

Department of Physics (Quantum Optics and Laser Science)

**Tools and Methods for the Distillation
of Entanglement in Continuous
Variable Quantum Optics**

Alvaro Felipe Feito Boirac

Thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy of the University of London and the Diploma of Membership of Imperial College,
September 30th 2008.

Abstract

Entanglement is a crucial resource to process and transmit information surpassing the limits of what is possible in classical physics. However environmental noise (or decoherence) puts limits on the performance quantum states can deliver. To overcome these shortcomings, distillation offers a protocol in which local operations on a number of states deliver a strongly entangled state (with little noise). In the broad field of quantum optics the continuous variables of light have been studied for over half a century. This grants the existence of numerous mathematical and experimental tools suitable to explore distillation. The development of some tools for the practical realization of such protocols constitutes the core of this research.

The first part of the thesis presents improvements to existing protocols aimed at optimizing optical resources and enhancing success probabilities. To this end I study new configurations of existing protocols and evaluate the advantages of using measurement devices with higher efficiencies. The collaboration with The Ultra Fast Group at the University of Oxford has lead to the first steps towards the experimental implementation of the aforementioned ideas. Many unanswered questions were met along the way: Is the purity of these optical states sufficient for distillation? Are these measuring devices operating in the quantum regime and if so how well? How could one rigorously characterize an entanglement increase in a continuous variables experiment? The second part of this thesis deals with the theoretical tools necessary to answer these questions and to develop them further experimentally. Among the answers I present new experimental entanglement characterization tools, and new developments in detector tomography.

Acknowledgements

I would like to express my gratitude to my supervisor Martin Plenio for his patience, dedication and insight all throughout my thesis. I am indebted to him and to Jens Eisert for their previous work which is the foundation on which this investigation was built. The numerous stimulating discussions with them and with Kenneth Pregnell have made this process more fun, interesting and rich. In addition, I have learnt a lot from my supporting and great colleagues: Nic, Moritz, Oscar, Fernando, Konrad, David, Terry, Stefan, Doug, Laleh, Michael, Alex, Masaki, Miguel, Philip, Chris, Alessio and Koenraad, thank you all. My collaborators from the University of Oxford and the Max Planck Research Group in Erlangen have been extremely welcoming and generous sharing their labs, spaces and knowledge. In particular I'd like to thank Ian Walmsley, Christine Silberhorn and all those in their groups including Wolfgang, Hendrik, Jeff, Brian and Graciana.

I gratefully acknowledge financial support from the EPSRC for the last three years and from QIP-IRC for funding all expenses that have made collaborations, presentations and conferences possible. I would also like to mention some of my early mentors and inspirational teachers such as Adán Cabello and Arkadi Levanyuk which gave me the tools and confidence to pursue my postgraduate studies. I would also like to thank Sarah for her unconditional support and curiosity and, finally, my family who made all the steps leading to this thesis possible.

To Jacques Sicard

Contents

1	Introduction	8
1.1	Entanglement: an Introduction	8
1.2	Monogamy, Decoherence and Other Challenges	11
1.3	Practical Quantum Optics CV Distillation	33
2	Continuous Variable Distillation of Entanglement with Linear Optics	36
2.1	Motivation	36
2.2	The $(\rho^{(i)}, \rho^{(i)})$ Protocol	37
2.3	Probabilities	44
2.4	Non-Gaussian Resource States	50
2.5	The $(\rho^{(i)}, \rho^{(0)})$ Protocol	61
2.6	Work in Progress: Homodyne Distillation Pumping	71
2.7	Open Problems and Conclusion	79
3	First Steps Towards Experimental Procrustean Distillation of Entanglement	81
3.1	Introduction	81
3.2	Squeezed Vacuum States, How useful are they?	83
3.3	Ideal Photon Subtraction	84
3.4	Chosen Subtraction Setup	91
3.5	Entanglement Quantification	92

3.6	Characterization Strategy	95
4	Detector Tomography	98
4.1	Introduction	99
4.2	The Detectors	102
4.3	Reconstruction	107
4.4	Ill Conditioning and Regularisation	116
4.5	Detailed Assumptions	130
4.6	Conclusion	134
5	Conclusion	136
5.1	Summary of Thesis Achievements	136
5.2	Future Work	137
A	Tomography Appendix	140
B	Homodyne Measurement Appendix	151
B.1	Explicit Calculation for the Gaussian Projective Measurement of a Gaussian State	152
C	Quantavo Appendix	162
C.1	The Quantavo <i>Maple</i> Toolbox	163
C.2	Getting Started	164
C.3	Toolbox	167
C.4	Practical Example	189
.1	Dictionary of Procedures:	197
.2	Copyright and Disclaimer	218
.3	Maple Code	219
	Bibliography	275

Publications

- A. Cabello, A. Feito, A.L. Linares
Bell's inequalities with realistic noise for polarization-entangled photons Phys. Rev. A **72**, 052112 (2005)
- J. Eisert, M. B. Plenio, D. E. Browne, S. Scheel and A. Feito
On the experimental feasibility of continuous-variable optical entanglement distillation, Optics and Spectroscopy, Volume 103, Number 2 p 173-177 (2007)
- H. B. Coldenstrodt-Ronge, J. S. Lundeen, K. L. Pregnell, A. Feito, B. J. Smith, C. Silberhorn, J. Eisert, M. B. Plenio and I. A. Walmsley. *A proposed testbed for detector tomography*, Journal of Modern Optics (2008)
- A. Feito
A Maple toolbox for quantum optics in Fock space, arXiv.org: quant-ph/0806.2171 (2008),
- J.S. Lundeen, A. Feito, H. Coldenstrodt-Ronge, T.C. Ralph, K.L.Pregnell, Ch. Silberhorn, J. Eisert, M.B. Plenio, and I.A. Walmsley, *Measuring measurement* , Nature Physics (2008)

Abbreviations

CV	-	Continuous Variables
LOCC	-	Local Operations and Classical Communication
GLOCC	-	Gaussian Local Operations and Classical Communication
BS	-	Beam Splitter
APD	-	Avalanche Photo Diode
LO	-	Linear Optics
TMD	-	Time Multiplexing Detector
PD	-	Probability Distribution
PDC	-	Parametric Down Conversion
SDP	-	Semi Definite Programming

1

Introduction

The thesis is organised as follows: This first chapter is a brief introduction to the literature and the context. Subsequently I introduce the motivation behind this thesis and my contributions to the field. The next chapter (2) deals with improvements to current entanglement distillation protocols. It can be read independently from the other chapters and constitutes the first block of the thesis. Chapter 3 is a more detailed analysis of the experimental and theoretical challenges involved in the actual implementation and can be read independently too. Chapter 4 deals with the concept of detector tomography. Even though this tomography is used to characterise the tools from chapter 3 it can also be read independently. Finally some software tools I developed are included for reference in the appendix but are not needed to understand the rest.

1.1 Entanglement: an Introduction

Since the publication of the famous paper in 1935 by Einstein Podolsky and Rosen [EPR35] the way we picture entanglement and how we talk about it have evolved greatly. It was described as “spooky action at a distance” and later as “statistical correlations revealing non-locality” [Pop95] to have widespread acceptance from the mid-1990’s on as a possible resource for communication and computation having its own

unit, the “ebit”. I will briefly review its origins, uses and why distillation of entanglement appears as a very natural question in the field of quantum information.

1.1.1 What is Entanglement?

Entanglement is a property which is only encountered in the realm of quantum mechanics and exhibits some striking features. One way to reveal its effects involves two distant particles having previously interacted in a specific way. When these particles are measured, the formalism seems to suggest that measuring one particle affects the properties of the distant particle instantaneously. It is worth noting that these effects cannot be seen in a single run experiment but become apparent in the statistics relating the results of many measurements. In that regard the exact interpretation of this result requires a careful examination of the many assumptions involved in the formalism [Bel87, HR07]. However, irrespective of the interpretation, the coexistence of concepts such as locality, realism, logic or probability sets is brought into question or may even have to be abandoned [EPR35, Har93, Pop95, Ish97].

To give a more formal definition of entanglement we must note that two key elements give rise to the structure of entanglement:

- First the mathematical object that describes the properties of an isolated physical system is a ray embedded in a Hilbert space. Therefore linear combinations of states also describe physical systems.
- Secondly the way to describe two or more particles or systems makes use of the direct product. The states defined by linear combinations of direct products of rays can give rise to the correlations defining entanglement.

To be more precise and general about the definition of entanglement in the bipartite case we can refer to [Wer89] :

Definition 1. *The state described by the density matrix ρ on $\mathcal{H}_A \otimes \mathcal{H}_B$ is said to be separable iff it can be written as the convex combination:*

$$\rho_{AB} = \sum_{i=1}^N p_i \rho_A^i \otimes \rho_B^i, \quad 0 \leq p_i \leq 1, \quad \sum_{i=1}^N p_i = 1$$

or can be approximated in trace norm by the states of that form. Otherwise the state is said to be entangled.

(Note that the multipartite generalisation of the definition is more or less straightforward although the concept of N-separability adds to its complexity [HHHH07]).

1.1.2 Why Use Entanglement?

The existence or not of these correlations between distant particles has been the subject of heated debate for most of the 20th century [EPR35, Boh35, Sch35, Ein53, Bel64, Bel87]. However, in the 1980's a series of experiments in quantum optics confirmed many of the properties entanglement was expected to display [CHSH69, APR81, AGR82, ADR82, SA88]. Giving a practical use to this correlations has sparked the imagination of numerous scientists in computation, information science and physics.

Among the notable applications of entanglement for communication we find the so called ‘teleportation’ [BBC⁺93a], secret key distribution for quantum cryptography [Eke91] or quantum dense coding [BW92]. In the field of computation various algorithms have been found for which entangled states allow some computational problems to be solved far faster than is possible using classical resources alone [Ste98, Sho97, Sho96]. However, the exact role entanglement plays (or not) in this speed-up is still a subject of active investigation.

Additionally the epistemological and philosophical questions involved in the transition from classical physics to quantum physics can be elucidated as we enhance our control of quantum systems and refine our means of interacting with them.

1.2 Monogamy, Decoherence and Other Challenges

It must be noted that the very structure of entanglement, while enabling useful correlations between some particles, can also introduce unwanted ones. Entanglement between any number of particles is generated by means of pairwise or multiple-particle interactions. It should therefore be obvious that when unwanted interactions (for instance with the environment) occur, entanglement beyond the reach of our measurement devices can be generated. Two key concepts help us grasp the problem behind this phenomenon: *Monogamy of entanglement* and *decoherence*.

Loosely speaking monogamy of entanglement [CKW00] makes reference to the following: If two particles A and B share certain amount of entanglement and one becomes entangled with a third one the pairwise entanglement between A and B will weaken. As a limiting case when two particles are maximally entangled, they cannot be entangled with a third one in that same degree of freedom. More specifically we can write the state of these three particles and look at the entanglement between different parts of the system. For instance, the entanglement between A and the pair BC can be quantified by a certain entanglement measure $E(A(BC))$. This quantity then limits the entanglement A can have with B and C taken individually. For qubits this relation can be expressed by the inequality $E(AB) + E(AC) \leq E(A(BC))$, where $E()$ is some entanglement measure (or originally the square of the concurrence)[CKW00].

It is often impossible to control all the interactions between a system and the environment (air, optical fibres, atoms, radiation, etc). Therefore if our system becomes entangled with third particles the entanglement between the two (or more) particles that interest us will eventually decrease. Similar results hold for more particles, higher dimensions and for certain infinite dimensional systems making it a general problem [KW04, AI06, AI07, AI08] .

It is however not always possible to describe or have access to every particle that interacts with our system and we often adopt a more coarse grained point of view. For our quantum communication purposes, particles will need to be sent through a channel (optical fibre or air for instance). The inevitable interaction with the particles of the channel cannot be described in full detail but will be effectively described as a decoherence process. Tracing out the channel will also reveal a state whose correlations have ‘leaked’ into the environment [NC00, Pre98]. Put differently, part of the entanglement that the original system contained is now shared with inaccessible particles.

On top of the decoherence noisy channels introduce, the creation of entanglement can be a noisy process in the first place. Entangled states are often the outcome of probabilistic events and need to be described with mixed states similar to those having experienced decoherence. These two considerations (partial accessibility and uncertainty in the state generation) imply that usually the available states are less entangled than the pure state description we often encounter. We can loosely say that entanglement is either difficult to produce or is lost in inaccessible parts of the system.

To summarise, the mathematical structure of quantum mechanics reveals the interesting and useful property of entanglement. However the correct description of experimentally available states shows that entanglement is hard to produce, manipulate and preserve.

1.2.1 Fighting Decoherence with Distillation

As we have seen above, if we want to exploit the power pure entangled states have to offer, we need to fight decoherence and mixedness (or lack of purity in the quantum sense). In other words we ultimately want to have highly entangled pure states as a resource. Many areas in Quantum Information can help achieve this goal. Notably better sources of particles will contribute to the solution (be it photons [MLSW08], trapped ions [LDM⁺03], Nuclear Magnetic Resonance [LKC⁺02] or any other implementation). Isolating our qubits from the environment is obviously another interesting path.

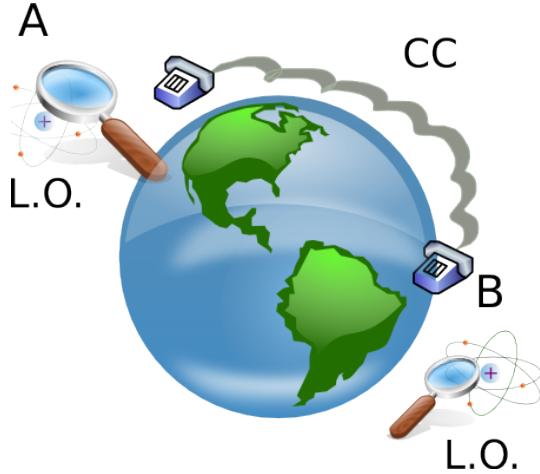


Figure 1.1: LOCC: Local Operations and Classical Communications are the operations Alice and Bob are allowed to use in their quantum communication tasks. LOCC precludes the coherent exchange of quantum particles between the two or more parties.

This idea has lead to clever arrangements of particles which protect the coherent information [Kit02, FKLW01].

However, we must remember that if these particles are to travel long distances even highly entangled pure states will degrade through a noisy channel. We may nevertheless choose to tame the effects of this noise with accurate and controlled local quantum processes in the distant labs. These are commonly referred to as local quantum operations (LO). Local operations are aimed at avoiding the typical long-distance decoherence. Additionally, standard telecom technologies allow perfect classical communication which can help coordinate the quantum operations in each lab (see fig. 1.1). The use of these resources is commonly described under the acronym LOCC: Local Operations and Classical Communication. Another crucial aspect of LOCC-operations is its fundamental relationship to entanglement. LOCC-operations between two non-entangled particles cannot create entanglement. In fact one can define classical correlations between two quantum systems as those arising from LOCC-operations alone [EJPP00, CLP01].

The compelling question is therefore: can LOCC fight decoherence or even increase

the entanglement if it is present? There is a partial affirmative answer to this question. Distillation can achieve this but with certain restrictions. This will not be a deterministic process and not all states will be distillable [Ken98a, Rai99, PV07]. Purity and entanglement-increasing LOCC-operations are generally classified depending on the resource states and the target states. Increasing the entanglement of pure states is called *entanglement concentration*. When this operation encompasses the use of mixed states too it is called *entanglement distillation*. Also, *purification* refers to the process of increasing the purity which can be quantified with measures such as $\text{Tr} \{ \rho^2 \}$ or the Von Neumann entropy.

Entanglement Concentration A simple example introduced in 1995 will help us grasp the spirit of distillation. The first insight leading towards distillation was that LOCC operations on a state not violating any bell inequality could turn it, probabilistically, into a state revealing non-locality [Pop95]. To understand this, consider a state of the form:

$$\alpha|00\rangle + \beta|11\rangle.$$

Without loss of generality we will restrict ourselves to the case in which $\alpha > \beta$. Let Alice add an ancillary qubit in the state $|0\rangle$ forming the state:

$$\alpha \underbrace{|00\rangle}_{\text{Alice}} \underbrace{|0\rangle}_{\text{Bob}} + \beta \underbrace{|01\rangle}_{\text{Alice}} \underbrace{|1\rangle}_{\text{Bob}}$$

and let her perform a unitary operation on her two qubits that will map:

$$|01\rangle \longrightarrow |01\rangle \quad \text{and} \quad |00\rangle \longrightarrow \frac{\beta}{\alpha}|00\rangle + \sqrt{1 - \left(\frac{\beta}{\alpha}\right)^2}|10\rangle$$

returning the state:

$$|\phi\rangle = \beta|0\rangle(|00\rangle + |11\rangle) + (\alpha^2 - \beta^2)|1\rangle|00\rangle.$$

Now with probability β^2 measuring the ancillary state will give the result “0”. If this result is obtained the other two particles will be in a maximally entangled state whereas obtaining “1” will return the product state $|00\rangle$. Using *LO* (ancilla + local unitary) and *CC* (i.e. communicating the success or failure of the measurement) to increase entanglement was introduced by Bennett et al. [BBPS96] as well as Gisin [Gis96] who described it as “hidden quantum non-locality revealed by local filters”. A single pure state probabilistic distillation of entanglement following that scheme was done in 2001 [KBLSG01]. The last 12 years have of course seen impressive advances beyond the pure, single copy, bipartite and probabilistic distillation that was just presented. Dealing with mixed states, qudits, infinite dimensions or multipartite settings have been some of the issues addressed. A recurring issue concerns the speed at which one can distill these pure entangled states, the resources involved and the trade-off between probabilities and yield (be it maximally entangled states or other target states).

1.2.2 Distillation in Finite Dimensional Hilbert Spaces

Although the first appearance of entanglement [EPR35] was in continuous variables, the Bohm-experiment version of the EPR paradox did get more attention due to its simplicity. Two level quantum systems indeed provide a simple system to work with. Photon polarization, two level atoms or spin- $\frac{1}{2}$ particles are ideal systems for the study of entanglement. The collaboration of information scientists, computer scientists and physicists has also brought a lot of attention towards systems containing ‘qubits’ due to the analogies with the classical ‘bits’. Such systems also suffer the effects of decoherence which hinder the realisation of many quantum information tasks. Distillation of entanglement is therefore a crucial question in these systems.

Bipartite Entanglement Distillation: This example is one of the most frequently studied due to its obvious consequences for quantum information theory. Sharing, encoding or sending a message using ‘qubits’ between two distant parties A and B are basic communication tasks. Furthermore some discoveries about its mathematical structure under

LOCC make it a fertile ground for investigation (for example its relation to majorization [LP01]). Let us then study some of the generalisations beyond the simple example of concentration presented above.

Bennett et al.

The BBPSSW recurrence protocol [BBP⁺96] is a first well defined contribution to distillation aimed at facilitating teleportation. It works for $2 \otimes 2$ states with fidelity $F = \langle \phi^+ | \rho | \phi^+ \rangle > 1/2$. The resource is N copies of a state ρ . Both parties perform $U \otimes U^*$ twirling to get N copies of a $2 \otimes 2$ isotropic state ρ_F and then locally Alice and Bob perform two pairs of XOR operations (also known as CNOT gates):

$$U_{XOR}|x\rangle|y\rangle = |x\rangle|x+y\rangle \quad (1.1)$$

Where the sum in (1.1) is performed modulo 2. The first particle will be called source and the second target. To complete the distillation they take pairs of ρ_F states. Source particles are taken from the first of the two pairs and target particles from the second pair. This leads to many copies of

$$\rho' = U_{XOR_A} \otimes U_{XOR_B} (\rho_F \otimes \rho_F) U_{XOR_A}^\dagger \otimes U_{XOR_B}^\dagger$$

For each of these four qubit states Alice and bob measure target qubits locally in the computational basis. If the results agree they keep the remaining pair of source particles and “twirl” it. Otherwise they discard it. The surviving pairs will have a new fidelity of:

$$F'(F) = \frac{F^2 + \frac{1}{9}(1-F)^2}{F^2 + \frac{2}{3}F(1-F) + \frac{5}{9}(1-F)^2}. \quad (1.2)$$

$F'(F)$ is continuous, $F'(F) > F$ for $F > 1/2$ and $F'(1) = 1$. Therefore one may reach arbitrarily high fidelity if the procedure is iterated.

