraban, S. A. *Approximate unitary t -designs by short random quantum circuits using nearest-neighbor and long-range gates*. 2018. [URL](#).
- [243] Mezher, R. et al. *Efficient approximate unitary t -designs from partially invertible universal sets and their application to quantum speedup*. 2019. [URL](#).
- [244] Haferkamp, J. et al. *Quantum homeopathy works: Efficient unitary designs with a system-size independent number of non-Clifford gates*. 2020. [URL](#).
- [245] Emerson, J. et al. “Pseudo-Random Unitary Operators for Quantum Information Processing.” In: *Science* vol. 302, no. 5653 (2003), pp. 2098–2100. [URL](#).
- [246] Banchi, L., Burgarth, D., and Kastoryano, M. J. “Driven Quantum Dynamics: Will It Blend?” In: *Phys. Rev. X* vol. 7 (4 Oct. 2017), p. 041015. [URL](#).
- [247] Vermersch, B. et al. “Unitary n -designs via random quenches in atomic Hubbard and spin models: Application to the measurement of Rényi entropies.” In: *Phys. Rev. A* vol. 97 (2 Feb. 2018), p. 023604. [URL](#).
- [248] Brydges, T. et al. “Probing Rényi entanglement entropy via randomized measurements.” In: *Science* vol. 364, no. 6437 (2019), pp. 260–263. [URL](#).
- [249] Li, J. et al. “Experimental Implementation of Efficient Quantum Pseudorandomness on a 12-Spin System.” In: *Phys. Rev. Lett.* vol. 123 (3 July 2019), p. 030502. [URL](#).
- [250] Roberts, D. A. and Yoshida, B. “Chaos and complexity by design.” In: *Journal of High Energy Physics* vol. 2017, no. 4 (Apr. 2017), p. 121. [URL](#).
- [251] Sieberer, L. M. et al. “Digital quantum simulation, Trotter errors, and quantum chaos of the kicked top.” In: *npj Quantum Information* vol. 5, no. 1 (Sept. 2019). [URL](#).

- [252] Ohliger, M., Nesme, V., and Eisert, J. “Efficient and feasible state tomography of quantum many-body systems.” In: *New Journal of Physics* vol. 15, no. 1 (Jan. 2013), p. 015024. [URL](#).
- [253] Hauke, P. et al. “Can one trust quantum simulators?” In: *Reports on Progress in Physics* vol. 75, no. 8 (July 2012), p. 082401. [URL](#).
- [254] James, D. F. V. et al. “Measurement of qubits.” In: *Phys. Rev. A* vol. 64 (5 Oct. 2001), p. 052312. [URL](#).
- [255] Cramer, M. et al. “Efficient quantum state tomography.” In: *Nature Communications* vol. 1, no. 1 (Dec. 2010). [URL](#).
- [256] Mohseni, M., RezaKhani, A. T., and Lidar, D. A. “Quantum-process tomography: Resource analysis of different strategies.” In: *Phys. Rev. A* vol. 77 (3 Mar. 2008), p. 032322. [URL](#).
- [257] Flammia, S. T. and Liu, Y.-K. “Direct Fidelity Estimation from Few Pauli Measurements.” In: *Phys. Rev. Lett.* vol. 106 (23 June 2011), p. 230501. [URL](#).
- [258] Lanyon, B. P. et al. “Efficient tomography of a quantum many-body system.” In: *Nature Physics* vol. 13, no. 12 (Sept. 2017), pp. 1158–1162. [URL](#).
- [259] Knill, E. et al. “Randomized benchmarking of quantum gates.” In: *Phys. Rev. A* vol. 77 (1 Jan. 2008), p. 012307. [URL](#).
- [260] Onorati, E., Werner, A. H., and Eisert, J. “Randomized Benchmarking for Individual Quantum Gates.” In: *Phys. Rev. Lett.* vol. 123 (6 Aug. 2019), p. 060501. [URL](#).
- [261] Derbyshire, E. et al. “Randomized benchmarking in the analogue setting.” In: *Quantum Science and Technology* vol. 5, no. 3 (Apr. 2020), p. 034001. [URL](#).
- [262] Elben, A. et al. “Cross-Platform Verification of Intermediate Scale Quantum Devices.” In: *Phys. Rev. Lett.* vol. 124 (1 Jan. 2020), p. 010504. [URL](#).
- [263] Amico, L. et al. “Entanglement in many-body systems.” In: *Rev. Mod. Phys.* vol. 80 (2 May 2008), pp. 517–576. [URL](#).
- [264] Horodecki, R. et al. “Quantum entanglement.” In: *Rev. Mod. Phys.* vol. 81 (2 June 2009), pp. 865–942. [URL](#).
- [265] Calabrese, P. and Cardy, J. “Evolution of entanglement entropy in one-dimensional systems.” In: *Journal of Statistical Mechanics: Theory and Experiment* vol. 2005, no. 04 (Apr. 2005), P04010. [URL](#).
- [266] Fagotti, M. and Calabrese, P. “Evolution of entanglement entropy following a quantum quench: Analytic results for the XY chain in a transverse magnetic field.” In: *Phys. Rev. A* vol. 78 (1 July 2008), p. 010306. [URL](#).
- [267] Page, D. N. “Average entropy of a subsystem.” In: *Phys. Rev. Lett.* vol. 71 (9 Aug. 1993), pp. 1291–1294. [URL](#).