Efficiency

The problem is that the success probability goes to zero in the limit $F \rightarrow 1$ with the above protocol. In fact it can be shown that this is the case for almost all mixed states [Ken98b]. Nevertheless if F is high enough to ensure that $S < 1$ where S is the Von Neumann entropy, then the hashing protocol [BDSW96] gives asymptotically nonzero distillation rate providing $(1 - S)N$ maximally entangled pairs. Following these same ideas another protocol was presented and applied to Quantum Privacy Amplification.

Quantum Privacy Amplification (QPA)

This method aimed at increasing the security of quantum cryptography over noisy channels (in the entanglement based scheme) appears in [DEJ⁺96]. Based on the ideas from the distillation protocols above it improves the rates and applies it to key distribution over noisy channels.

Requirements

Let us assume that pairs are generated in the state $|\phi^+\rangle$ and then become mixed when distributed over a noisy channel. The basis to describe the state of our pairs will be the Bell state basis $\{|\phi^+\rangle, |\psi^-\rangle, |\psi^+\rangle, |\phi^-\rangle\}$. In it, the density operator will by assumption have diagonal elements $\{a, b, c, d\}$ following the notation in [BEE00]. Therefore, the first diagonal element will be the fidelity: $a = \langle \phi^+ | \rho | \phi^+ \rangle$. The purpose of QPA will be to achieve $a = 1$ and therefore $b = c = d = 0$. The off diagonal elements do not contribute on average to the QPA algorithm so one does not need to specify them. The details of the procedure are similar to the one presented above by Bennett et al. Pairs of states are considered, U_A and U_B rotations are applied on both pairs at each respective side followed by a Controlled-NOT operation on both copies. Afterwards the target pair is measured and coinciding outcomes are kept.

Efficiency

The QPA procedure loses at least one half of the particles (the ones used as targets) at each iteration. In spite of this, it is about 1000 times more efficient than the proposal in [BBP⁺96] when a is close to 1/2. It has also been proved [C.98] analytically that the target point $a = 1, b = 0, c = 0, d = 0$ is a global attractor for $a > \frac{1}{2}$. Nevertheless it does not guarantee the security of the cryptographic protocol because of finite detection efficiencies.

Some general statements

When it is possible for Alice and Bob to transform one or more copies of the resource state ρ into at least one copy of $|\phi^+\rangle$ with high accuracy using LOCC, ρ is loosely said to be distillable. The question “are all entangled states distillable?” had, as was mentioned, a negative answer. Indeed one can see [HHH97] that for example entanglement of $n \otimes m$ positive partial transpose (PPT) states cannot be distilled and they are nevertheless entangled. This leads to the concept of ‘free’ and ‘bound’ entanglement. The former being distillable and the latter not. All currently known examples of bound entangled states have a positive semi definite partial transpose of the density operator. Every PPT state is known to be undistillable. The converse is a central open question.

Another concept arises when trying to optimise this process. The concept of distillable entanglement of a state ρ is intuitively the maximum over all allowable protocols of the expected rate at which “good” EPR pairs can be obtained from a sequence of identical states. A rigorous formulation of it was given by E.M. Rains [Rai99]. To ease the formulation of a rigorous and computable definition it is useful to consider a more powerful set of operations than LOCC, namely operations preserving the positivity of the partial transpose (PPT operations). This set is easier to describe but PPT operations allow more general operations. For instance, to map a product state onto a bound en-

tangled state and ensure the distillability of any NPT (negative partial transposed) state [EVWW01]. Nevertheless It is still an open question if NPT bound entangled states exist with respect to LOCC. The distillable entanglement of a bipartite state ρ under LOCC can be expressed as [APE03]:

Definition 2. *The optimal rate of maximally entangled states that can be distilled from ρ , by LOCC, in the asymptotic limit is:*

$$E_D^{LOCC}(\rho) = \sup_{\{K_n\}} \left\{ \liminf_{n \rightarrow \infty} \frac{\log K_n}{n} \right\}$$

such that

$$\lim_{n \rightarrow \infty} \left(\inf_{\Lambda \in LOCC} \|\Lambda(\rho^{\otimes n}) - \Phi(K_n)\|_1 \right) = 0$$

Where $\Phi(K) = \frac{1}{K} \sum_{i=1}^K \sum_{j=1}^K |i, i\rangle \langle j, j|$ is the maximally entangled state in K dimensions, and the supremum is taken over all possible sequences of integers $\{K_n\}$.

The distillable entanglement for example provides a bound to the optimal rate any protocol of the BBPSSW kind may achieve [BBP⁺96]. However it remains a definition of limited practical applications due to the difficult optimisations it entails.

Now the next obvious question is which states are distillable. If ρ is a pure entangled state, distillation is always possible [BBPS96]. If ρ has a small amount of entanglement, sufficiently many copies of it allow copies of $|\phi^+\rangle$ to be distilled with high accuracy. Furthermore, if ρ is a mixed state of exactly two qubits, if it is entangled it is distillable [BBP⁺96, HHH97]. In a more general fashion a necessary and sufficient condition for a state to be distillable can be expressed as [HH01]:

Proposition 1. *A bipartite state ρ on $\mathcal{H}_{AB} = \mathcal{H}_A \otimes \mathcal{H}_B$ is distillable iff for some two-dimensional projectors P, Q and for some number N , the “two-qubit-like” state*

$$\rho'_N(\rho) = \frac{P \otimes Q \rho^{\otimes N} P \otimes Q}{Tr [P \otimes Q \rho^{\otimes N} P \otimes Q]}$$

is entangled.

In spite of the above proposition there is no **effective** known procedure to determine whether a given state is distillable or not.

A further subtlety was introduced when enquiring precisely how many copies are required for a given distillation [Wat04]. The concept of *n-distillable* state was introduced. A state is said to be *n*-distillable if there exists an LOCC protocol that allows Alice and Bob to convert *n* copies of ρ to a shared pair of qubits that is entangled. Note that *n*-distillability does not require the *n* copies of ρ to become a maximally entangled pair, but only the conversion to an entangled pair. Therefore ρ is distillable iff ρ is *n*-distillable for some *n*. Indeed once we have grouped our states in groups of *n* and distilled entangled pairs, we can use these in a usual protocol like [HHH97] to distill singlets. Actually for pure and mixed states on a single shared pair of qubits distillability and 1-distillability are equivalent. An interesting result from [Wat04] is that

Proposition 2. *For any choice of integers $d \geq 3$ and $n \geq 1$, there exists a $d^2 \otimes d^2$ bipartite mixed quantum state that is distillable but not *n*-distillable.*

This means that entanglement distillation is nonlinear with respect to the number of copies used in the distillation process. There are instances of states ρ where 10^6 copies do not suffice for a single shared pair of non-separable qubits to be created. That distillability is in general not equivalent to *n*-distillability has therefore important consequences.

For a rigorous treatment of distillability and bound entanglement see [Rai99]. For a review containing the first ideas see [HH01, VP98]. For a thorough and rigorous treatment see [DW05].

Multipartite entanglement distillation

Various proposals have been made to distill multipartite entanglement [MPP⁺98, ADB05, DAB03]. This can lead to the distillation of Greenberger-Horne-Zeilinger (GHZ) states

[GHSZ90], that is states of the form:

$$|\phi^\pm\rangle = \frac{1}{\sqrt{2}} (|00\dots 0\rangle + |11\dots 1\rangle).$$

These and other truly entangled multipartite states are a very interesting resource for quantum communication in networks. Indeed communication networks usually involve more than two parties so this will be essential for scalable quantum information processing. Multipartite entanglement is more difficult to quantify in high dimensions but the creation of multipartite entangled particles is already an experimental reality for photons [BPD⁺99, LZG⁺07] molecules [LKZ⁺98], spins in diamonds [NMR⁺08] or ions [HHR⁺05].

Pair assisted distillation

Another multipartite distillation approach involves purifying entangled pairs first and building multipartite entanglement afterwards using the methods from teleportation [BBC⁺93b, ZHWidZ97, BVK98]. Since we know how to purify two particles we can do that first and then entangle them successively. Let us study the case for tripartite distillation.

Description

The procedure [ZHWidZ97, ZHWidZ97] consists of four main steps:

1. Divide the original ensemble in two equal sub-ensembles.
2. Bob and Claire perform projections of particles onto:

$$|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$$

Bob does it with particles from one sub-ensemble and Claire with particles of the other. When they obtain a successful projection onto $|-\rangle$ Alice performs a σ_z on her particles, otherwise she does nothing.

3. A-B on one side and A-C on the other then perform a standard two particle purification process. This results in two maximally entangled ensembles of pairs of particles, shared between Alice and Bob and between Alice and Claire.
4. To obtain a single GHZ state out of two maximally entangled pairs shared between A-B and A-C she chooses one entangled pair from each sub-ensemble. She performs a CNOT operation on her two particles and projects the target particle onto $|0\rangle$ or $|1\rangle$. A successful projection onto $|1\rangle$ is followed by a σ_z operation on Claire's particle, and otherwise nothing is done.

Requirements

Since two particle entanglement distillation requires $f > 1/2$ if we do not know the initial state this will set a limit here too. Otherwise, if we have additional information, a state not fulfilling the above condition could be purified [HHH97]. For more than three particles the criteria are more difficult and we have to turn our attention to schemes that directly distill multipartite entanglement.

Direct distillation

For two particles the singlet state is invariant under any bilateral rotation and this plays an important role in the aforementioned purification schemes. For three or more particles there is not always a known maximally entangled state which is invariant under multi-lateral rotations. This makes it difficult to convert an arbitrary state into a Werner state. In the absence of a maximally entangled state invariant under random bilateral rotations we may introduce a *Werner-type* state [BEE00]:

$$\rho_W = p|\phi^+\rangle\langle\phi^+| + \frac{1-p}{2^N}\mathbb{I} \quad (1.3)$$

This state could describe the attempt to distribute a state $|\phi^+\rangle$ to many parties through a noisy channel. The fidelity of the transmitted state would be evaluated as:

$$f = \langle\phi^+|\rho_W|\phi^+\rangle \quad (1.4)$$

resulting therefore in the expression $f = p + \frac{(1-p)}{2^N}$ for Werner-type states.

A protocol going beyond the pairwise distillation was proposed in 1998 [MPP⁺98] and was called P1+P2. It can purify a Werner-type state of any number of particles, provided the fidelity of the initial mixed state is above a certain threshold.

Description

The protocol consists of Alice and Bob performing each on their side iterations of the operations P1 followed by P2. P1 is a local CNOT and a measurement M1. M1 keeps the control qubits if an even number of target qubits are measured in the state $|1\rangle$. Otherwise the control qubits are discarded.

P2 is a local CNOT operation and a measurement M2 in which the control qubits are kept if all target qubits are found to be in the same state (otherwise they are discarded). Therefore when purifying 3 particles only $|000\rangle$ and $|111\rangle$ are kept.

For instance 4 states can be taken by Bob. P1 is done on one pair and P1 is done on another pair. Two states come out of each P1 operation. Two states are afterwards fed to the P2 operation.

Achievements

This purification is not restricted to Werner-states. Other states can be purified by P1 or P2 alone. For example if the initial state has no weight on $|\phi^-\rangle$ and the other states have

equal weights (or even if other states have zero weight) then P2 alone can purify to $|\phi^+\rangle$

1.2.3 Distillation in Infinite Dimensional Hilbert Spaces

A lot of progress has been made studying qubits or low dimensional systems. Qubits, due to the low dimensionality and symmetries in two dimension offer many opportunities to solve quantum information problems. The maintained efforts in the experimental community to isolate and control two level systems have also contributed to the progress in the area. However both technical and mathematical limitations exist which make it worthwhile exploring beyond qubits or qutrits.

To begin with, the infinite dimensional stage offers new and as yet unexplored possibilities. The rich structure of the infinite Hilbert space makes its analysis more complex but also could unveil new insights, protocols and technologies. For instance one can find highly entangled states very close (in trace norm) to non-entangled ones [ESP02b]. Another characteristic is that very few states are non-distillable [HCL01] or that one can in principle achieve arbitrarily high entanglement. All these features make it an exciting arena. However, one of the main aspects that encourages the use of infinite dimensional systems is the available expertise in quantum optics. The quadrature amplitudes of the quantised electromagnetic field provide the continuous variables, which observe commutation relations analogous to those of position (X) and momentum (P) in the quantum harmonic oscillator.

Most quantum communication protocols require some form of preparation, unitary manipulation and efficient measurement. It turns out that standard optical tools such as non-linear crystals, beam splitters, phase shifters or phase-quadrature measurements fulfill all these requirements. Moreover the breadth of possible continuous variables (CV) implementations makes its exploration very encouraging. Among the physical systems where CV are studied we can count phonons, photons, polarisation of intense beams, cold atoms, Josephson Junction Circuits, Bose-Einstein condensates or nano-

mechanical resonators. In many of these systems standard techniques such as Quantum Key Distribution [GG02, Ral03], teleportation [Vai94, DBL⁺03], quantum erasing [AGL⁺04] or Universal Quantum Computing [SBd02, BL05] have been ported from the discrete variables setting. Decoherence obviously affects continuous variables too [SPID05] and so the distillation concepts must be adapted to this setting.

Bipartite entanglement distillation

In the distant lab paradigm, the case of two parties is the simplest. The infinite dimensional Hilbert space however introduces many difficulties. For instance on a bi-partite infinite-dimensional Hilbert space one can find arbitrarily close states (in trace-norm) whose difference in entropy of entanglement is infinite [ESP02b]. In that sense, many of the definitions from the finite dimensional setup need to be carefully revised or redefined. Since many problems and open questions remain unsolved in finite dimensional spaces it would seem that little can be said about the complex infinite dimensional case. Nevertheless some subsets of states defined in the continuous variable setting prove to be easily described. A notable example are Gaussian states and their manipulation through Gaussian operations. Gaussian states have various advantages since they have a simple mathematical description, are easily generated and standard optical tools can apply Gaussian operations to them.

Gaussian States in CV

Since Gaussian states can be described by a small number of parameters (as opposed to an infinite number of parameters for a general CV state), and due to their importance in linear optics they play a special role in the field of distillation in infinite dimensions. Let us review some of their properties.

We can define some position and momentum operators (for instance representing the position and momentum of a harmonic oscillator or the quadratures of an electromagnetic field) as a linear combination of creation and annihilation operators: $\hat{X} = \sqrt{\frac{1}{2}}(\hat{a} + \hat{a}^\dagger)$,

$\hat{P} = -i\sqrt{\frac{1}{2}}(\hat{a} - \hat{a}^\dagger)$. These operators will then obey the commutation relation $[\hat{X}, \hat{P}] = i$. Let us then define the *Weyl Operator* as:

$$W(X, P) = \exp \left[i(X\hat{P} - P\hat{X}) \right] \quad (1.5)$$

where X, P are real valued variables. For entangled states, and therefore for states with more than one mode, $\hat{\mathbf{R}} = (\hat{X}_1, \hat{P}_1, \dots, \hat{X}_n, \hat{P}_n)$ and the commutation relations can be generalised to:

$$[\hat{R}_j, \hat{R}_k] = i\Sigma_{j,k} \quad (1.6)$$

where Σ is the *symplectic matrix*,

$$\Sigma = \bigoplus_{i=1}^n \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}. \quad (1.7)$$

In this case too the Weyl operator can be generalised to

$$\hat{W}_{\xi} = \exp \left[i \xi^T \Sigma \hat{\mathbf{R}} \right] \quad (1.8)$$

where we have employed the *symplectic product*, $\xi^T \Sigma \hat{\mathbf{R}}$ between the vector of real valued variables ξ and the vector of operators $\hat{\mathbf{R}}$.

We will say that the state ρ is Gaussian when its *characteristic function*

$$\chi_\rho(\xi) = \text{Tr} \left\{ \rho \hat{W}_{\xi} \right\}. \quad (1.9)$$

is Gaussian in the variables ξ [SSM87, AMS97]. This means that the characteristic function $\chi_\rho(\xi)$ can be cast in the form:

$$\chi_\rho = \exp \left[-\frac{1}{4} \xi^T \Sigma \Gamma \Sigma^T \xi + i \mathbf{d}^T \Sigma \xi \right]. \quad (1.10)$$

The $2n \times 2n$ matrix Γ is called the *covariance matrix* and \mathbf{d} is the *displacement vector*. The displacement vector gives the coordinates of the centre of the Gaussian in phase-space, and the *covariance matrix* contains the variances and co-variances. The first and

second moments $\{\mathbf{d}, \Gamma\}$ fully characterize Gaussian states and therein lies the simplicity of their description.

To make things more interesting, many states currently produced in standard quantum optics labs are Gaussian. For instance thermal states, coherent (Glauber) states, or squeezed states [LK87]. More interesting yet is the fact that Gaussian operations (those mapping Gaussian states onto Gaussian states) can be just as easily described [EP03, ADMS95]. These operations can be implemented using phase shifters, beam splitters, squeezers and homodyne detection; again standard tools in the linear optics experimental scene.

What Gaussian states will not do

An optimistic hope before 2002 was that distillation could be done using Gaussian states and Gaussian operations. However it was shown [ESP02a, Fcv02, GIC02] not to be true. More precisely distilling Gaussian states with Gaussian local operations and classical communication (GLOCC) is impossible. It was later shown that the more general set of non-Gaussian operations allows for distillation of Gaussian states [GDCZ01]. A number of protocols that use this alternative have been put forward. All of them require some non-Gaussian element. Either using non-Gaussian states, non-linear interactions or non-Gaussian measurements. I will outline different proposals presenting the ideas and methods involved.

CV Entanglement swapping and entanglement distillation

An early proposal from 1999 [PBP00] noted that purification is always possible if the CV entangled states are projected onto the two levels of the Schmidt basis with the largest Schmidt coefficients. Afterwards one can perform a standard discrete distillation towards two level states with possibly higher entanglement than the original ones. However, an approach producing CV entangled states was also presented beyond the discrete case. A procedure to distill superpositions of coherent states (called cat-states)

was introduced in [PBP00] inspired by entanglement swapping. The key idea was to substitute the Bell state measurement by a reverse entangling operation accompanied by a projective measurement on the two particles. For certain parameters of the initial cat-states distillation was proved to be possible.

These continuous variable macroscopic systems are very interesting systems for distillation. Systems like Bose-Einstein condensates (BEC) or coherent light states are optimal candidates. Further research in the linear optics domain using cat states was developed in 2001 [JK02b]. In this context quasi Bell-states are:

$$|\Phi_{\pm}\rangle_{ab} = N_{\pm} (|\alpha\rangle_a |\alpha\rangle_b \pm |-\alpha\rangle_a |-\alpha\rangle_b) \quad (1.11)$$

$$|\Psi_{\pm}\rangle_{ab} = N_{\pm} (|\alpha\rangle_a |-\alpha\rangle_b \pm |-\alpha\rangle_a |\alpha\rangle_b). \quad (1.12)$$

Here $|\alpha\rangle$ is a one mode coherent state of light and their overlap will decrease with the distance as:

$$\langle \Psi_+ | \Phi_+ \rangle = \frac{1}{\cosh 2|\alpha|^2}.$$

An interesting property is that quasi-Bell states can be unambiguously discriminated using only linear elements like beam splitters and homodyning. The purification aims at purifying states like:

$$\rho_{ab} = F |\Phi_-\rangle \langle \Phi_-| + (1 - F) |\Psi_-\rangle \langle \Psi_-|$$

where the fidelity F is defined as $\langle \Phi_- | \rho_{ab} | \Phi_- \rangle$. In the protocol two copies of ρ_{ab} are taken and modes a, a' are mixed in a beam splitter on Alice's side and b, b' on Bob's side. After the beam splitter, each party performs measurements on the out-coming branches testing if a and a' (b and b') are in the same state by means of a BS and two detectors. In principle a photon parity measurement can reveal which quasi-Bell state was obtained. Practical implementations however suffer a high sensitivity to photon loss given that a single photon lost will change the parity [JK02a]. However, assuming this

is overcome, with certain probability the states are kept and the new fidelity becomes:

$$F' = \frac{F^2}{F^2 + (1 - F)^2}$$

giving an increase as long as $F > 1/2$. The probabilities involved can be quite high and for $|\alpha| \gg 1$ they can be $1/8 \leq P_{\text{succ}} \leq 1/4$. The use of linear optics and the high probabilities make it a promising approach.

Entanglement distillation in continuous variables using non-linearities

To overcome the no-go theorems found in [ESP02a, Fcv02, GIC02] another idea is to make the states interact with non-linear media. The problem is often that the size of the non-linearities reduce the probability of the distillation to impractical levels.

Protocols

Two protocols were published in [FMF03] in 2003. They attempt to distill two-mode squeezed vacuum states of the form:

$$|\psi_q\rangle = \sqrt{1 - q^2} \sum_{n=0}^{\infty} q^n |n, n\rangle.$$

The first scheme is based on the dispersive interaction of a two level atom with the microwave cavity field together with atomic state detection. The second scheme makes use of a cross Kerr interaction, coherent states, homodyne measurements and linear optics.

Description

The mechanism behind these schemes involves an ancillary system. The ancilla experiences a phase shift dependent on the number of photons on one mode of the shared state. This phase modulation is then converted into amplitude modulation via interference which allows to control the amplitude of the Schmidt coefficients. This is a proba-

bilistic method which relies on the result of the measurement on the ancillary state. This result will tell us whether the distillation succeeded or not. Recent developments have refined this techniques but non-linearities are still too small for practical applications [MK06, MK07].

Entanglement distillation in continuous variables by means of linear optics and light measurements.

This idea was introduced in [OKW00] as a method to increase teleportation fidelity through a photon-number measurement. The entanglement increase was however limited by the detector inefficiencies and could not improve much beyond the subtraction. A more general scheme introduced the non-Gaussian character of the procedure in the resource states [BESP03, EBSP04]. This last protocol was iterative and therefore one could increase the entanglement beyond the photon subtraction. It was then the first feasible protocol for distillation of entanglement in continuous variables that used exclusively linear optics and non-number resolving photo-detection.

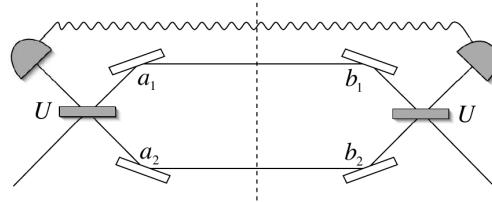


Figure 1.2: borrowed from [BESP03], Diagram showing one iteration of the linear optics distillation protocol for continuous variables. The tensor product $\rho \otimes \rho$ distinguishes the upper and lower branch, and the product $|n\rangle \odot |m\rangle$ the left and right modes.

One step of the iterative procedure from the [BESP03, EBSP04] distillation protocol is depicted above. Two copies $\rho \otimes \rho$ (upper and lower copies) of two mode states are mixed at the beam splitters (BSs). This is followed by an avalanche photo-diode (APD) measurement on the two upper modes. A successful step of the protocol occurs when zero photons are detected, and the out-coming modes are then kept. One such step of the protocol can be described as taking $\rho^{(i)}$ to $\rho^{(i+1)}$ by means of:

$$\rho^{(i+1)} = \langle 0|_{c_1} \langle 0|_{d_1} (U \odot U)(\rho^{(i)} \otimes \rho^{(i)})(U \odot U)^\dagger |0\rangle_{c_1} |0\rangle_{d_1}$$

where the unitary operations U describe the action of the BS. Another version of it [EPB⁺07] uses the more efficient Homodyne Detection instead of vacuum projections. This setup can be equivalent but is more complex to analyse in its full generality as we will see in section 2.6. It has been shown that each starting state $\rho^{(0)}$ needs to be non-Gaussian if $\rho^{(N)}$ is to have an entanglement greater than the one of the original state. These non-Gaussian resource states $\rho^{(0)}$ can be obtained for example using APD detectors and linear optics in a photon subtraction scheme.

It has been shown in [EBSP04] that non-Gaussian $\rho^{(0)}$'s lead to enhanced entanglement and purity after an arbitrary number of iterations of the protocol. Note that that mixed input states can converge to pure Gaussian ones. This has been shown to be possible for mixed and pure states and for current APD detection efficiencies. Furthermore the family of states leading to this increase has been characterised. These states converge towards a Gaussian state after a few iterations.

Therefore, once the necessary non-Gaussian states are obtained the distillation procedure mixes them to obtain a single state with higher purity and entanglement. The lowest probabilities involved in the problem are found in the ‘Degaussification’, or preparation of the resource states [BESP03]. These low probabilities make it hard to scale the procedure to distill many copies. Additionally in that protocol the resources scale exponentially with the number of steps of distillation. I will present and study these problems in chapter 2.

Multipartite entanglement distillation

This idea is promising since the CV states can be easy to manipulate with linear optics and easy to distribute. Monogamy of entanglement in this case introduces different limitations since maximally entangled states are not limited to have 1 unit of entangle-

ment but can achieve arbitrarily high values both in the bipartite and multipartite case [AEI07, AI07].

Distillation in a CV multipartite network

This idea has barely been developed. Nevertheless a distillation in a CV quantum teleportation network has been proposed [vB00]. The basic idea is that a single mode squeezed state is sufficient to allow quantum teleportation between any two of N parties with the help of all other parties. The assistance of the other N-2 parties relies only on LOCC. Because of these N-2 measurements, bipartite entangled states are distilled from the initial N-partite entanglement.

1.2.4 Alternatives

We could also ask if there are alternatives to quantum entanglement distillation that overcome the difficulty of not having maximally entangled states. Indeed, in continuous variables, it has been shown that it is possible to distill a secure secret key with Gaussian operations on Gaussian states. For example [IVAC04] shows that the transmission of Gaussian-modulated coherent states and homodyne detection is equivalent to an entanglement purification protocol using CSS error correcting codes [CS96, Ste96] followed by key extraction. Also in [NBC⁺04] it is shown that it is possible to distill a secure key (under certain assumptions) from sufficiently entangled Gaussian states with non-positive partial transposition. This process does not require distillation and makes use of Gaussian states and Gaussian operations alone. Other novel ideas include using the effect of the environment to enhance entanglement [GMN06], or converting stabilizer codes to distillation protocols [Mat02].

As we have seen, many new ideas are being developed with the Gaussian state CV formalism trying to improve probabilities, use of resources and scalability of the protocols. As it stands it is a challenging area full of theoretical and experimental open questions.

1.3 Practical Quantum Optics CV Distillation

Our focus will be on implementations of CV distillation that try to use linear optics and photo-detection as the main resources. Part of the reason is that these resources are readily available today. Photons with CV entanglement were generated 20 years ago using squeezed states [WXK87]. However, distillation from these sources remains up to this day an experimental challenge. Some of the reasons involve not having pure enough states or not having them at sufficiently high rates. Other reasons have to do with low probabilities or with the complexity of the setups. On a different arena, revealing entanglement increase requires one to measure the entanglement before and after the distillation procedure. Finding simple and practical ways to determine this CV entanglement rigorously has been part of the problem too. I will explain how to use some mathematical tools to evaluate this entanglement precisely in chapter 3.

1.3.1 Photon Subtraction and Procrustean Distillation

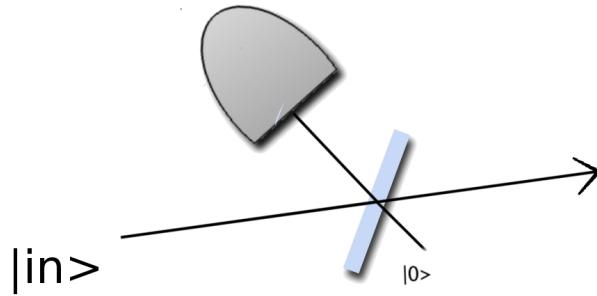


Figure 1.3: Diagram of a photon subtraction setup. A quantum state goes through a beam splitter (meeting the vacuum at the other port). A successful subtraction occurs when a photon is measured in one of the ports. This non-Gaussian operation can create non-Gaussian states.

As was discussed earlier, distillation with the standard linear optics (and thus with Gaussian operations) requires non-Gaussian states. Generating non-classical non-Gaussian states is an active and challenging field in quantum optics with promising applications

to quantum information. One simple way to generate non-Gaussian states involves subtracting a photon from a Gaussian state as shown in Fig C.1. Great progress has been made generating cat states, kittens and photon subtracted states [OP05b, NNNH⁺06, AOBG07, OTBLG06, ODTBG07]. Other non-Gaussian states are NooN states (of the form $|\phi\rangle \sim |N0\rangle + |0N\rangle$) [SOG06, EHKB04, MLS04, WPA⁺03], close approximations to Fock states [BAS⁺06, LO05, qGlFzX03, ROW⁺07, OTBG06], or photon-added states [AZ04].

In spite of all this progress one crucial proof of principle experiment remains elusive: demonstrating an increase of CV entanglement solely with LOCC. My calculations in chapter 3 elucidate the challenges and opportunities in this area.

Another crucial aspect that requires careful examination is the adequate characterisation of entanglement. Experimental data can reveal strong correlations in quantum states and not imply any entanglement [AP06, GRW07]. The purity of the states and the relationships between the measured observables must be studied carefully in these cases. Of course, a full tomographic reconstruction of the density matrix can unveil the available entanglement. However, even avoiding the arduous task of complete tomography, one can construct rigorous statements about the entanglement present from partial measurement on the state. The use of quantitative entanglement witnesses [EBA07, AP06, GRW08] is a tool very well suited for such situations and I will show how it is applied to specific examples.

1.3.2 CV Experimental and Theoretical Tools in Quantum Optics

The complexity in the recent quantum optics experiments generating non-Gaussian states is all too often ignored or approximated. However studying some of its elements in greater detail can lead to improved performance and understanding. The full description of the down-converted states in all their degrees of freedom (frequency, polarization, photon-number, angular-momentum, etc) is one example. Another exam-

ple relates to the detectors and processes used to manipulate, measure and prepare the states. Photo-detection has been acquiring increasing complexity. In the last decade apparatuses such as single-carbon-nanotube detectors [FMM⁺03], charge integration photon detectors (CIPD) [SWM⁺04], Visible Light Photon Counters (VLPC) [KTYH99], quantum dot arrays [SOF⁺00], superconducting edge or picosecond sensors[MNMS03, GOC⁺01] or time multiplexing detectors [ASS⁺03a, ASS⁺03b] have made their appearance. Understanding in full detail the physical processes that occur in such detectors is of course out of our reach and we must resort to partial calibrations. Yet we rely on these detectors for state preparation and state tomography. I will present the first results attempting a more rigorous characterisation of detectors through detector tomography [LSS99, LFCR⁺08].

2

Continuous Variable Distillation of Entanglement with Linear Optics

This second chapter builds upon the ideas presented by Browne et al. between 2003 and 2005 [BESP03, EBSP04, Bro05]. It analyses the inherent limitations of those optical CV distillation protocols and introduces modifications geared towards making the protocols less resource consuming and more efficient.

2.1 Motivation

The practical implementations of entanglement distillation in discrete finite Hilbert spaces often run into crucial limitations. Proof-of-principle experiments have been performed in optics [KBLSG01, ZYC⁺03, YKOI03] but they require single photons or photon number resolving counters which are either difficult to produce or expensive. Furthermore current experiments require the destruction of the state in order to prove the distillation was successful (making any iteration impossible). Proposals without post-selection exist (for 2-dimensional systems) [XbH03, HK07] but are yet to be implemented. For more details concerning the problems detector efficiency, mode matching, post-selection or bandwidth impose on distillation see, for example, Rohde et. al,

[RRM06].

It is in this context that we turn our attention to entanglement distillation in continuous degrees of freedom by means of linear optics (Gaussian operations). This proposal has seen an increasing experimental interest [EPB⁺07, FMF03] due to major experimental improvements in both linear optics and the detection of light. After presenting the scheme from [BESP03, EBSP04] I will identify the problems involved in a realistic implementation. I will evaluate the success probabilities for distillation and the resources needed. In an attempt to improve it, I will introduce my variations on the protocol to obtain a deeper insight. This will raise different questions answered in further chapters.

2.2 The $(\rho^{(i)}, \rho^{(i)})$ Protocol

We will call the protocol from Browne et al. [BESP03] the $(\rho^{(i)}, \rho^{(i)})$ protocol. The motivation behind this notation will be clarified later. We will also refer to it as the ‘Gaussifier’ or ‘Gaussification protocol’ since states converge towards Gaussian states when fed into the protocol.

2.2.1 Brief Description

To picture the whole distillation process we can first imagine that the two parties, say Alice and Bob, have 2^N -copies of a bipartite state. They also have lots of beam-splitters (BS) and avalanche photo-diode (APD) detectors. Now they group those bipartite states in pairs, say $\rho_{AB} \otimes \rho_{AB}$. They will use BS and APD-s to operate on each of the 2^{N-1} pairs. If the procedure is successful (and classical communication will inform Alice and Bob of it) each pair will generate a single bipartite state $\rho_{AB} \otimes \rho_{AB} \rightarrow \sigma_{AB}$. The 2^{N-1} states left will be grouped in 2^{N-2} pairs. Each pair will be acted on with the same configuration of BS, APD and classical communication (CC) as before. If they are all successful, we will now have 2^{N-3} pairs and so on until we have a single state left.

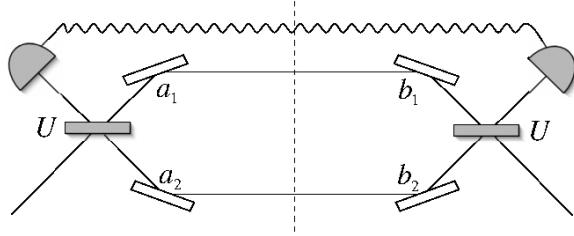


Figure 2.1: borrowed from [BESP03], The tensor product $\rho \otimes \rho$ distinguishes the upper and lower branch, and the product $|n\rangle \odot |m\rangle$ the left and right modes. The measured modes will be a_2 and b_2 , and the ones transmitted to the next stage will be a_1 and b_1 .

One step of the iterative procedure from the distillation protocol is depicted in Fig 2.1. Two copies $\rho \otimes \rho$ (upper and lower copies) of two mode states (modes a and b) are mixed at a pair of beam splitters (BSs). This is followed by an avalanche photo-diode (APD) measurement on the two upper modes. A successful step of the protocol occurs when zero photons are detected (no click event). In that case the out-coming modes are kept. One such step of the protocol can be described as taking two copies of $\rho^{(i)}$ to a single copy of $\rho^{(i+1)}$ by means of:

$$\rho^{(i+1)} = \langle 0|_{a_2} \langle 0|_{b_2} (U \odot U)(\rho^{(i)} \otimes \rho^{(i)})(U \odot U)^\dagger |0\rangle_{a_2} |0\rangle_{b_2} \quad (2.1)$$

where the unitary operations U describe the action of the BS. Note that the symbol \odot indicates a tensor product between left and right, whereas the symbol \otimes is a tensor product between the upper and lower branches of Fig 2.1.

The procedure can be iterated as shown in Fig. 2.2. That way, after successfully obtaining two copies of $\rho^{(i+1)}$, we can map them in turn to a copy of $\rho^{(i+2)}$. The way states are combined in the $(\rho^{(i)}, \rho^{(i)})$ protocol can be schematically represented by figure 2.3. Note that if one of the steps is unsuccessful, the procedure has to be started from $\rho^{(0)}$ again.

One of the results from the protocol is that Gaussian states are a fixed point of the map in Eq. 2.1. In other words, and as expected from the use of GLOCC, the pro-

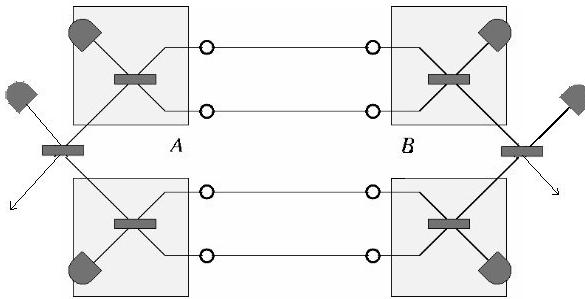


Figure 2.2: This diagram shows two steps of the Gaussification (or distillation) procedure from [BESP03]. After the first step occurs in the gray boxes, the successful states are mixed in the outer beam-splitters, detected, and the remaining modes become the output (arrows). We can say that four copies of $\rho_{AB}^{(0)}$ are mapped onto two copies of $\rho_{AB}^{(1)}$ which are in turn mapped to a single $\rho_{AB}^{(2)}$. This of course requires classical feed forward.

toocol will not increase the entanglement of Gaussian states. The key result is that the use of certain non-Gaussian states as resource states ($\rho^{(0)}$) achieves arbitrarily high entanglement and purity in the output state ($\rho^{(N)}$) for sufficiently large N . In fact for sufficiently high N the output state converges towards a Gaussian state with arbitrarily high entanglement. Furthermore, mixed states have also been shown to become more pure throughout their Gaussification. To carry out an implementation of the protocol we must then consider the “de-gaussification” or generation of non-Gaussian states as a preparation step. The original proposal introduces a method to generate non-Gaussian states with linear optics and APDs. The setup from Fig. 2.4 illustrates this procedure. Two squeezed states are used to generate a state that approximates $|\phi\rangle = |00\rangle + \mu|11\rangle$ well. Feeding $|\phi\rangle$ to the gaussification protocol will make $\rho^{(N)}$ converge towards $\sum_n \mu^n |n, n\rangle$ for large N . Therefore achieving $\mu \simeq 1$ is very advantageous. To do so, two squeezed states can be combined as shown in Fig. 2.4. This involves the coincident detection of two *click* events on the modes a_1 and b_2 and is therefore a probabilistic preparation event.

Both the use of non-Gaussian states and the way they are combined (see fig 2.3) hint at some of the problems of the protocol. The number of starting states increases expo-

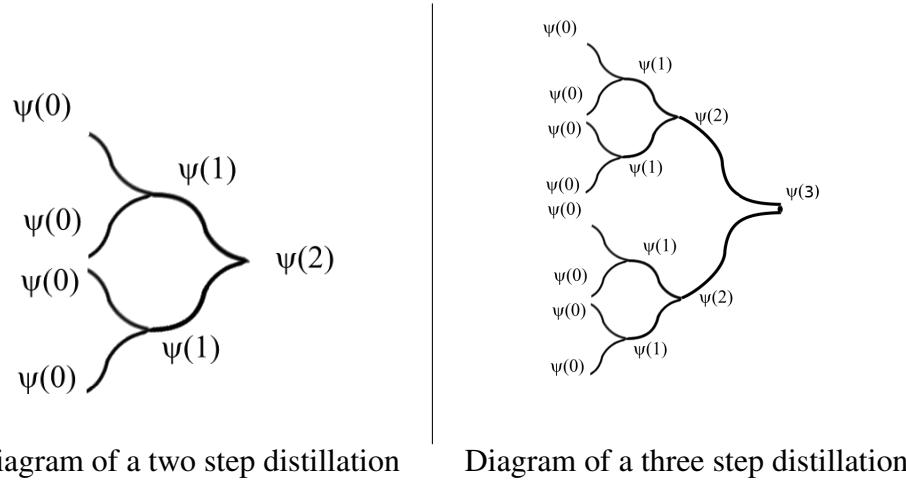


Figure 2.3: : These diagrams show how the resources for the $(\rho^{(i)}, \rho^{(i)})$ protocol scale exponentially.

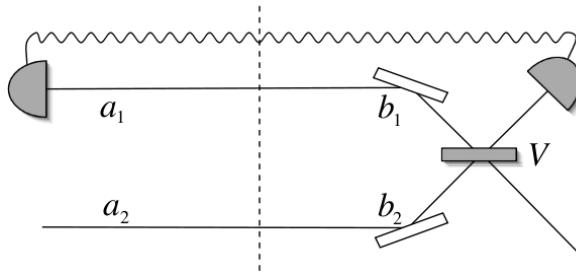


Figure 2.4: borrowed from [BESP03], setup for the preparation of non-Gaussian states in a probabilistic manner. $|Sq\rangle_{AB} \otimes |Sq\rangle_{AB}$ are our starting states which are two squeezed vacuum states. The successful creation of the non-Gaussian state involves two “clicks” at the detectors.

nentially with the number of iterations. Additionally, unless we have some switching or storing for the photons, the number of BS and APDs also seems to do so. And finally, non-Gaussian states are usually hard to generate or are generated with low probabilities. Let us then look at the question of resources in more detail.