- [268] Eisert, J., Cramer, M., and Plenio, M. B. “Colloquium: Area laws for the entanglement entropy.” In: *Rev. Mod. Phys.* vol. 82 (1 Feb. 2010), pp. 277–306. [URL](#).
- [269] Abanin, D. A. et al. “Colloquium: Many-body localization, thermalization, and entanglement.” In: *Rev. Mod. Phys.* vol. 91 (2 May 2019), p. 021001. [URL](#).
- [270] Heyl, M. “Dynamical quantum phase transitions: a review.” In: *Reports on Progress in Physics* vol. 81, no. 5 (Apr. 2018), p. 054001. [URL](#).
- [271] Bergschneider, A. et al. “Experimental characterization of two-particle entanglement through position and momentum correlations.” In: *Nature Physics* vol. 15, no. 7 (Apr. 2019), pp. 640–644. [URL](#).
- [272] Daley, A. J. et al. “Measuring Entanglement Growth in Quench Dynamics of Bosons in an Optical Lattice.” In: *Phys. Rev. Lett.* vol. 109 (2 July 2012), p. 020505. [URL](#).
- [273] Abanin, D. A. and Demler, E. “Measuring Entanglement Entropy of a Generic Many-Body System with a Quantum Switch.” In: *Phys. Rev. Lett.* vol. 109 (2 July 2012), p. 020504. [URL](#).
- [274] Pichler, H. et al. “Thermal versus entanglement entropy: a measurement protocol for fermionic atoms with a quantum gas microscope.” In: *New Journal of Physics* vol. 15, no. 6 (June 2013), p. 063003. [URL](#).
- [275] Islam, R. et al. “Measuring entanglement entropy in a quantum many-body system.” In: *Nature* vol. 528, no. 7580 (Dec. 2015), pp. 77–83. [URL](#).
- [276] Kaufman, A. M. et al. “Quantum thermalization through entanglement in an isolated many-body system.” In: *Science* vol. 353, no. 6301 (2016), pp. 794–800. [URL](#).
- [277] Elben, A. et al. “Rényi Entropies from Random Quenches in Atomic Hubbard and Spin Models.” In: *Phys. Rev. Lett.* vol. 120 (5 Feb. 2018), p. 050406. [URL](#).
- [278] Elben, A. et al. “Statistical correlations between locally randomized measurements: A toolbox for probing entanglement in many-body quantum states.” In: *Phys. Rev. A* vol. 99 (5 May 2019), p. 052323. [URL](#).
- [279] Elben, A. et al. “Mixed-State Entanglement from Local Randomized Measurements.” In: *Phys. Rev. Lett.* vol. 125 (20 Nov. 2020), p. 200501. [URL](#).
- [280] Walter, M., Gross, D., and Eisert, J. “Multipartite Entanglement.” In: *Quantum Information*. John Wiley and Sons, Ltd, 2016. Chap. 14, pp. 293–330. [URL](#).
- [281] Szalay, S. “Multipartite entanglement measures.” In: *Phys. Rev. A* vol. 92 (4 Oct. 2015), p. 042329. [URL](#).
- [282] Ketterer, A., Wyderka, N., and Gühne, O. “Characterizing Multipartite Entanglement with Moments of Random Correlations.” In: *Phys. Rev. Lett.* vol. 122 (12 Mar. 2019), p. 120505. [URL](#).
- [283] Ketterer, A., Wyderka, N., and Gühne, O. “Entanglement characterization using quantum designs.” In: *Quantum* vol. 4 (Sept. 2020), p. 325. [URL](#).