2.2.2 Resources

To evaluate the resources needed we will have to make some assumptions about the setup and specifically about how one step of the protocol feeds into the next one. Let us for now consider three simple scenarios:

- Coincidence Scenario: A successful distillation occurs when all the resource non-Gaussian states $\rho^{(0)}$ are probabilistically obtained simultaneously, the subsequent $\rho^{(1)}$ are also obtained simultaneously in the next step, and so on until $\rho^{(N)}$ is reached. Therefore many coincidences are needed.

In this scenario, if one wants to implement N steps of the protocol 2^N copies of $\rho^{(0)}$ are required. If these are generated in the described photon subtraction scheme, 2×2^N squeezed states are needed. If each BS and detector is used for one mode only, then one needs $2^{N+1} - 2$ detectors and BSs for the gaussification and 2×2^N detectors plus 2^N BSs for the preparation step. For instance 3 steps of the protocol require 16 squeezed states, 24 BSs and 30 detectors.

- Storage Scenario: Here, the first $\rho^{(0)}$ state obtained can be stored until another one has been produced to begin the distillation. They are then combined pairwise and each intermediate $\rho^{(i)}$ can be stored until its homologue $\rho^{(i)}$ is obtained. Afterwards, once $\rho^{(i)} \otimes \rho^{(i)}$ is available they are combined, the result stored and so on. This protocol produces both copies $\rho^{(i)} \otimes \rho^{(i)}$ with the same BSs and APDs. When both are obtained they are then mixed in a new BS and detected with new detectors.

One can make a chain needing $(2N + 4)$ APD detectors and $(2N + 2)$ BS to carry out N steps. Therefore, 3 steps require only 4 squeezed states, 8 BS and 10 detectors. The resource consumption is therefore reduced from exponential to linear. In fact we can go even further as shown in Fig. 2.5 . We can go from $\mathcal{O}(2^N)$ to 4 APDs and 4 BS if we use a clever combination of time delays, classical communication and storage. Again once a successful state has been created it is stored while the matching pair is created. When it is, the stored one is put back into the circuit. Nevertheless the optical switches, the storage in loops and the subsequent mode matching both in time and space have problems of their own. The loss from commercially available switches introduces decoherence problems which may be

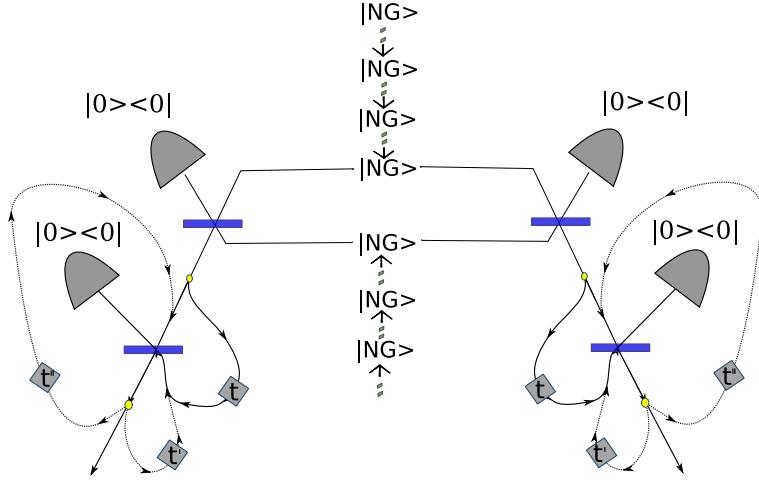


Figure 2.5: Distillation setup with storage. The non-Gaussian states $|NG\rangle$ are fed at set time intervals to the first stage of the protocol. Successful states are stored in fibres (by means of optical switches (yellow dots) and time delays (grey boxes)). Note that classical communication must coordinate 4 detectors + 4 switches + 6 time delays.

harder to overcome than the low probabilities ¹.

- The Random Walk Scenario: The aim of this setup is to reduce the number of discarded states: If each run of the protocol requires $\mathcal{O}(2^N)$ states and it is highly unlikely that the distillation will succeed, we are throwing away a large number of entangled states each time. The basic idea is to keep the states whatever the result and follow the evolution of their entanglement. Specifically, the original protocol requires all detection events to be a “no click” event. Yet there are many combinations of success and failure chains that can lead to increased entanglement without discarding immediately any “click” event. Let us consider this notation: The usual protocol describes the transition from one iteration to the next one as: $\rho^{(i)} \otimes \rho^{(i)} \rightarrow \rho^{(i+1)}$. Implicit in the description is that the transformation in Eq. 2.1 maps one stage to the next one. We could instead define a more general map:

$$\rho_{o1}^{(i+1)} = \langle 1|_{a2} \langle 0|_{b2} (U \odot U) (\rho_{o1}^{(i)} \otimes \rho_{o2}^{(i)}) (U \odot U)^\dagger |1\rangle_{a2} |0\rangle_{b2} \quad (2.2)$$

¹Some standard multiport switches in telecom (Polartis for example) operate in the millisecond regime with loss < 0.3dB (7%), but the faster ones (≈ 10 nsec) usually exhibit more loss $\approx 5 - 15$ dB (68% - 96%) (IEICE Electronics Express, Vol.5, No.6, 181). One could maybe use a slow one discarding all the events outside the switching window.

where $\vec{o1}$ is a list carrying the information about all the previous outcomes. For example, $\rho_{\vec{o1}}^{(2)}$ might be the state created by the events $\vec{o1} = \{1, 0, 0, 0; 0, 1\}$ and therefore meaning

- $\rho^{(0)} \otimes \rho^{(0)} \rightarrow \rho_{\{1,0\}}^{(1)}$ (one click on the left arm)
- $\rho^{(0)} \otimes \rho^{(0)} \rightarrow \rho_{\{0,0\}}^{(1)}$ (no clicks)
- $\rho_{\{1,0\}}^{(1)} \otimes \rho_{\{0,0\}}^{(1)} \rightarrow \rho_{\{1,0,0,0;0,1\}}^{(2)}$ (one click on the right arm)

Depending on the original $\rho^{(0)}$ state, the output state could have more entanglement than the starting one. Tapping into this resource of otherwise discarded states can also improve the use of resources. The measurements can of course be generalized to POVMs and account for lossy APDs and mixed states.

The main conclusion is that the number of states and the number of optical elements from the original proposal need to be reduced. Reducing it too much requires complex or deficient technology, so reaching a compromise that improves feasibility is our goal. The next sections will explore different modifications of the original protocol and the improvements achieved. However, before we do so we must address another limitation in more detail.

The Entangled States Resource

In the long run, the distillation is aimed at distilling CV states that have suffered decoherence in an optical channel. Mixing can have its source in various physical processes: depolarisation, phase mixing, photon loss, Gaussian noise, etc. Due to the complexity of describing states that would occur in practical quantum communication I will adopt two approaches. The behaviour of pure states in general distillation schemes will be studied. When possible, conclusions will be generalised to the case of absorbing fibres (Gaussian Channel) or to arbitrarily mixed states. Also, further chapters will detail the effect of optical loss and imperfect measurement on coherence in simple examples.

In the pure case, the most common entangled states will be two mode squeezed vacuum states due to their availability. Typically they are described in the Fock basis by:

$$|Sq\rangle = \text{sech}(r) \sum_{n=0}^{\infty} [-e^{i\phi} \tanh(r)]^n |n\rangle |n\rangle = \sqrt{1 - \lambda^2} \sum_{n=0}^{\infty} \lambda^n |n, n\rangle \quad (2.3)$$

Where r is generally called the degree of squeezing². Generating two mode squeezed states with high average photon number $\langle n \rangle = \frac{2\lambda^2}{1-\lambda^2}$ is limited by the gain in the parametric amplifier and so λ is usually small. The highest degree of squeezing achieved with a continuous wave laser stands at -9dB [TYYF07] ($r \approx 1$). For pulsed light on the other hand the highest squeezing achieved is -4.6 dB [HFT⁺05] ($r \approx 0.52$ and $\lambda \approx 0.48$). These are however difficult experiments and it is more common to find for $\chi^{(2)}$ pulsed laser Parametric Down Conversion $\lambda \in [0.01, 0.2]$. This values will eventually set limits to the rates at which states can be distilled as we will see in the following section.

2.3 Probabilities

The lowest probabilities involved in the problem are found in the ‘Procrustean³ preparation’ introduced in [BESP03]. We will see that the probabilities for successful distillation are quite low as soon as we attempt to do many distillation steps.

2.3.1 Probability for the Procrustean Preparation

To gain some insight we will approximate squeezed states neglecting any term of order $\mathcal{O}(\lambda^3)$ or higher: $|\phi_{sq}\rangle \approx \frac{1}{\sqrt{1+\lambda^2+\lambda^4}} (|00\rangle + \lambda|11\rangle + \lambda^2|22\rangle)$ Two bipartite states $|\phi_{sq}\rangle \otimes |\phi_{sq}\rangle$ start in the upper and lower branches pictured in Fig. 2.6. The left mode of the

²Squeezing reduces the uncertainty in a quadrature of the electromagnetic field. This uncertainty is often described in the experimental literature as *noise* and measured in decibels. The noise d and squeezing parameter r are related by $r = (\ln(10)/20)d \approx 0.115d$ [Bro05].

³The term ‘Procrustean’ alludes to Procrustes, “a mythical Greek giant who stretched or shortened captives to make them fit his beds”. In the distillation context it refers to a method which using LOCC and with certain probability aims at ‘cutting off’ the weight of certain coefficients of the state expanded in the Schmidt basis. In this case it refers to the probabilistic method of generating non-Gaussian states while increasing entanglement.

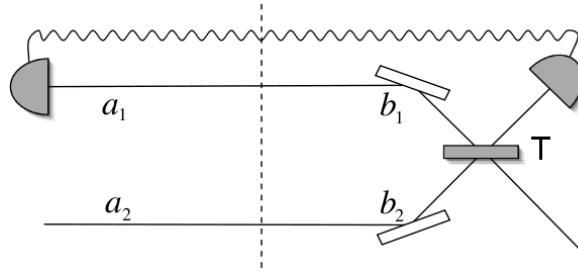


Figure 2.6: borrowed from [BESP03], Procrustean generation of the state $|\phi^{(0)}\rangle \sim |00\rangle + \mu|11\rangle$ from two squeezed states $|Sq\rangle_{a1,b1} \otimes |Sq\rangle_{a2,b2}$.

upper branch is measured and the probability of a click in that detector is simply

$$P(\text{click}_1) = \frac{\lambda^2 + \lambda^4}{1 + \lambda^2 + \lambda^4} \approx \frac{\lambda^2}{1 + \lambda^2}. \quad (2.4)$$

Disregarding the two photon event's probability ($\mathcal{O}(\lambda^4)$), the state of the system after the measurement is approximately:

$|\phi'\rangle \approx \frac{1}{\sqrt{1+\lambda^2+\lambda^4}} (|100\rangle + \lambda|111\rangle + \lambda^2|122\rangle)$. Applying the beam splitter transformation,

$$\begin{aligned} \hat{b}_1^\dagger &= t\hat{b}_1^\dagger + r\hat{b}_2^\dagger \\ \hat{b}_2^\dagger &= t^*\hat{b}_2^\dagger - r^*\hat{b}_1^\dagger \end{aligned}$$

and assuming $t, r \in \mathbb{R}$ it becomes

$$\begin{aligned} \hat{U}|\phi'\rangle \sim & r|0, 1, 0\rangle + \lambda(t^2 - r^2)|1, 1, 1\rangle + t|1, 0, 0\rangle - \sqrt{2}\lambda tr|2, 0, 1\rangle + \\ & \sqrt{2}\lambda tr|0, 2, 1\rangle + \sqrt{3}\lambda^2 tr^2|3, 0, 2\rangle + \lambda^2(-2t^2r + r^3)|2, 1, 2\rangle + \\ & (-2\lambda^2 tr^2 + \lambda^2 t^3)|1, 2, 2\rangle + \sqrt{3}\lambda^2 t^2 r|0, 3, 2\rangle \end{aligned}$$

And after detection of the second photon, we can write the density matrix as follows:

$$\rho^{(0)} \sim \text{Tr}_2 \left\{ (\mathbb{I} - |0\rangle\langle 0|) \hat{U} |\phi'\rangle\langle\phi'| \hat{U}^\dagger \right\}$$

	$\langle 0, 0 $	$\langle 0, 1 $	$\langle 1, 1 $	\dots
$ 0, 0\rangle$	r^2	0	$\lambda (-r^2 + t^2) r$	
$ 0, 1\rangle$	0	$2\lambda^2 t^2 r^2$	0	
$ 1, 1\rangle$	$\lambda (-r^2 + t^2) r$	0	$\lambda^2 (-r^2 + t^2)^2$	
\vdots				\ddots

$$\rho^{(0)} \approx \begin{array}{c|cccc} & \langle 0, 0 | & \langle 0, 1 | & \langle 1, 1 | & \dots \\ \hline |0, 0\rangle & r^2 & 0 & \lambda (-r^2 + t^2) r & \\ |0, 1\rangle & 0 & 2\lambda^2 t^2 r^2 & 0 & \\ |1, 1\rangle & \lambda (-r^2 + t^2) r & 0 & \lambda^2 (-r^2 + t^2)^2 & \\ \vdots & & & & \end{array} \quad (2.5)$$

The second event (or *click*) occurs with probability,

$$P(\text{click}_2) = -\frac{(-3r^4 - 2 + 3r^2)\lambda^4 + (-1 - r^4)\lambda^2 - r^2}{1 + 2\lambda^2 + 2\lambda^4} \quad (2.6)$$

if we disregard terms of order λ^6 (note that this is consistent with the limit $R \rightarrow 1$ where the photon from the upper branch $|1\rangle_{A2}$ is reflected by the BS and into the detector. In that case we recover a probability of 1 independent of λ).

One should remember that creating a non-Gaussian state of the form: $|00\rangle + \mu|11\rangle$, with $\lambda \ll \mu$ is very advantageous as the distillation can drive it to a highly entangled state. To achieve this, we need to choose a BS with r such that

$$|\phi^{(0)}\rangle \sim r|00\rangle + \lambda rt\sqrt{2}|01\rangle + \lambda(t^2 - r^2)|11\rangle \simeq |00\rangle + \mu|11\rangle$$

And such that all other terms in the rest of the density matrix $\rho^{(0)}$ are sufficiently small.

To achieve this we require $r = \lambda(t^2 - r^2)$ which leads to:

$$r = \left| \frac{-\mu + \sqrt{\mu^2 + 8\lambda^2}}{4\lambda} \right| = \left| \frac{2\lambda}{-\mu - \sqrt{\mu^2 + 8\lambda^2}} \right|$$

which for small λ gives an approximate $r \sim \left| \frac{\lambda}{\mu} \right|$.

We can see if the approximation holds for the case where $\rho_{0,0,0,0}^{(0)} \simeq \rho_{1,1,1,1}^{(0)}$. Indeed taking $r,t \subset \mathbb{R}$ and considering as above $r = \left| \frac{1-\sqrt{1+8\lambda^2}}{4\lambda} \right|$ we can expand the ratios of the other density matrix elements for small values of λ :

$\rho_{0,1,0,1}^{(0)}/\rho_{0,0,0,0}^{(0)} = \rho_{2,2,0,0}^{(0)}/\rho_{0,0,0,0}^{(0)} = \rho_{0,2,0,2}^{(0)}/\rho_{0,0,0,0}^{(0)} = \rho_{1,2,1,2}^{(0)}/\rho_{0,0,0,0}^{(0)} = \rho_{2,2,1,1}^{(0)}/\rho_{0,0,0,0}^{(0)} = O(\lambda^2)$
 $\rho_{1,2,1,2}^{(0)}/\rho_{0,0,0,0}^{(0)} = \rho_{2,2,2,2}^{(0)}/\rho_{0,0,0,0}^{(0)} = O(\lambda^4)$. It is then apparent that the remaining coefficients will be small for the chosen beam-splitter and a small λ . We can then substitute the reflectivity r back in probability expression (2.6) to find,

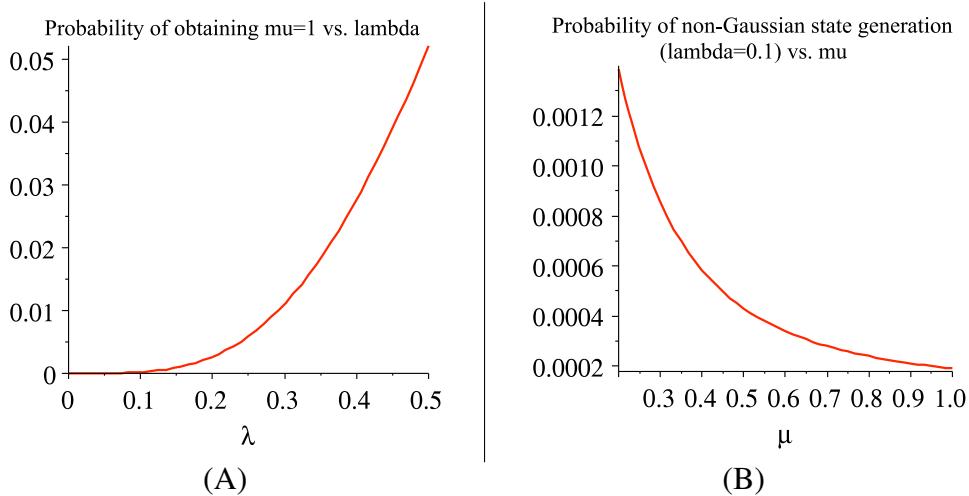


Figure 2.7: Overall probability of creating a non-Gaussian state of the form $|00\rangle + \mu|11\rangle$. (A) shows the probability of obtaining $\mu = 1$ as a function of the squeezing λ and (B) shows the probability for different μ starting with two approximate squeezed states with $\lambda = 0.1$ squeezing.

$$P(\text{click}_2) \approx 2 \frac{\lambda^2 \left(2 + \mu^2 + \mu \sqrt{\mu^2 + 8\lambda^2} \right)}{(1 + \lambda^2 + \lambda^4) \left(\mu + \sqrt{\mu^2 + 8\lambda^2} \right)^2}$$

Fig. 2.7 shows the total probability with the only approximation being our original squeezed state. If we assume a usual low intensity pulsed PDC source with 1dB squeezing or $\lambda \approx 0.1$ then $\mu = 1$ is obtained with probability $2 \cdot 10^{-4}$. Obviously if we need to create independently 2^N such states to run N stages of the distillation, probabilities decay very rapidly. Fortunately increasing to 1.8dB ($\lambda \approx 0.2$) increases the probability by an order of magnitude. We must however explore how these probabilities integrate in the whole distillation process.

2.3.2 Probability for the Gaussification

We can see in [BESP03] that the protocol will transform two copies of two-mode non Gaussian states in the following way:

$$|\psi^{(i+1)}\rangle = \langle 00|(U_{12} \otimes U_{12})|\psi^{(i)}\rangle|\psi^{(i)}\rangle \quad (2.7)$$

Therefore the probability of vacuum outcomes being detected in both modes, at the $i + 1$ -th iteration, or the probability of obtaining $|\psi^{(i+1)}\rangle$ given two copies of $|\psi^{(i)}\rangle$ is:

$$P(\psi^{(i+1)}) = \frac{\langle\psi^{(i+1)}|\psi^{(i+1)}\rangle}{|\langle\psi^{(i)}|\psi^{(i)}\rangle|^2}$$

Let us consider the simple example of pure states expressed in the Schmidt basis as $|\psi^{(i)}\rangle = \sum_n \alpha_n^{(i)} |n, n\rangle$. The relation in 2.7 leads to a recurrence relation [BESP03] which results in a probability:

$$P(\psi^{(i+1)}) = \frac{\sum_{n=0}^{\infty} 2^{-2n} \sum_{r,l=0}^n \binom{n}{r} \binom{n}{l} \alpha_r^{(i)} \alpha_{n-r}^{(i)} \overline{\alpha_l^{(i)} \alpha_{n-l}^{(i)}}}{\sum_{n,m=0}^{\infty} |\alpha_n^{(i)}|^2 |\alpha_m^{(i)}|^2}. \quad (2.8)$$

For example if $|\psi^{(0)}\rangle \sim |00\rangle + \mu|11\rangle$, so that $\alpha_0^{(0)} = 1$ and $\alpha_1^{(0)} = \mu$ then

$$P(\psi^{(1)}) = \frac{1 + \mu^2 + \frac{1}{4}\mu^4}{(1 + \mu^2)^2}$$

And so on. These probabilities are quite high. For instance if starting with $\lambda = 0.1$ we obtain with probability $2.3 \cdot 10^{-4}$ a non-Gaussian state with $\mu = 0.8$ then,

- $P(\psi^{(1)}) = 0.64,$
- $P(\psi^{(2)}) = 0.62,$
- $P(\psi^{(3)}) = 0.57,$
- ⋮
- $P(\psi^{(12)}) = 0.36.$

Nevertheless the success probability of the overall process is much lower.

2.3.3 Overall Success Probability and Outlook

The way states are combined in the $(\rho^{(i)}, \rho^{(i)})$ protocol is represented in Fig. 2.3 and explained in section 2.2.2. We can then express the overall probability of obtaining the state $|\psi^{(n)}\rangle$ as

$$P_O(\psi^{(n)}) = \prod_{r=1}^{n-1} [P(\psi^{(n-r)})]^{2^r}$$

which obviously decreases exponentially. For example to reach the third iteration the probability can be expressed as: $P_O(\psi^{(3)}) = P(\psi^{(0)})^{2^3} P(\psi^{(1)})^{2^2} P(\psi^{(2)})^{2^1} P(\psi^{(3)})$ where $P(\psi^{(0)})$ is given by (2.6 and 2.4) and the rest by (2.8). Considering the above more or less realistic values, $\lambda = 0.1$, $\mu = 0.8$, we obtain the following overall probabilities:

- $P_O(\psi^{(1)}) \approx 3 \cdot 10^{-8}$,

- $P_O(\psi^{(2)}) \approx 8 \cdot 10^{-16}$,

- $P_O(\psi^{(3)}) \approx 7 \cdot 10^{-24}$,

⋮

This may seem incredibly low, but one must remember that a single step requires creating two non-Gaussian states (4 coincident detections of photons) and two coincident detections of vacuum. Two steps already requires the coincident detection of 8 clicks and 4 no-clicks, which already exceeds what is possible with a Megahertz pulsed laser (one $\psi^{(2)}$ event every 10 days approximately!).

A few parameters can help us reach a compromise between entanglement increase and probability. The degree of squeezing λ and the main coefficient in the non-Gaussian

state μ leave room for improvement, but not much. In the case of an optimistic $\lambda = 0.3$ and a conservative $\mu = 0.6$ we obtain $P_O(\psi^{(1)}) \approx 1.7 \cdot 10^{-4}$, $P_O(\psi^{(2)}) \approx 2 \cdot 10^{-8}$, $P_O(\psi^{(3)}) \approx 4 \cdot 10^{-16}$, putting the third iteration out of reach.

This chapter has lead us to identify two major problems in the $(\rho^{(i)}, \rho^{(i)})$ protocol. The exponential use of resources and the exponentially small probabilities of success. This invites us to explore other methods to produce non-Gaussian states and new ways of combining them.

2.4 Non-Gaussian Resource States

Performance of the Non-Gaussian States

Our first goal was to investigate which useful non-Gaussian states can be created in a Procrustean step to feed in the gaussification-distillation protocol. To do so I estimated the outcome when starting with two copies of a two mode squeezed state, using a beam splitter and conditioning success upon detection in both detectors (where detectors only discriminate between presence or absence of photons). I found however that the probabilities were prohibitively small and the resources large (two squeezed states for each non-Gaussian state). One could even wonder, if with real avalanche photo diode (APD) detectors and without neglecting terms of order λ^4 , the non-Gaussian state is non-Gaussian and shows an increase in entanglement. If lossy detectors introduce mixing, the entanglement might not increase that much. Furthermore, if the approximation $|00\rangle + \mu|11\rangle$ doesn't hold, the higher order terms might make it a state close to a Gaussian state (with the coefficient of $|22\rangle$ close to λ^2 , and equally for higher orders) mixed with cross terms. If the non-Gaussian state is indeed too close to a Gaussian one, it will converge in a few iterations to that state, bypassing the entanglement increase.

To study this question I developed a series of procedures in Maple grouped under the module **Quantavo** presented in appendix C.1. With it one can quickly answer this question to a good degree of approximation. The states leading to the Procrustean non-Gaussian mixed state are simulated up to 6 photons and I model our detectors as

imperfect lossy APDs (putting a 50% reflective BS in front of them). That way, for instance, the approximate single photon heralded in the upper branch of fig 2.6 is proportional to:

$$\begin{pmatrix} & \langle 1| & \langle 2| & \langle 3| & \langle 4| & \langle 5| \\ |1\rangle & 0.51 & 0 & 0 & 0 & 0 \\ |2\rangle & 0 & 0.759 \lambda^2 & 0 & 0 & 0 \\ |3\rangle & 0 & 0 & 0.882 \lambda^4 & 0 & 0 \\ |4\rangle & 0 & 0 & 0 & 0.94 \lambda^6 & 0 \\ |5\rangle & 0 & 0 & 0 & 0 & 0.971 \lambda^8 \end{pmatrix}.$$

I will refer to the state created combining this one and the squeezed state (fig. 2.6) using imperfect detection as $\rho_P^{(0)}$, where P stands for Procrustean. To confirm whether the state $\rho_P^{(0)}$ is non-Gaussian or not some elements related to its Wigner function can be studied [Wig32]. The Wigner function, or Wigner quasi-probability-distribution is the Fourier transform of the characteristic function introduced earlier (1.9):

$$\mathcal{W}_\rho(\xi, s) = \frac{1}{(2\pi)^{2N}} \int e^{i\xi\sigma\eta} \chi(\eta) e^{s/4||\eta||^2} d^{2N}\eta \quad (2.9)$$

where $\mathcal{W}_\rho(\xi, 0)$ is the Wigner Function, and the related distribution $\mathcal{W}_\rho(\xi, -1)$ is the Q function. The Wigner function is also Gaussian for Gaussian states and provides an alternative but complete representation of states in phase space [Sch01]. Since $\rho_P^{(0)}$ is a bipartite entangled state, it's Wigner function is a 4-dimensional object which we can't visualize easily. However one can plot partial information about it and draw conclusions about its Gaussianity. I suggest two approaches do to so:

The first one is to plot what would be seen if we measured both modes together. That way, both modes would be merged and could be represented in 2D. For instance, if we wanted to measure $|0, 0\rangle + |1, 1\rangle + |1, 0\rangle$ but the modes were travelling on a same spatial mode we could choose not to distinguish them and effectively measure $|0\rangle + |2\rangle + |1\rangle$.

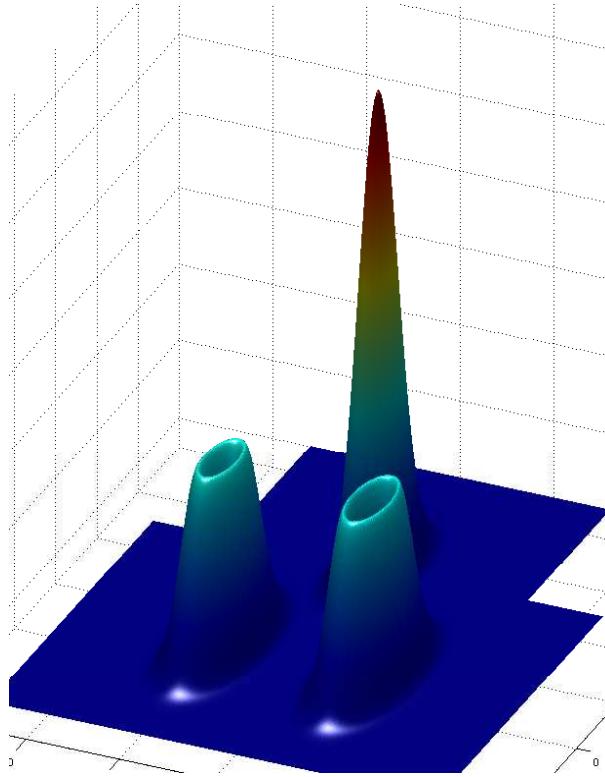


Figure 2.8: Wigner functions of our starting states. The tall Gaussian function corresponds to a single mode of $|Sq\rangle \sim \sum_{n=0}^6 0.1^n |n, n\rangle$ obtained as $\text{Tr}_A \{ |Sq\rangle\langle Sq| \}$. The function on the far right is $\text{Tr}_A \{ \rho_P^{(0)} \}$, the most mixed starting state and has the deepest dip while remaining positive. Finally the non-Gaussian one on the left is $\text{Tr}_A \{ C(|00\rangle + 0.8|11\rangle) \times (\text{c.c.}) \}$ which has a shallow dip and remains positive for all values of X and P .

The idea is to plot the Wigner function of that last state. To show that this preserves Gaussianity consider the following: To merge two modes $|n\rangle|m\rangle$ we need to implement the map $|n, m\rangle \longrightarrow |n + m\rangle$. If this map preserves Gaussian states, then the Wigner function of both $\sum_{n,m} \alpha_{n,m} |n, m\rangle$ and $\sum_{n,m} \alpha_{n,m} |n + m\rangle$ should be Gaussian. This can be shown to hold since the map $|n\rangle_A |m\rangle_B \longrightarrow |n + m\rangle_A |0\rangle_B$ can be implemented with Gaussian operations alone. Indeed combining modes $_A$ and $_B$ at a beam splitter and conditioning upon vacuum detection does just that. Operationally, one would combine the two modes in a BS and homodyne port A only when port B measured the vacuum.

The second method involves tracing out one of the modes and plotting the Wigner

function of the remaining one. Indeed, if ρ is Gaussian, then its characteristic function $\chi_\rho(\xi)$ is too. But from the definition of $\chi_\rho(\xi)$ (1.9) we see that the different modes can be factorised, and therefore the characteristic function without the traced-out mode will be a Gaussian too. This lets us draw the conclusion that if the Wigner function of $\text{Tr}_A \left\{ \rho_P^{(0)} \right\}$ is not Gaussian then the state $\rho_P^{(0)}$ is not Gaussian either.

We plot the Wigner function for a squeezed state and two non-Gaussian states in fig. 2.8. The plot displays the Wigner functions obtained tracing out one mode. In it we can see the striking non-Gaussian character of the state $|00\rangle + 0.8|11\rangle$. It also shows how the state $\rho_P^{(0)}$ which approximates the latter is highly non-Gaussian too in spite of the imperfect detection that generated it. It is quite remarkable that both $\rho_P^{(0)}$ and the pure $|00\rangle + 0.8|11\rangle$ have very similar Wigner functions. However we must remember that we trace out one of the modes. This makes both states mixed while retaining similar photon number distributions. This single mode Wigner function representation obviously conceals some of the state's properties but reveals its non-Gaussian properties. We also studied different ranges of detector efficiencies (40% – 84%) and squeezing $\lambda \in [0.1, 0.5]$ finding similar qualitative results.

I also explored the second way of visualizing Wigner functions of two mode states. I will call this representation, which measures two modes as one, a merged Wigner function. The merged Wigner function also provides an incomplete description but reveals new properties. Obviously the squeezed states will still appear as Gaussians for any λ . However $|00\rangle + 0.8|11\rangle$ and $\rho_P^{(0)}$ display the profile of combinations of even Fock states. Furthermore they exhibit new non-classical features such as negative Wigner functions as shown in Fig. 2.9. Again this is obviously a good method to observe Gaussianity or lack thereof.

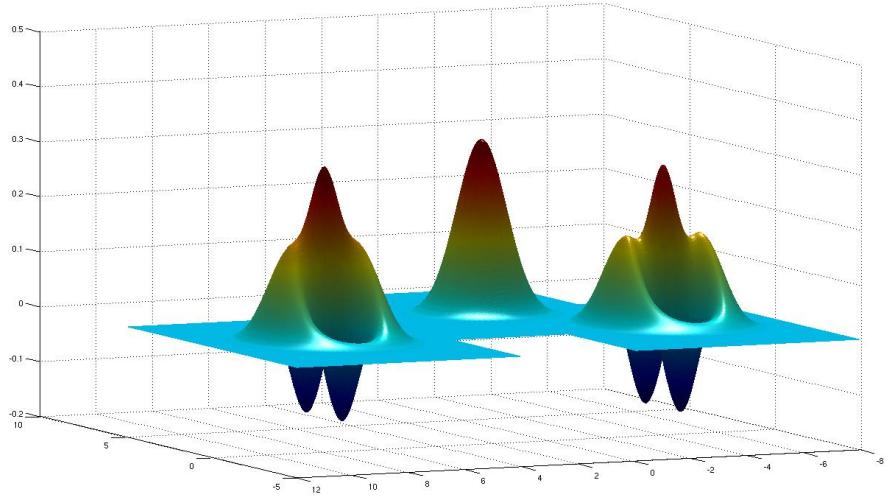


Figure 2.9: Merged Wigner functions of our starting states. The tall Gaussian function corresponds to a single mode of $|Sq\rangle \sim \sum_{n=0}^6 0.1^n |n, n\rangle$ transformed to $|Sq\rangle \sim \sum_{n=0}^6 0.1^n |2n\rangle$. The Wigner function on the right is derived from $\rho_P^{(0)}$ also with the Gaussian procedure $|n, m\rangle \langle s, t| \rightarrow |n + m\rangle \langle s + t|$. On the far left we can see the state corresponding to $|00\rangle + 0.8|11\rangle$ and therefore to $|0\rangle + 0.8|2\rangle$. The merged Wigner functions of both non-Gaussian states display negative values providing no classical analogue for this state.

Initial Entanglement

Beyond the non-Gaussianity we would also like to know if the method to produce $\rho_P^{(0)}$ with reasonable detector efficiencies and squeezing enhances entanglement. Fig. 2.10 compares the entanglement of a pure squeezed state and that of the non-Gaussian state generated from it (2.5). We plot the Logarithmic Negativity:

$$\mathcal{E}_N = \log_2 \|\rho_A^\Gamma\|_1$$

which is a measure of Entanglement [Ple05, PV07] (with Γ_A being the partial transpose with respect to subsystem A , and $\|\cdot\|_1$ the trace norm).

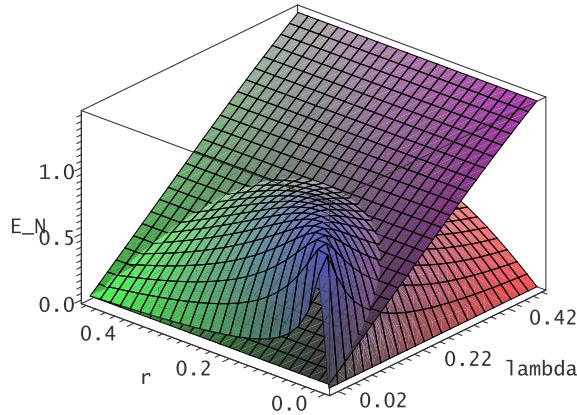


Figure 2.10: Logarithmic Negativity vs. (r, λ) where λ describes the squeezing of our original state $|Sq\rangle \sim \sum_{n=0}^6 \lambda^n |n, n\rangle$ and r is the reflectivity amplitude for the BS involved in the Procrustean preparation of non-Gaussian states from [BESP03]. The surface independent of r is the Log-Negativity of the original squeezed state, and the other surface is the Log-Negativity of the non-Gaussian state $\rho_P^{(0)}$ discussed in the last section.

Thanks to this accurate description we can see that there is a range of (λ, r) for which the Procrustean preparation succeeds at increasing the entanglement (even using 50% efficient APDs). We find however, that as λ increases, the range of adequate r -s diminishes. For $\lambda > 0.3$ the mixing that higher photon number contributions introduce is too great. Looking at the Negativity surface we can choose a pair $(\lambda, r) = (0.1, 0.1)$ such that the initial entanglement is enhanced.

We simulated a few steps of the Gaussification with a mixed and non-Gaussian state

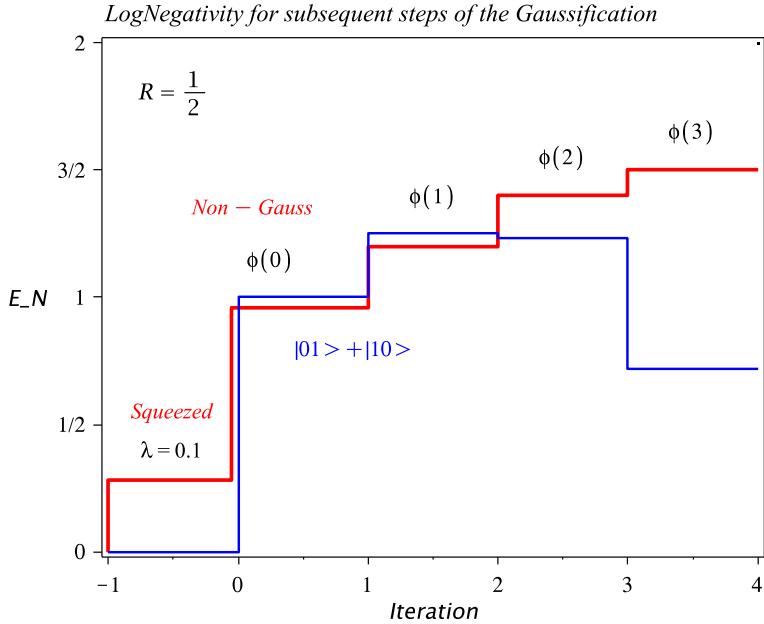


Figure 2.11: Log-Negativity vs. number of iterations for the $(\rho^{(i)}, \rho^{(i)})$ protocol. We display the evolution of two starting states: in RED: The initial $\rho^{(0)}$ is created with the Procrustean protocol discussed earlier (cf. 2.5) choosing the parameters $(\lambda, r) = (0.1, 0.1)$ and using imperfect APDs. in BLUE: The initial state is a perfect $\sqrt{(1/2)}|01\rangle + |10\rangle$ as discussed in 2.4. In both cases the detectors involved in the Gaussification have a 50% efficiency.

having parameters $(\lambda, r) = (0.1, 0.1)$ and ignoring $\mathcal{O}(\lambda^6)$ or higher. The evolution of entanglement for these steps is displayed in Fig. 2.11 in RED. We discover that in spite of the mixing, the protocol could increase the entanglement five-fold in a few steps. We must remember however that the chosen parameters imply unrealistic probabilities. Let us then look at other alternatives preserving this performance.