- [284] Knips, L. et al. “Multipartite entanglement analysis from random correlations.” In: *npj Quantum Information* vol. 6, no. 1 (June 2020). [URL](#).
- [285] Imai, S. et al. “Bound Entanglement from Randomized Measurements.” In: *Phys. Rev. Lett.* vol. 126 (15 Apr. 2021), p. 150501. [URL](#).
- [286] Pezzé, L. and Smerzi, A. “Entanglement, Nonlinear Dynamics, and the Heisenberg Limit.” In: *Phys. Rev. Lett.* vol. 102 (10 Mar. 2009), p. 100401. [URL](#).
- [287] Hyllus, P. et al. “Fisher information and multiparticle entanglement.” In: *Phys. Rev. A* vol. 85 (2 Feb. 2012), p. 022321. [URL](#).
- [288] Yu, M. et al. “Experimental estimation of the quantum Fisher information from randomized measurements.” In: *Phys. Rev. Research* vol. 3 (4 Nov. 2021), p. 043122. [URL](#).
- [289] Rath, A. et al. “Quantum Fisher Information from Randomized Measurements.” In: *Phys. Rev. Lett.* vol. 127 (26 Dec. 2021), p. 260501. [URL](#).
- [290] Landau, L. D. “On the theory of phase transitions.” In: *Zh. Eksp. Teor. Fiz.* vol. 7 (1937), pp. 19–32. [URL](#).
- [291] Ginzburg, V. “Some remarks on phase transitions of the second kind and the microscopic theory of ferroelectric materials.” In: *Soviet Phys. Solid State* vol. 2 (1961), pp. 1824–1834.
- [292] Wilson, K. G. and Kogut, J. “The renormalization group and the ϵ - expansion.” In: *Physics Reports* vol. 12, no. 2 (1974), pp. 75–199. [URL](#).
- [293] Wen, X.-G. “Topological orders in rigid states.” In: *International Journal of Modern Physics B* vol. 04, no. 02 (1990), pp. 239–271. [URL](#).
- [294] Kitaev, A. and Preskill, J. “Topological Entanglement Entropy.” In: *Phys. Rev. Lett.* vol. 96 (11 Mar. 2006), p. 110404. [URL](#).
- [295] Levin, M. and Wen, X.-G. “Detecting Topological Order in a Ground State Wave Function.” In: *Phys. Rev. Lett.* vol. 96 (11 Mar. 2006), p. 110405. [URL](#).
- [296] Gu, Z.-C. and Wen, X.-G. “Tensor-entanglement-filtering renormalization approach and symmetry-protected topological order.” In: *Phys. Rev. B* vol. 80 (15 Oct. 2009), p. 155131. [URL](#).
- [297] Pollmann, F. et al. “Symmetry protection of topological phases in one-dimensional quantum spin systems.” In: *Phys. Rev. B* vol. 85 (7 Feb. 2012), p. 075125. [URL](#).
- [298] Chen, X. et al. “Symmetry protected topological orders and the group cohomology of their symmetry group.” In: *Phys. Rev. B* vol. 87 (15 Apr. 2013), p. 155114. [URL](#).
- [299] Haegeman, J. et al. “Order Parameter for Symmetry-Protected Phases in One Dimension.” In: *Phys. Rev. Lett.* vol. 109 (5 July 2012), p. 050402. [URL](#).