Single photons on demand

In the Procrustean step that generates $\rho_P^{(0)}$, the upper branch basically prepares a heralded photon which is then mixed at the BS. An alternative would be then to find a deterministic source producing single-photons to a certain degree of approximation. Great progress has been made in recent years in the generation of single photons [BAS⁺06, Shi07, HSG⁺05, WKU⁺05, LO05, LHSJ⁺08, GSV04, DGI⁺07], so this avenue could be an interesting one for the near future. More specifically if we had on-demand ph-

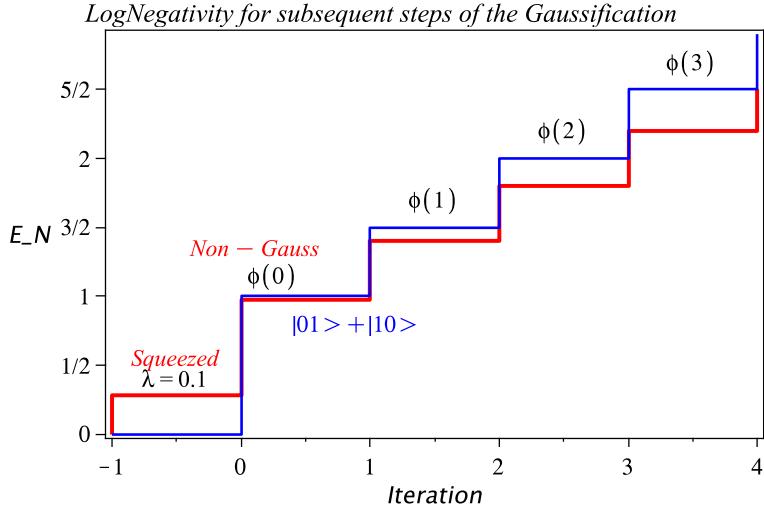


Figure 2.12: Log-Negativity vs. number of iterations for the $(\rho^{(i)}, \rho^{(i)})$ protocol. We display the evolution of two starting states: in RED: The initial $\rho^{(0)}$ is created with the Procrustean protocol discussed earlier (cf. 2.5) choosing the parameters $(\lambda, r) = (0.1, 0.1)$ and using imperfect APDs. in BLUE: The initial state is a perfect $\sqrt{1/2}(|01\rangle + |10\rangle)$ obtained from on-demand sources or heralded photons. In both cases the detectors involved in the Gaussification have a 84% efficiency.

tons matching the frequency of the down-converted photons we could feed them to the upper port of the BS to generate the Non-Gaussian states from eq. (2.5). this would change the probabilities of the distillation presented above ($\lambda = 0.1, \mu = 0.8$) to:

- $P_O(\psi^{(1)}) \approx 3 \cdot 10^{-8} \longrightarrow 3 \cdot 10^{-4}$
- $P_O(\psi^{(2)}) \approx 8 \cdot 10^{-16} \longrightarrow 9 \cdot 10^{-8}$
- $P_O(\psi^{(3)}) \approx 7 \cdot 10^{-24} \longrightarrow 4 \cdot 10^{-15}$
- ⋮

making another iteration possible. Obviously this implementation has its own challenges like mode matching, availability of on-demand photons of matching frequency, etc. A more interesting alternative is to use a different initial state. Having single photons we can easily generate entangled states. Indeed, combining the photon with the vacuum at a BS will create a state proportional to $|01\rangle + |10\rangle$. Assuming we have such a perfect source I have plotted the evolution of the entanglement in Fig. 2.12 and 2.11. We notice that this type of initial state is not as resilient to imperfect detectors as $\rho_P^{(0)}$.

In fact 50% efficient detectors only ensure a temporary entanglement enhancement introducing too much mixing after the second iteration. Fortunately we can easily study the dependence of the achieved Log-Negativity with the detector efficiency as shown in fig. 2.13. However the most important point raised by the use of on-demand photons

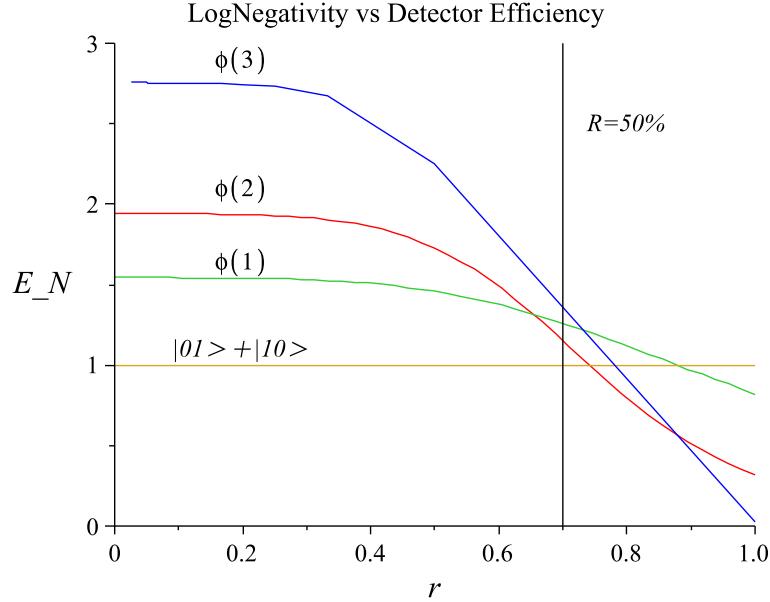


Figure 2.13: Log-Negativity vs. detector efficiency. (The detector efficiency T is modelled with a BS in front of the detector having reflectivity $R = r^2$ and such that $T + R = 1$). The initial state is a perfect $\sqrt{1/2}|01\rangle + |10\rangle$ and we study 3 iterations of the $(\phi^{(i)}, \phi^{(i)})$ protocol. Note that for efficiencies higher than 64% entanglement will increase at each step.

concerns probabilities. Consider the following overall probabilities for two different detection efficiencies:

T=50%	T=70%5A
$P_O(\psi^{(1)}) \approx 0.62$	$P_O(\psi^{(1)}) \approx 0.54$
$P_O(\psi^{(2)}) \approx 0.2$	$P_O(\psi^{(2)}) \approx 0.12$
$P_O(\psi^{(3)}) \approx 0.017$	$P_O(\psi^{(3)}) \approx 0.004$

This makes it possible to perform N iterations if photons can be produced at rates above $1/P_O(\psi^{(N)})$ (a few tens of kHz suffice for the first steps).

The values of the probabilities and Log-Negativity we have obtained are quite encouraging. It means that single photon sources can be used for distillation beyond finite dimensions and into CV distillation with linear optics.

Photon Subtraction and Heralding

Another path for improvement, particularly concerning resources and probabilities is using a single squeezed state to produce a single Non-Gaussian state. Great progress has been made in recent years in the generation of non-Gaussian states with photon subtraction [OP05b, NNNH⁺06, AOBG07, OTBG06]. This technique refers to interposing a highly transmissive beam splitter on the path of the Gaussian state and measuring a photon on the reflected mode. This operation approximates the *subtraction* of a photon from the beam thus implementing a non-Gaussian operation through measurement. Ubiquitous PDC crystals can be used to generate the state from which the photon will be subtracted and we will study this case in detail in chapter 3. Again the parameters from the BS and initial squeezing require careful examination taking into account mixing and imperfect detection. Since only one detection is involved to generate the non-Gaussian states the probabilities are more favourable than for the two-detection- $\rho_P^{(0)}$ discussed earlier.

Given the interesting results one can obtain using single photons, heralding is an interesting substitute for on-demand deterministic single photons. For PDC sources producing two mode squeezed vacuum states it is possible to measure a down converted photon preparing a single photon on the other branch. Thanks to the contact with the *Ultrafast Group* at the University of Oxford I have been able to collaborate in the performance assessment of such schemes. More details about practical proof of principle experiments with heralding and photon subtraction will be discussed in chapter 3.

State Combination

Another problem which seems at the core of the $(\rho^{(i)}, \rho^{(i)})$ protocol is the way states are combined to reach the i -th iteration. Obviously the exponential scaling of non-Gaussian resource states makes probabilities sink very quickly. This suggests that maybe the $(\rho^{(i)}, \rho^{(i)}) \rightarrow \rho^{(i+1)}$ structure is not the optimal one. We can think in broad terms about the protocol and try to extract the crucial element that allows the entanglement increase. So, what does one iteration of the protocol really do to increase the entanglement? Simply put, if no photons are detected in the upper detectors, then the photons from the upper and lower branch are now “part of the output state”. Obviously a more rigorous statement could be constructed with the probability amplitudes, but we only want to gain some intuition. In fact, the way this doubling of the number of photons affects the photon number distribution is the key. When starting from simple states such as $|00\rangle + |11\rangle$ it is easy to get an intuition. At each stage more and more Fock layers are populated, and their respective weights are related to the binomial coefficients (arising from the BS) and to the initial state’s photon number distribution. Gaussian states, due to their thermal distribution exactly match this redistribution of weights, and as more photons are added they exactly reproduce the original distribution. Roughly speaking, if we want to use this philosophy together with linear optics all we need to do is make sure non-Gaussian states are *added* shifting the weight of the smaller Fock layers to the upper ones. However combining always identical states as the $(\rho^{(i)}, \rho^{(i)})$ protocol does is not a necessary requirement for distillation. Coming back to the distillation paradigm, if we have $\rho_0^{\otimes N}$ non-Gaussian states, one needs not combine them in pairs, then make new pairs, etc. Thinking of other more creative ways to combine them is of course possible, but the less structure we impose the harder the analysis will become. Let us then look at the first and simplest modification of the protocol which reduces the exponential consumption of optical elements to linear.

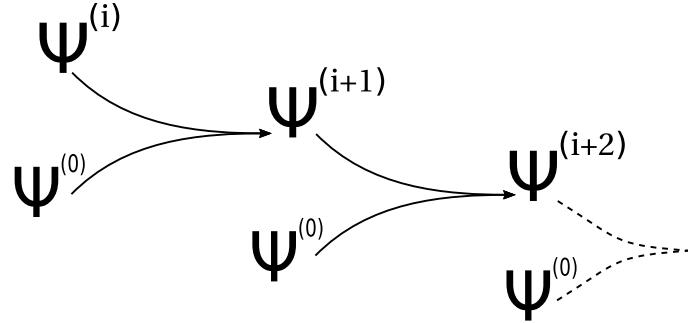


Figure 2.14: Diagram of how the states are combined in the $(\psi^{(i)}, \psi^{(0)})$ setup.

2.5 The $(\rho^{(i)}, \rho^{(0)})$ Protocol

We will study the pure case where the $(|\psi^{(i)}\rangle, |\psi^{(i)}\rangle)$ protocol is substituted by the $(|\psi^{(i)}\rangle, |\psi^{(0)}\rangle)$ one in an attempt to increase the success probabilities and experimental feasibility. We will also study different properties of the new setting like its fixed points and the evolution of simple starting states.

2.5.1 The pure $(|\psi^{(i)}\rangle, |\psi^{(0)}\rangle)$ case.

Description

The protocol considered will mix the two mode state $|\psi^{(i)}\rangle$ in the upper branch with the two mode state $|\psi^{(0)}\rangle$ in the lower branch at 50/50 beam splitters as shown in figure 2.15. After non-detection of photons at the two detectors, we will obtain the two mode state $|\psi^{(i+1)}\rangle$ in the output ports. This last state will be mixed in the next iteration with another $|\psi^{(0)}\rangle$ state, and so repeatedly. In a first instance we will analyse the properties of pure states in Schmidt form $|\psi_0\rangle = \sum_{n=0}^{\infty} \alpha_{n,n} |n, n\rangle$. Two copies of them are mixed at 2 beam splitters which implement the transformation:

$$\begin{pmatrix} \hat{a}_1^\dagger \\ \hat{a}_2^\dagger \end{pmatrix} = \begin{pmatrix} T & R \\ -R^* & T^* \end{pmatrix} \begin{pmatrix} \hat{c}_1^\dagger \\ \hat{c}_2^\dagger \end{pmatrix} \quad (2.10)$$

Let us denote the tensor product between the upper and lower branches by \odot to avoid confusion with the right/left tensor product \otimes . The beam splitters will then transform

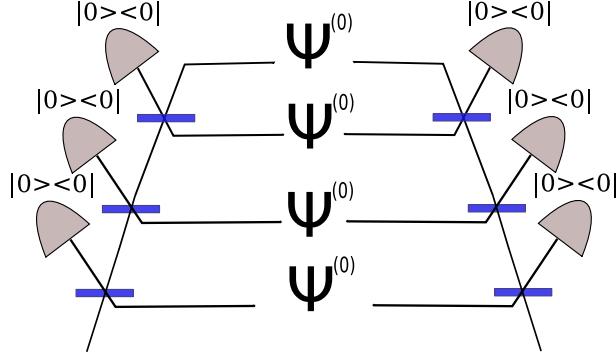


Figure 2.15: $(\psi^{(i)}, \psi^{(0)})$ setup. After combining $\psi^{(0)} \otimes \psi^{(0)}$ and the detection, the output in the lower branch will be $\psi^{(1)}$. This last one, combined with another $\psi^{(0)}$ will become $\psi^{(2)}$, and so on.

the states as follows:

$$\begin{aligned}
 (\hat{U}_{12} \bigcirc \hat{U}_{12}) |\psi_0\rangle |\psi_0\rangle &= (\hat{U}_{12} \bigcirc \hat{U}_{12}) \sum_{n,m=0}^{\infty} \alpha_{n,n} \alpha_{m,m} |n_A, n_B\rangle |m_A, m_B\rangle \\
 &= \sum_{n,m=0}^{\infty} \alpha_{n,n} \alpha_{m,m} \\
 &\quad \frac{(R_A c_2^\dagger + T_A c_1^\dagger)^n}{\sqrt{(n!)}} \frac{(T_A^* c_2^\dagger - R_A^* c_1^\dagger)^m}{\sqrt{(m!)}} \frac{(R_B d_2^\dagger + T_B d_1^\dagger)^n}{\sqrt{(n!)}} \frac{(T_B^* d_2^\dagger - R_B^* d_1^\dagger)^m}{\sqrt{(m!)}} |0\rangle \\
 &= \sum_{n,m=0}^{\infty} \sum_{k,j=0}^n \sum_{l,p=0}^m \Gamma_{j,p}^{k,l} \\
 &\quad |m+n-l-k\rangle_{c2} |l+k\rangle_{c1} |m+n-p-j\rangle_{d2} |p+j\rangle_{d1}
 \end{aligned} \tag{2.11}$$

With

$$\begin{aligned}
 \Gamma_{j,p}^{k,l} &= \frac{\alpha_{n,n} \alpha_{m,m}}{n! m!} \binom{n}{k} \binom{m}{l} \binom{n}{j} \binom{m}{p} \\
 &\quad (R_A)^{n-k} (T_A)^k (T_A^*)^{m-l} (R_A^*)^l (R_B)^{n-j} (T_B)^j (T_B^*)^{m-p} (R_B^*)^p \\
 &\quad (-1)^{l+p} \sqrt{(p+j)!(l+k)!(m+n-p-j)!(m+n-l-k)!}
 \end{aligned}$$

where the indices n and m are self understood for the sake of clarity.

Measurement

One iteration of the protocol can be described by the following operation:

$$|\psi^{(i+1)}\rangle = \langle 0|_{c_1} \langle 0|_{d_1} (\hat{U}_{12} \bigcirc \hat{U}_{12}) |\psi^{(i)}\rangle |\psi^0\rangle$$

It can be seen from eq. (2.11) that the coefficients $\alpha_{n,n}^{(i)}$ after detection of zero photons in the upper branches will be mapped onto the coefficients $\alpha_{n,n}^{(i+1)}$. Let us study the particular case when $R = T = \frac{1}{\sqrt{2}}$,

$$(R_A)^{n-k} (T_A)^k (T_A^*)^{m-l} (R_A^*)^l (R_B)^{n-j} (T_B)^j (T_B^*)^{m-p} (R_B^*)^p = \frac{1}{2^{n+m}}.$$

If we restrict our attention to this case and after detecting 0 photons in the paths c_1 and d_1 the state will have non vanishing terms for $k + l = 0$ and $p + j = 0$. Therefore when $k = l = p = j = 0$ we obtain the final state:

$$|\psi^{(i+1)}\rangle = \sum_{n,m=0}^{\infty} \frac{1}{2^{n+m}} \alpha_{n,n}^{(i)} \alpha_{m,m}^{(0)} \frac{(n+m)!}{n!m!} |n+m\rangle_{c1} |m+n\rangle_{d1}$$

That can be expanded as:

$$\begin{aligned} |\psi^{(i+1)}\rangle &= \frac{1}{2^0} \alpha_{0,0}^{(i)} \alpha_{0,0}^{(0)} |0\rangle |0\rangle + \frac{1}{2^1} (\alpha_{0,0}^{(i)} \alpha_{1,1}^{(0)} + \alpha_{1,1}^{(i)} \alpha_{0,0}^{(0)}) |1\rangle |1\rangle + \\ &\quad \frac{1}{2^2} \left(\frac{2!}{2!1!} \alpha_{0,0}^{(i)} \alpha_{2,2}^{(0)} + \frac{1!}{1!1!} \alpha_{1,1}^{(i)} \alpha_{1,1}^{(0)} + \frac{2!}{1!2!} \alpha_{0,0}^{(i)} \alpha_{2,2}^{(0)} \right) |2\rangle |2\rangle + \dots \\ &\quad \dots + \frac{1}{2^n} \sum_{r=0}^n \frac{n!}{(n-r)!r!} \alpha_{r,r}^{(i)} \alpha_{n-r,n-r}^{(0)} |n\rangle |n\rangle + \dots = \sum_{n=0}^{\infty} \left[\frac{1}{2^n} \sum_{r=0}^n \binom{n}{r} \alpha_{r,r}^{(i)} \alpha_{n-r,n-r}^{(0)} \right] |n\rangle |n\rangle \end{aligned}$$

So that we obtain the recurrence:

$$\alpha_{n,n}^{(i+1)} = \frac{1}{2^n} \sum_{r=0}^n \binom{n}{r} \alpha_{r,r}^{(i)} \alpha_{n-r,n-r}^{(0)} \quad (2.12)$$

Remark: Note that as long as we choose the same phase for both beam splitters the result is unaffected since:

$$\frac{(-c_1^\dagger + c_2^\dagger)^n}{\sqrt{(n!)}} \frac{(c_2^\dagger + c_1^\dagger)^m}{\sqrt{(m!)}} \frac{(-d_1^\dagger + d_2^\dagger)^n}{\sqrt{(n!)}} \frac{(d_2^\dagger + d_1^\dagger)^m}{\sqrt{(m!)}} = (-1)^{2n} \frac{(c_1^\dagger - c_2^\dagger)^n}{\sqrt{(n!)}} \frac{(c_2^\dagger + c_1^\dagger)^m}{\sqrt{(m!)}} \frac{(d_1^\dagger - d_2^\dagger)^n}{\sqrt{(n!)}} \frac{(d_2^\dagger + d_1^\dagger)^m}{\sqrt{(m!)}}$$

2.5.2 Fixed points and Uniqueness

We will designate the map in eq. (2.12) as $\Phi(\alpha^{(i)}) = \alpha^{(i+1)}$. It is then straightforward to see that if our Schmidt coefficients are $\{\alpha_n^{(0)}\}_{n=0}^{\infty} = \{\lambda^n\}_{n=0}^{\infty}$ where λ is a constant satisfying $1 > \lambda \geq 0$, then the states so defined are fixed points of the map achieved by the protocol in the sense that $\Phi(\alpha^{(i)}) = \alpha^{(i)}$ and therefore $|\psi^{(i+1)}\rangle = |\psi^{(i)}\rangle = |\psi^{(0)}\rangle \quad \forall i$.

Uniqueness of the solution:

Any fixed point will have to satisfy the following conditions:

$$1. \alpha_0^{(i)} = \alpha_0^{(i)} \cdot \alpha_0^{(0)} \implies \boxed{\alpha_0^{(i)} = 1}$$

$$2. \alpha_1^{(i)} = \frac{1}{2}(\alpha_0^{(i)} \cdot \alpha_1^{(0)} + \alpha_1^{(i)} \cdot \alpha_0^{(0)}) \implies \alpha_1^{(i)} = \frac{1}{2}(\alpha_1^{(0)} + \alpha_1^{(i)}) \implies \boxed{\alpha_1^{(i)} = \alpha_1^{(0)} = \lambda}$$

$$3. \alpha_2^{(i)} = \frac{1}{4}(\alpha_0^{(i)} \cdot \alpha_2^{(0)} + 2\alpha_1^{(i)} \cdot \alpha_1^{(0)} + \alpha_2^{(i)} \cdot \alpha_0^{(0)}) = \frac{1}{4}(\alpha_2^{(0)} + 2\lambda^2 + \alpha_2^{(i)}) \implies \alpha_2^{(i)} = \frac{1}{3}(\alpha_2^{(0)} + 2\lambda^2) = \alpha_2^{(0)} \implies \boxed{\alpha_2^{(i)} = \lambda^2}$$

$$4. \alpha_3^{(i)} = \frac{1}{7}(\alpha_3^{(0)} + 4\lambda^2\lambda + 2\lambda^3) = \alpha_3^{(0)} \implies \boxed{\alpha_3^{(i)} = \lambda^3}$$

\vdots

\vdots

N. We will prove by induction that if $\alpha_n^{(i)} = \lambda^n$ holds for $n < N$ then it is true for N . Indeed if $\Phi(\alpha^{(i)}) = \alpha^{(i)} \quad \forall i$, then $\alpha_N^{(0)} = \alpha_N^{(1)} = \alpha_N^{(i)}$ and therefore:

$$\begin{aligned}
 \alpha_{N,N}^{(i)} &= \frac{1}{2^N} \sum_{r=0}^N \binom{N}{r} \alpha_{r,r}^{(i)} \alpha_{N-r,N-r}^{(0)} \\
 &= \frac{1}{2^N} \left[\alpha_{N,N}^{(0)} + \sum_{r=1}^{N-1} \binom{N}{r} \alpha_{r,r}^{(i)} \alpha_{N-r,N-r}^{(0)} + \alpha_{N,N}^{(i)} \right] \\
 &= \frac{1}{2^N} \left[\alpha_{N,N}^{(0)} + \sum_{r=1}^{N-1} \binom{N}{r} \lambda^r \lambda^{N-r} + \alpha_{N,N}^{(i)} \right]
 \end{aligned}$$

The invariance $\Phi(\alpha^{(i)}) = \alpha^{(i)} \quad \forall i$ imposes

$$\begin{aligned}
 \alpha_{N,N}^{(i)} &= \frac{1}{2^N - 1} \left[\alpha_{N,N}^{(0)} + \sum_{r=1}^{N-1} \binom{N}{r} \lambda^r \lambda^{N-r} \right] = \alpha_{N,N}^{(0)} \\
 \Rightarrow \alpha_{N,N}^{(0)} &= \frac{1}{2^N - 2} \left[\sum_{r=0}^N \binom{N}{r} \lambda^N - 2\lambda^N \right] \\
 \alpha_{N,N}^{(0)} &= \frac{1}{2^N - 2} (2^N - 2) \lambda^N \quad QED.
 \end{aligned}$$

It can be concluded that only the squeezed vacuum states are fixed points (in the sense described above) of the map defined by this protocol.

2.5.3 Convergence

Convergence of an arbitrary starting state:

It would be interesting to know the answer to the following question:

given $\{\alpha_{n,n}^{(0)}\}_{n=0}^\infty$ with $\alpha_{0,0}^{(0)} = 1$ and $0 \leq \alpha_{1,1}^{(0)} < 1$, what is

$$\lim_{i \rightarrow \infty} \alpha_{n,n}^{(i)} ?$$

Our starting state can be described by $\{\alpha_n^{(0)}\}_{n=0}^{\infty}$. Let us for now consider only an arbitrary set of real numbers that we will call $\{k_n\}_{n=0}^{\infty}$. If so, what will the expression $\{\alpha_n^{(i)}\}_{n=0}^{\infty}$ be? In general, the n -th term after i iterations $\alpha_n^{(i)}$ will be a function of i, n and the set $\{k_n\}_{n=0}^{n-1}$. We can see that as follows: equation 2.12 gives a recurrence relation for each n so that:

$$\bullet \text{ n=0} \quad \alpha_0^{(i+1)} = \alpha_0^{(i)} \cdot \alpha_0^{(0)} \text{(recurrence)} \implies \alpha_0^{(i)} = k_0^{i+1} \text{(general term)}$$

$$\bullet \text{ n=1}$$

$$\begin{aligned} \alpha_1^{(i+1)} &= \frac{1}{2}(\alpha_0^{(i)} \cdot \alpha_1^{(0)} + \alpha_1^{(i)} \cdot \alpha_0^{(0)}) \\ &= \frac{k_0^{i+1} k_1}{2} + \frac{k_0}{2} \alpha_1^i \\ &= \Theta_{1,i} + \frac{k_0}{2} \alpha_1^i \implies \alpha_1^{(i)} = \sum_{l=0}^{i-1} \Theta_{1,i-1-l} k_0^l + \left(\frac{k_0}{2}\right)^i k_1 \end{aligned}$$

$$\vdots$$

$$\bullet \text{ n}$$

$$\begin{aligned} \alpha_n^{(i+1)} &= \frac{1}{2^n} \sum_{r=0}^n \binom{n}{r} \alpha_r^{(i)} \cdot \alpha_{n-r}^{(0)} \\ &= \Theta_{n,i} + \frac{k_0}{2^n} \alpha_n^i \implies \alpha_n^{(i)} = \sum_{l=0}^{i-1} \Theta_{n,i-1-l} k_0^l + \left(\frac{k_0}{2^n}\right)^i k_n \end{aligned} \tag{2.13}$$

Only the term of the sum $\alpha_n^{(i)}$ in eq. 2.13 will define a recurrence, and the other terms will introduce a constant dependent on i, n and $\{k_n\}_{n=0}^n$, namely $\Theta_{n,i}(k_0, k_1, \dots, k_n, i, n)$: a polynomial of all the starting coefficients lower than n . Having broken the symmetry from the original $(|\psi^{(i)}\rangle, |\psi^{(i)}\rangle)$ we can see that some terms that cancelled out in these recursions [BESP03] do not anymore. This makes the general states of this protocol more difficult to analyse. However, at the present time, a few steps of the distillation are

still a challenging and interesting task. We will therefore analyse in detail the evolution of a couple of simple states.

2.5.4 The $|\psi^{(0)}\rangle = |0,0\rangle + \lambda|1,1\rangle$ case

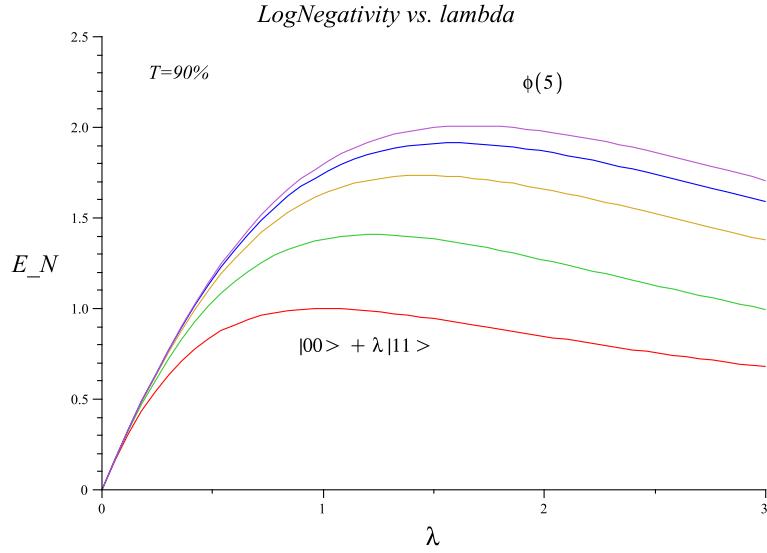


Figure 2.16: Log-Negativity vs. lambda. The initial state is a perfect $\sqrt{1/2}(|00\rangle + \lambda|11\rangle)$ and we study 5 iterations of the $(\phi^{(i)}, \phi^{(0)})$ protocol. (The increasing order of the curves matches that of the iteration number). To simulate the protocol we used detectors with 90% efficiency.

Interestingly, if we take the initial state: $|\psi^{(0)}\rangle = \frac{1}{\sqrt{1+\lambda^2}}(|0,0\rangle + \lambda|1,1\rangle)$, therefore $\left\{ \{\alpha_n^{(0)}\}_{n=0}^{\infty} / \alpha_n^{(0)} = 0 \quad \forall n > 1 \right\}$ the recurrence in eq. (2.12) will become:

$$\alpha_n^{(i+1)} = \frac{n\lambda}{2^n} \alpha_{n-1}^{(i)} + \frac{1}{2^n} \alpha_n^{(i)}$$

which leads in the limit $i \rightarrow \infty$ to a state of the form:

$$\lim_{i \rightarrow \infty} |\psi^{(i)}\rangle = K \left(|0,0\rangle + \lambda|1,1\rangle + \frac{2}{3}\lambda^2|2,2\rangle + \frac{2}{7}\lambda^3|3,3\rangle + \dots + \frac{n!}{\prod_{j=1}^n (2^j - 1)} \lambda^n |n,n\rangle \right)$$

Therefore this new recurrence implies:

$$\lim_{i \rightarrow \infty} \alpha_n^i = \frac{n!}{\prod_{j=1}^n (2^j - 1)} \lambda^n \quad (2.14)$$

as well as

$$\alpha_n^i = 0 \quad \text{if} \quad i \leq n - 1$$

We can illustrate the evolution of the coefficients with each iteration in the following matrix:

	$i \longrightarrow$							
	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	λ	λ	λ	λ	λ	λ	λ	λ
$n \downarrow$	0.0	$0.5 \lambda^2$	$0.6 \lambda^2$	$0.6 \lambda^2$	$0.6 \lambda^2$	$0.6 \lambda^2$	$0.6 \lambda^2$	$0.6 \lambda^2$
	0.0	0.0	$0.2 \lambda^3$	$0.2 \lambda^3$	$0.3 \lambda^3$	$0.3 \lambda^3$	$0.3 \lambda^3$	$0.3 \lambda^3$
	0.0	0.0	0.0	$0.04 \lambda^4$	$0.06 \lambda^4$	$0.07 \lambda^4$	$0.07 \lambda^4$	$0.07 \lambda^4$
	0.0	0.0	0.0	0.0	$0.007 \lambda^5$	$0.01 \lambda^5$	$0.01 \lambda^5$	$0.01 \lambda^5$
	0.0	0.0	0.0	0.0	0.0	$0.0007 \lambda^6$	$0.001 \lambda^6$	$0.001 \lambda^6$
	0.0	0.0	0.0	0.0	0.0	0.0	$0.00003 \lambda^7$	$0.00005 \lambda^7$
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	$0.000001 \lambda^8$

This clearly indicates (as does eq. 2.14) that the Entanglement is going to saturate eventually after a few iterations since any large coefficient will be damped by the term $1/\prod_{j=1}^n (2^j - 1)$. This can be seen to hold also for imperfect detectors as evidenced by Fig. 2.16. In this case $\lambda = 1$ will not be optimal, but rather $\lambda = 2$. Again the resilience to detector loss is displayed in fig. 2.17. It is interesting to note that for the above starting state the $(\rho^{(i)}, \rho^{(0)})$ protocol drives $\lim_{i \rightarrow \infty} \rho^{(i)}$ to a non-Gaussian state. This contrasts with the $(\rho^{(i)}, \rho^{(i)})$ protocol but has its advantages. For example the generation of non-Gaussian states could be geared towards the violation of Bell inequalities.

As we discussed earlier for the $(\rho^{(i)}, \rho^{(i)})$ protocol, single photons sources make it worthwhile studying the performance of initial states of the form $|01\rangle + |10\rangle$. We find however that they are not very robust to detector loss as fig. 2.18 shows. In fact detectors need efficiencies close to 95% to achieve 4 iterations making the use of such states impractical in the $(\rho^{(i)}, \rho^{(0)})$ setup. Since 95% is already quite high, calculation were not made beyond 4 iterations but the tendency suggests a need for even higher

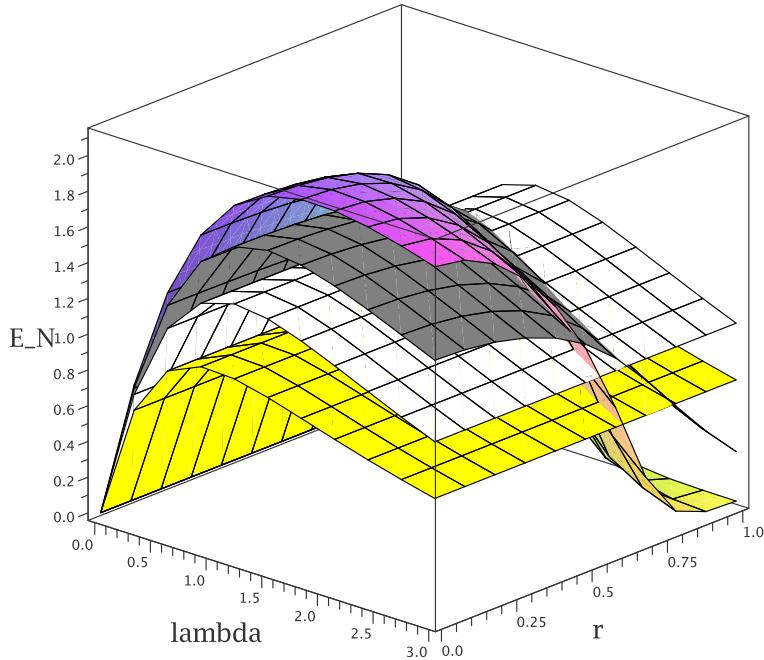


Figure 2.17: Log-Negativity vs. detector efficiency and λ from the initial state $\sim |00\rangle + \lambda|11\rangle$. (The detector efficiency T is modelled with a BS in front of the detector having reflectivity $R = r^2$ and such that $T + R = 1$). We study 3 iterations of the $(\phi^{(i)}, \phi^{(i)})$ protocol where the yellow surface illustrates the entanglement of the initial state. Note that for efficiencies higher than 64% entanglement will increase at each step.

efficiencies as iterations increase.

2.5.5 The $(|\psi^{(i)}\rangle, |\psi^{(0)}\rangle)$ parameters

2.5.6 Introduction

In the following section we will investigate if any further improvement can be achieved choosing specific beam splitters (the same on both sides). In particular, starting with the simple case of $|\psi^{(0)}\rangle = |0,0\rangle + \lambda|1,1\rangle$, we will study the relationship between R, T and λ that optimizes the increase of entanglement. In this case then Γ from eq. (2.11) becomes:

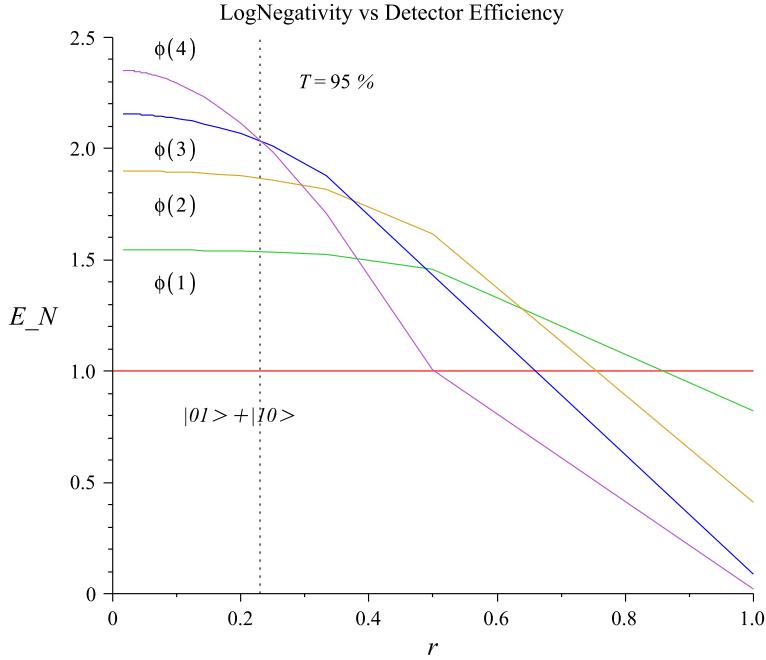


Figure 2.18: Log-Negativity vs. detector efficiency. (The detector efficiency T is modelled with a BS in front of the detector having reflectivity $R = r5A5A5A5A5A^2$ and such that $T + R = 1$). The initial state is proportional to a perfect $|01\rangle + |10\rangle$ and we study 4 iterations of the $(\phi^{(i)}, \phi^{(0)})$ protocol. Note that only for efficiencies higher than 95% entanglement will increase at each step.

$$\Gamma_{n,m,l,k,p,j} = \frac{1}{n!m!} \binom{n}{k} \binom{m}{l} \binom{n}{j} \binom{m}{p} \frac{(R)^{2n-k+l-j+p}}{(T)^{2m-l+k-p+j}} (-1)^{l+p} \sqrt{(p+j)!(l+k)!(m+n-p-j)!(m+n-l-k)!}$$

In this first approach, we have taken R and T to be real for simplicity. Now, after detecting 0 photons in the paths c1 and d1 the state will have non vanishing terms for $k + l = 0$ and $p + j = 0$ and therefore when $k = l = p = j = 0$ results in a final state:

$$\sum_{n,m=0}^{\infty} \alpha_{n,n}^{(i)} \alpha_{m,m}^{(0)} R^{2n} T^{2m} \frac{(n+m)!}{n!m!} |n+m\rangle_{c2} |m+n\rangle_{d2}$$

Which can be rewritten as

$$|\psi^{(i+1)}\rangle = \sum_{n=0}^{\infty} \left[\sum_{r=0}^n \binom{n}{r} R^{2r} T^{2n-2r} \alpha_{r,r}^{(i)} \alpha_{n-r,n-r}^{(0)} \right] |n\rangle |n\rangle \quad (2.15)$$

So that:

$$\alpha_{n,n}^{(i+1)} = \sum_{r=0}^n \binom{n}{r} R^{2r} T^{2n-2r} \alpha_{r,r}^{(i)} \alpha_{n-r,n-r}^{(0)} \quad (2.16)$$

Again for the simple starting state $|\psi^{(0)}\rangle = |0,0\rangle + \lambda|1,1\rangle$ the second coefficient of the recurrence becomes:

$$\alpha_1^{(i+1)} = T^2 \lambda + R^2 \alpha_1^{(i)}$$

Which again guarantees $\alpha_1^{(i)} = \lambda \forall i$ since $R^2 + T^2 = 1$. Therefore the above general recurrence (2.16) will become:

$$\alpha_n^{(i+1)} = R^{2n} \left\{ n \left(\frac{1}{R^2} - 1 \right) \lambda \alpha_{n-1}^{(i)} + \alpha_n^{(i)} \right\}$$

Very quickly after a few iterations the state becomes quite complex and it becomes easier to explore the interesting parameters with the **Quantavo** procedures. The basic conclusion is that other values different from $R = \sqrt{1/2}$ can guarantee maximum entanglement increase, and therefore it is a parameter worth optimizing when designing practical distillation implementations.

2.6 Work in Progress: Homodyne Distillation Pumping

In this section we give an introduction to work in progress aimed at improving the performance of a continuous variable (CV) linear optics (LO) entanglement distillation scheme. In the previous section we discussed the strengths and weaknesses of the Gaussification and distillation procedure from [BESP03, EBSP04]. Although not crucial, inefficiencies in detection affect the performance of the protocol. This section

discusses some of the options available to improve such performance and the way to assess it.

There are currently very precise and efficient photodiode detectors with near single photon sensitivity. However the optimal sensitivity is only achieved above certain intensity. When only a few photons arrive at the detector, dark counts are too important and mask the signal. However when many photons arrive it has a very good response and can distinguish N and $N+1$ photons. How does one then measure low-intensity states with such detectors? The response to that question, known as quantum homodyne detection is the following: If one shines a coherent state at a 50/50 BS and detects both output ports with two photodiodes one will find the same intensity on each. However if on the other incoming port of the BS our low-intensity state ρ meets the coherent state (or Local Oscillator) $|\alpha\rangle$, then it will perturb the outputs which won't be equal anymore. The difference in the intensities seen by each detector allow us to reconstruct ρ [Leo97]. In fact, the coherent state (also called phase reference or local oscillator) in a setup of beam splitters before 2 or 4 detectors can effectively implement the POVMs $|\alpha\rangle\langle\alpha|$, $|X\rangle\langle X|$ or $|P\rangle\langle P|$. This can be done with detector efficiencies above 90%, making it an extraordinary tool.

In our case, we would like to describe the evolution of states transformed according to the Gaussification protocol when we exchange the vacuum projection $|0\rangle\langle 0|$ with 8 port homodyning $|\alpha\rangle\langle\alpha|$ as was suggested in [EBSP04, EPB⁺07]. Of course it should be apparent that in the limit $\alpha \rightarrow 0$ the measurements become formally equivalent. The practical realization however is more subtle and the formalism has to be studied carefully. Our aim is then to describe the evolution of the states through such a protocol characterizing their entanglement and purity evolution. We will introduce some of the notation, techniques and formalism used to tackle this problem.