- [300] Pollmann, F and Turner, A. M. “Detection of symmetry-protected topological phases in one dimension.” In: *Phys. Rev. B* vol. 86 (12 Sept. 2012), p. 125441. [URL](#).
- [301] Elben, A. et al. “Many-body topological invariants from randomized measurements in synthetic quantum matter.” In: *Science Advances* vol. 6, no. 15 (2020), eaaz3666. [URL](#).
- [302] Cian, Z.-P. et al. “Many-Body Chern Number from Statistical Correlations of Randomized Measurements.” In: *Phys. Rev. Lett.* vol. 126 (5 Feb. 2021), p. 050501. [URL](#).
- [303] Vojta, M. “Quantum phase transitions.” In: *Reports on Progress in Physics* vol. 66, no. 12 (Nov. 2003), pp. 2069–2110. [URL](#).
- [304] Zanardi, P. and Paunković, N. “Ground state overlap and quantum phase transitions.” In: *Phys. Rev. E* vol. 74 (3 Sept. 2006), p. 031123. [URL](#).
- [305] Gu, S.-J. “Fidelity approach to quantum phase transitions.” In: *International Journal of Modern Physics B* vol. 24, no. 23 (Sept. 2010), pp. 4371–4458. [URL](#).
- [306] Wang, L. et al. “Fidelity Susceptibility Made Simple: A Unified Quantum Monte Carlo Approach.” In: *Phys. Rev. X* vol. 5 (3 July 2015), p. 031007. [URL](#).
- [307] Zhang, J. et al. “Detection of Quantum Critical Points by a Probe Qubit.” In: *Phys. Rev. Lett.* vol. 100 (10 Mar. 2008), p. 100501. [URL](#).
- [308] Jensen, R. V. “Quantum chaos.” In: *Nature* vol. 355 (Jan. 1992), pp. 311–318. [URL](#).
- [309] Gomez, I. S., Losada, M., and Lombardi, O. “About the Concept of Quantum Chaos.” In: *Entropy* vol. 19, no. 5 (2017). [URL](#).
- [310] Larkin, A. and Ovchinnikov, Y. N. “Quasiclassical method in the theory of superconductivity.” In: *Sov Phys JETP* vol. 28, no. 6 (1969), pp. 1200–1205. [URL](#).
- [311] Maldacena, J., Shenker, S. H., and Stanford, D. “A bound on chaos.” In: *Journal of High Energy Physics* vol. 2016, no. 8 (Aug. 2016). [URL](#).
- [312] Leone, L., Oliviero, S. F. E., and Hamma, A. “Isospectral Twirling and Quantum Chaos.” In: *Entropy* vol. 23, no. 8 (Aug. 2021), p. 1073. [URL](#).
- [313] Chan, A., De Luca, A., and Chalker, J. T. “Solution of a Minimal Model for Many-Body Quantum Chaos.” In: *Phys. Rev. X* vol. 8 (4 Nov. 2018), p. 041019. [URL](#).
- [314] Kos, P., Ljubotina, M., and Prosen, T. ž. “Many-Body Quantum Chaos: Analytic Connection to Random Matrix Theory.” In: *Phys. Rev. X* vol. 8 (2 June 2018), p. 021062. [URL](#).
- [315] Leone, L. et al. “Quantum Chaos is Quantum.” In: *Quantum* vol. 5 (May 2021), p. 453. [URL](#).
- [316] Joshi, L. K. et al. “Probing Many-Body Quantum Chaos with Quantum Simulators.” In: *Phys. Rev. X* vol. 12 (1 Jan. 2022), p. 011018. [URL](#).

- [317] Sekino, Y. and Susskind, L. “Fast scramblers.” In: *Journal of High Energy Physics* vol. 2008, no. 10 (Oct. 2008), pp. 065–065. [URL](#).
- [318] Iyoda, E. and Sagawa, T. “Scrambling of quantum information in quantum many-body systems.” In: *Phys. Rev. A* vol. 97 (4 Apr. 2018), p. 042330. [URL](#).
- [319] Cotler, J. et al. “Chaos, complexity, and random matrices.” In: *Journal of High Energy Physics* vol. 2017, no. 11 (Nov. 2017). [URL](#).
- [320] Swingle, B. and Chowdhury, D. “Slow scrambling in disordered quantum systems.” In: *Phys. Rev. B* vol. 95 (6 Feb. 2017), p. 060201. [URL](#).
- [321] Swingle, B. et al. “Measuring the scrambling of quantum information.” In: *Phys. Rev. A* vol. 94 (4 Oct. 2016), p. 040302. [URL](#).
- [322] Da ģ, C. B. and Duan, L.-M. “Detection of out-of-time-order correlators and information scrambling in cold atoms: Ladder- XX model.” In: *Phys. Rev. A* vol. 99 (5 May 2019), p. 052322. [URL](#).
- [323] Gärttner, M. et al. “Measuring out-of-time-order correlations and multiple quantum spectra in a trapped-ion quantum magnet.” In: *Nature Physics* vol. 13, no. 8 (May 2017), pp. 781–786. [URL](#).
- [324] Li, J. et al. “Measuring Out-of-Time-Order Correlators on a Nuclear Magnetic Resonance Quantum Simulator.” In: *Phys. Rev. X* vol. 7 (3 July 2017), p. 031011. [URL](#).
- [325] Wei, K. X. et al. “Emergent Prethermalization Signatures in Out-of-Time Ordered Correlations.” In: *Phys. Rev. Lett.* vol. 123 (9 Aug. 2019), p. 090605. [URL](#).
- [326] Bohrdt, A. et al. “Scrambling and thermalization in a diffusive quantum many-body system.” In: *New Journal of Physics* vol. 19, no. 6 (June 2017), p. 063001. [URL](#).
- [327] Yao, N. Y. et al. *Interferometric Approach to Probing Fast Scrambling*. 2016. [URL](#).
- [328] Zhu, G., Hafezi, M., and Grover, T. “Measurement of many-body chaos using a quantum clock.” In: *Phys. Rev. A* vol. 94 (6 Dec. 2016), p. 062329. [URL](#).
- [329] Hashizume, T. et al. “Deterministic Fast Scrambling with Neutral Atom Arrays.” In: *Phys. Rev. Lett.* vol. 126 (20 May 2021), p. 200603. [URL](#).
- [330] Swingle, B. and Yunger Halpern, N. “Resilience of scrambling measurements.” In: *Phys. Rev. A* vol. 97 (6 June 2018), p. 062113. [URL](#).
- [331] Yoshida, B. and Yao, N. Y. “Disentangling Scrambling and Decoherence via Quantum Teleportation.” In: *Phys. Rev. X* vol. 9 (1 Jan. 2019), p. 011006. [URL](#).
- [332] Landsman, K. A. et al. “Verified quantum information scrambling.” In: *Nature* vol. 567, no. 7746 (Mar. 2019), pp. 61–65. [URL](#).
- [333] Khurana, D., Krithika, V. R., and Mahesh, T. S. *Unambiguous measurement of information scrambling in a hierarchical star-topology system*. 2019. [URL](#).

- [334] González Alonso, J. R., Yunger Halpern, N., and Dressel, J. “Out-of-Time-Ordered-Correlator Quasiprobabilities Robustly Witness Scrambling.” In: *Phys. Rev. Lett.* vol. 122 (4 Feb. 2019), p. 040404. [URL](#).
- [335] Blocher, P. D. et al. *Measuring out-of-time-ordered correlation functions without reversing time evolution*. 2020. [URL](#).
- [336] Vermersch, B. et al. “Probing Scrambling Using Statistical Correlations between Randomized Measurements.” In: *Phys. Rev. X* vol. 9 (2 June 2019), p. 021061. [URL](#).
- [337] Joshi, M. K. et al. “Quantum Information Scrambling in a Trapped-Ion Quantum Simulator with Tunable Range Interactions.” In: *Phys. Rev. Lett.* vol. 124 (24 June 2020), p. 240505. [URL](#).
- [338] Nie, X. et al. *Detecting scrambling via statistical correlations between randomized measurements on an NMR quantum simulator*. 2019. [URL](#).
- [339] Tomkovic, J. et al. “Experimental observation of the Poincaré-Birkhoff scenario in a driven many-body quantum system.” In: *Phys. Rev. A* vol. 95 (1 Jan. 2017), p. 011602. [URL](#).
- [340] Vermersch, B. et al. “Unitary n -designs via random quenches in atomic Hubbard and spin models: Application to the measurement of Rényi entropies.” In: *Phys. Rev. A* vol. 97 (2 Feb. 2018), p. 023604. [URL](#).
- [341] Schwinger, J. “ON ANGULAR MOMENTUM.” In: (Jan. 1952). [URL](#).
- [342] Strobel, H. “Fisher Information and entanglement of non-Gaussian spin states.” PhD thesis. Universität Heidelberg, 2016. [URL](#).
- [343] Preiss, P. M. “Atomic Bose-Hubbard Systems With Single-Particle Control.” PhD thesis. Harvard University, 2016. [URL](#).
- [344] Liu, W. V., Wilczek, F., and Zoller, P. “Spin-dependent Hubbard model and a quantum phase transition in cold atoms.” In: *Phys. Rev. A* vol. 70 (3 Sept. 2004), p. 033603. [URL](#).
- [345] Lepori, L. et al. “Effective Control of Chemical Potentials by Rabi Coupling with RF-Fields in Ultracold Mixtures.” In: *Condensed Matter* vol. 3, no. 2 (2018). [URL](#).
- [346] Wei, R. and Mueller, E. J. “Magnetic-field dependence of Raman coupling in alkali-metal atoms.” In: *Phys. Rev. A* vol. 87 (4 Apr. 2013), p. 042514. [URL](#).
- [347] Murmann, S. et al. “Two Fermions in a Double Well: Exploring a Fundamental Building Block of the Hubbard Model.” In: *Phys. Rev. Lett.* vol. 114 (8 Feb. 2015), p. 080402. [URL](#).

Erklärung

Ich versichere, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Heidelberg, den 9 May 2022

.....

Rohit Prasad Bhatt