2.6.1 Non-Gaussian Previous Step

Our initial resource states are many copies of two mode squeezed vacuum states $\rho_{sq}^{\otimes N}$ that we expand in the Fock basis as:

$$\rho_{sq} \sim \sum_{n,m=0}^{\infty} \lambda^{n+m} |n, n\rangle \langle m, m|.$$

If we are to use Gaussian operations in the distillation scheme (BS, homodyning, phase shifters) then these states must be converted to non-Gaussian states prior to the application of the protocol. This can be done having recourse to a non Gaussian measurement (for instance a detection with a non-number-resolving detector) resulting in a Procrustean preparation of the required states (see chapters 2 and 3 for more details). Following the photon subtraction introduced earlier we can combine our two mode state with the vacuum at a BS and make a measurement with an APD detector. In the ideal case, the “click” outcome will be described with the Kraus operator $\mathbb{I} - |0\rangle\langle 0|$. We present these operations both in Hilbert space and in the language of first and second moments:

Input State

Hilbert space: $\rho_{sq} \otimes |0\rangle\langle 0|$

First and second moments: $\vec{d} = \vec{0}, \gamma = \begin{pmatrix} c & 0 & -s & 0 & 0 & 0 \\ 0 & c & 0 & s & 0 & 0 \\ s & 0 & c & 0 & 0 & 0 \\ 0 & s & 0 & c & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$

Where d and γ are the first and second moments (or first moments and covariance matrix) as introduced in 1.9. Note that the covariance matrix has been defined in terms of the squeezing parameter r as $c = \cosh(r)$, $s = \sinh(r)$.

Beam splitter operation

This operation corresponds to subtracting a photon in one of the modes of the two mode squeezed state:

$$\text{Hilbert Space: } \rho' = U (\rho_{sq} \otimes |0\rangle\langle 0|) U^\dagger$$

$$\text{First and second moments: } d' = S \vec{d} = \vec{0}$$

$$\gamma' = S\gamma S^T = \frac{1}{2} \begin{bmatrix} 2c & 0 & -\sqrt{2}s & 0 & \sqrt{2}s & 0 \\ 0 & 2c & 0 & \sqrt{2}s & 0 & -\sqrt{2}s \\ -\sqrt{2}s & 0 & 1+c & 0 & 1-c & 0 \\ 0 & \sqrt{2}s & 0 & 1+c & 0 & 1-c \\ \sqrt{2}s & 0 & 1-c & 0 & 1+c & 0 \\ 0 & -\sqrt{2}s & 0 & 1-c & 0 & 1+c \end{bmatrix}$$

$$\text{where } S_{6 \times 6} = \begin{pmatrix} \mathbb{I} & 0 & 0 \\ 0 & \mathbb{I} & \mathbb{I} \\ 0 & -\mathbb{I} & \mathbb{I} \end{pmatrix}, \text{ meaning } \mathbb{I} = \mathbb{I}_{2 \times 2}.$$

Note that S is the symplectic transformation corresponding to a BS.

APD measurement

$$\text{Hilbert space: } \rho'' = \mathbf{M}\rho'\mathbf{M}^\dagger$$

$$\text{First and second moments: } \vec{d}'' = ?$$

$$\gamma'' = \text{Shur}(\gamma') = A - C(B_0 + \mathbb{I})^+ C^T,$$

$$\text{where } \gamma' = \left(\begin{array}{ccc|c} & & & C_1 \\ & & & C_2 \\ A & & & C_3 \\ \hline C_1 & C_2 & C_3 & B_0 \end{array} \right),$$

$$\mathbf{M} := \mathbb{I} - |0\rangle\langle 0|,$$

$(B_0 + \mathbb{I})^+$ is the Moore-Penrose pseudo-inverse (B.1).

All this gives then in return the first and second moments of our starting states as:

$$\gamma'' = \begin{pmatrix} c - \frac{s^2}{3+c} & 0 & \gamma''_{1,3} & 0 \\ 0 & c - \frac{s^2}{3+c} & 0 & \gamma''_{2,4} \\ \gamma''_{3,1} & 0 & 1/2 + 1/2c - 2 \frac{(1/2-1/2c)^2}{3+c} & 0 \\ 0 & \gamma''_{4,2} & 0 & 1/2 + 1/2c - 2 \frac{(1/2-1/2c)^2}{3+c} \end{pmatrix}$$

$$\vec{d}'' = 0$$

With,

$$\begin{aligned} \gamma''_{3,1} &= \gamma''_{1,3} = -1/2 \sqrt{2}s - \frac{\sqrt{2}s(1/2-1/2c)}{3+c} \\ \gamma''_{2,4} &= \gamma''_{4,2} = 1/2 \sqrt{2}s + \frac{\sqrt{2}s(1/2-1/2c)}{3+c}. \end{aligned}$$

More details about how to describe the measurement results and probabilities in the language of moments and covariance matrix for Gaussian states can be found in appendix B.1. One interesting remark comes when considering the non-Gaussian operation:

$$\begin{aligned} \rho_{AB} &\longrightarrow (\mathbb{I} - |0\rangle\langle 0|) U \rho_{AB} U^\dagger (\mathbb{I} - |0\rangle\langle 0|) \\ &= U \rho_{AB} U^\dagger - \langle 0| U \rho_{AB} U^\dagger |0\rangle \\ &= \rho_1 - \rho_2 \end{aligned}$$

Indeed ρ_1 and ρ_2 are Gaussian when considered separately even though their difference isn't. That way, to study the evolution of the non-Gaussian state through the distillation protocol we could simply keep a list of the evolution of the covariance matrices for each state in the sum. This decomposition of the non-Gaussian state can help us describe this evolution.

2.6.2 Evolution in the Distillation Process

The notation and concepts introduced in the last section can help us describe the evolution of the states through subsequent stages of the protocol. One approach to explore this case is to try to make one protocol a certain limit of the other. Or possibly, since we know well the $(\rho^{(i)}, \rho^{(i)})$ protocol with the vacuum projection, trying to decouple the description of that same protocol from the displacement or squeezing that transforms $|0\rangle\langle 0|$ into say $|\alpha\rangle\langle\alpha|$ or $|x\rangle\langle x|$.

First and second moments

The map that describes one iteration of the protocol remains almost the same as in chapter 2, and thus,

$$\rho^{(i+1)} = \langle\xi|\langle\xi| (U \otimes U) (\rho^{(i)} \odot \rho^{(i)}) (U \otimes U)^\dagger |\xi\rangle|\xi\rangle. \quad (2.17)$$

We remind that $U \otimes U$ implement the two beam splitter (BS) operations between the upper and lower modes and that we measure only the two upper ports of the BSs. In the above map, the vacuum projection from Eq. (2.1) has been exchanged for the homodyning POVM $|\xi\rangle\langle\xi|$.

Interestingly, if $|\xi\rangle\langle\xi|$ represents direct homodyning, and therefore the idealized projection onto a quadrature, its relation to the vacuum projection will be given by the squeezing and displacement operator:

$$|\xi\rangle = S(s)D(d)|0\rangle$$

where $s \in \mathbb{R} - \{0\}$ is the squeezing parameter and d the displacement as shown in Fig.2.19. Let us then look at the case of zero displacement, i.e. only squeezing. Since BS and squeezing commute, $[U, S \odot S] = 0$, then the map from eq. 2.17 can be rewritten as,

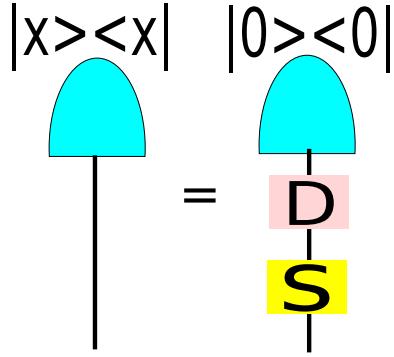


Figure 2.19: Equivalence of direct homodyning to local squeezing and displacement before vacuum projection.

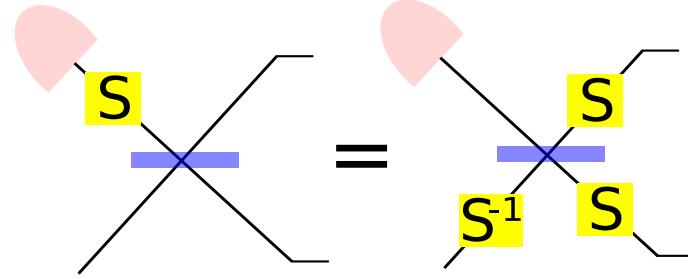


Figure 2.20: Equivalence of projecting onto a squeezed vacuum with squeezing the states and anti-squeezing the output. Here, the gray detector pictured implements the vacuum projection.

$$\begin{aligned}\langle \xi | U &= \langle 0 | (S(s)^\dagger \odot \mathbb{I}) U \\ &= \langle 0 | (\mathbb{I} \odot S(s)) U (S(s)^\dagger \odot S(s)^\dagger).\end{aligned}$$

This expression can be reinterpreted as saying that projecting onto the squeezed vacuum is equivalent to squeezing the two states coming to the BS and then making an inverse squeezing operation on the output port. This relation is shown in fig. 2.20.

We may also consider the case of displacement alone. In this case, if the vacuum projection has a displacement $d = (0, 0)$ in the (x, p) phase-space coordinates, then the displacement operator shown in fig. 2.19 transforms it to $d = (d_1, d_2)$. Now the BS's symplectic transformation will transform the displacements coming into its two ports:

$d = (d_1, d_2, u_1, u_2)$ according to,

$$d' = \frac{1}{2} \begin{pmatrix} \mathbb{I} & \mathbb{I} \\ -\mathbb{I} & \mathbb{I} \end{pmatrix} d^T = \begin{pmatrix} d_1 - e_1 \\ d_2 + e_2 \\ e_1 - d_1 \\ e_2 - d_2 \end{pmatrix}$$

where \mathbb{I} are two by two identity matrices. Therefore, if the detected mode has (d_1, d_2) and the other mode is empty $(0, 0)$ we can make a displacement before detection equivalent to displacing the states before the BS:

$$\langle 0 | (D(d) \odot \mathbb{I}) U = \langle 0 | U (D(d/\sqrt{1/2}) D(d/\sqrt{1/2})) \quad (2.18)$$

However, since we don't know the result of the displacement until it occurs probabilistically a more interesting question might be, what displacement should we implement to the output mode to compensate for the one the measurement introduced. The displacements resulting from the detection will depend on the correlations between the bipartite states and therefore will depend on the second moments. The final first moments for the bipartite state will then become after measurement of d_{AB} :

$$d_{AB}^T = \frac{\gamma_2 - \gamma_1}{2} \left(\frac{\gamma_2 - \gamma_1}{2} + \mathbb{I}_{4 \times 4} \right)^{-1} d_{AB}^T \quad (2.19)$$

That way, one could correct the measured displacement and come back to the results from the vacuum-projection protocol.

2.6.3 Open Questions

If real-time feedback can be implemented from the measurement results to the output states this could lead to an Efficient Entanglement Pumping Scheme using Linear Optics. However this is work in progress and a few answers still need to be addressed in full detail. The full treatment of the evolution of non-Gaussian states through the proto-

col needs to be made explicit. The full probability calculation is certainly another open question, although it should be relatively straightforward as detailed in appendix B.1. Finally, if after i steps we have obtained the normalized state:

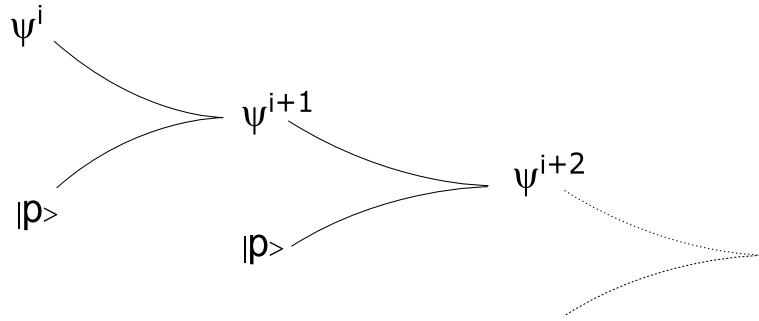
$$\rho^{(i)} = \sum_{j=0}^{N(i)} \kappa_j \rho_j^{(i)}$$

Where $\rho_j^{(i)}$ are Gaussian states and $\sum_{j=0}^{N(i)} \kappa_j = 1$, then we have a list of covariance matrices describing the state. How to infer the entanglement content of this state with that list is still an open question. In other words, providing a table (or expression) which relates the outcome of any possible homodyne measurements at each iteration with the entanglement increase is our next goal.

2.7 Open Problems and Conclusion

2.7.1 Conclusions

We have reviewed the ($\rho^{(i)}$, $\rho^{(i)}$) protocol focusing on its weaknesses, particularly from the point of view of probabilities and use of optical resources. However we have also presented many avenues for improvement and exploration. We have seen that alternative versions of the protocol can achieve entanglement distillation and that simple modifications of the non-Gaussian state generation can greatly enhance the probabilities of success. We have studied the properties of the ($\rho^{(i)}$, $\rho^{(0)}$) protocol finding that it shares the fixed points with the original one but that its asymptotic convergence is still unclear. The way both these protocols distill purity and entanglement greatly depends on the original states, the way we combine them and on the efficiencies and losses involved. To address this I have shown examples of how the **Quantavo** Maple procedures can help us assess the suitability of various strategies. Its combination of analytic and numerical tools allows us to quickly build models with mixed states, imperfect detectors, loss and calculate relevant entanglement measures from it. More details and calculations about realistic implementations can also be found in chapter 3.


 Figure 2.21: Combining $|\psi^{(i)}\rangle$ with $|p\rangle$ states

The specific possibilities that CV distillation with LO offers are still largely unexplored and many variations of the presented protocol can be studied. For example a simple variation on the original $(\rho^{(i)}, \rho^{(i)})$ protocol is simply feeding a state, different from $\rho^{(0)}$ in one of the ports of each iteration. This ρ could be the one resulting after one iteration or an altogether different one obtained for instance from heralded photons, on-demand photons or other sources different from the $\rho^{(i)}$ we are trying to purify and distill. In a more general fashion and trying to work around the quick saturation that the $(\rho^{(i)}, \rho^{(0)})$ version exhibits we could use an adaptive $(\rho^{(i)}, \sigma_i)$, where the non-Gaussian states σ_i are engineered (adjusting BS and detection events) to continue increasing the entanglement of the target state. Another area that is promising involves the distillation of Schrödinger cat states and their behaviour in this setup. Finally the way to mix the states and the possibility of storage or recycling of *unsuccessful* events should also be explored further. As a transition to the next chapter we remind that these protocols work well even with inefficient detectors but that this imposes constraints on probabilities and performance. To tackle this problem the use of the very efficient homodyne detection was suggested in [BESP03, EPB⁺07] and the first steps towards their use were shown in this chapter.

3

First Steps Towards Experimental Procrustean Distillation of Entanglement

In previous chapters we have studied different variations on the distillation protocol from [BESP03, EBSP04] exploring also how different detection strategies change their performance. We have explored briefly what resource states (i.e. non-Gaussian states) are more appropriate to achieve the highest degree of entanglement or entanglement increase with the protocol. However we have not yet looked in detail at the practical generation of these states with current technology. Given that the non-Gaussian states are the cornerstone of the distillation protocols, working towards their practical generation and manipulation is crucial. Is it possible, with current technology, to implement simple proof-of-principle experiments on entanglement distillation (ED) in continuous variables (CV)? I will present in this chapter the calculations and tools necessary to assess the feasibility of some of these experiments.

3.1 Introduction

Each stage of the distillation protocols presented in chapter 2 above requires the generation of non-Gaussian states. However that is not the sole motivation to study them.

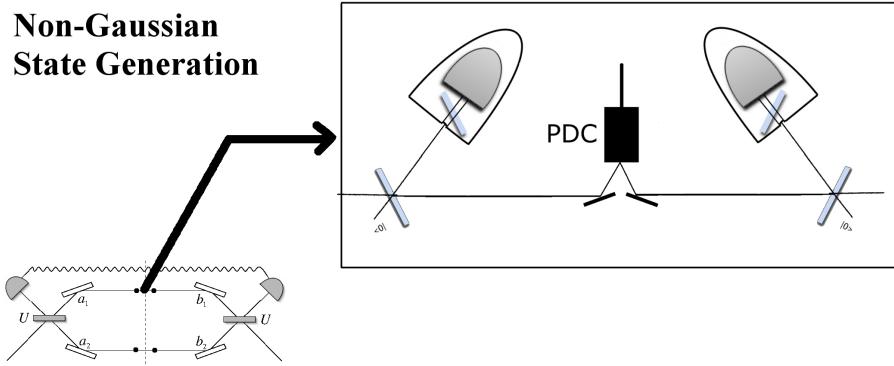


Figure 3.1: At each step of the distillation protocol we must generate non-Gaussian states. Here we depict a double-photon subtraction setup. The Parametric Down Conversion (PDC) produces a two mode state. A beam splitter (BS) is placed in each mode and a photon detected on each reflected path. Imperfect detectors are pictured with a BS in front of them to account for lost photons.

Non-Gaussian state generation can also shed light on the foundations of quantum optics, allowing the violation of bell inequalities [LV95, BW99b, BW99a] and providing key elements for CV quantum computing [LB99, GS07, BS02]. Generating entangled non-Gaussian states is a difficult task. Most of the standard linear optics tools such as beam splitters, phase shifters, squeezers or homodyne measurement implement Gaussian operations and therefore preserve the Gaussian character. That implies that using them on optical squeezed, displaced or coherent states will not result in non-Gaussian states [GIC02, ESP02a, Fcv02]. Of course some states become non-Gaussian in the process of decoherence, for example from non-Gaussian phase diffusion [HFD⁺07]. In that case the distillation scheme from [BESP03, EBSP04] can purify these non-Gaussian states and retrieve the original entanglement prior to the de-Gaussification. However we are more interested in increasing the entanglement of states subject to Gaussian noise or which have pure Gaussian entanglement to begin with. Generating non-Gaussian states without deteriorating their entanglement and purity requires then non-linear interactions between photons. While in principle Kerr non-linearities can implement such interactions the magnitude of the effect is still minuscule. Alternatively measurement can introduce non-linearities large enough to produce non-Gaussian states as has been

demonstrated experimentally [AOBG07, OTBG06, OTBLG06, ODTBG07, WTFS07]. However increasing entanglement with local operations and classical communication (LOCC) alone is as yet a standing challenge. For a specific use in distillation of CV entanglement we have described the generation of non-Gaussian states with the setup from [EBSP04] depicted in Fig. 2.4. In it, we try to obtain, to a good degree of approximation, a state close to $|\phi_0\rangle \sim |00\rangle + \mu|11\rangle$ with μ as close to 1 as possible. However, generating $|\phi_0\rangle$ requires two squeezed vacuum states and the improbable coincident detection of a photon in each of two detectors. A more sparing setup to generate non-Gaussian states is for instance the one described in Fig. 3.1. This setup does not require mode matching two squeezed states but uses a single one instead. Why then has it not been implemented? Various elements add to the difficulty of a real implementations:

- Inefficient photo-detectors, coupling losses and filtering will introduce mixing [OP05a, OP05b] (thus reducing the entanglement).
- To rigorously prove an increase in entanglement one has to be able to measure the amount of entanglement a CV state possesses which is not a trivial task.

I will present some of the calculations and parameter explorations carried out to spur progress in this area.

3.2 Squeezed Vacuum States, How useful are they?

So far we have described our squeezed vacuum states with $\sum_n \lambda^n |n, n\rangle$. This description makes the implicit assumption that the parametric down conversion process generates two mode squeezed states. However it is well known that photons produced in PDC crystals have a richer mode structure [LWE00, LE04, PK05]. The population of orbital angular momentum and spectral modes therefore affects the photon-number distribution distinguishing it from the usual two mode description [ACRL⁺08, WRFB08]. This implies that for multiphoton realizations involving interference a careful design of the initial states must be ensured. Fortunately many efforts have been made to address this problem: The first strategy is to generate the down-converted photons in single-mode

fibres or wave-guides to avoid the spread in free space [FAW⁺05, AFW⁺06, FAWR07]. The second strategy involves engineering the spectral mode structure to achieve higher purity [Mos07, CLS⁺08, MLSW08]. Finally, and in a different spirit, one can of course exploit the entanglement and correlations in the other modes. In fact, entanglement concentration has been shown to be possible with orbital angular momentum from PDC photons [AJWT⁺03].

For our calculations we will assume we have a spectrally pure squeezed vacuum state as in [MLSW08]. Even so there will be sources of mixing in any LO circuit. Indeed coupling the state to a fibre to implement a BS or for detection as well as filtering will discard some modes in our state. This effect results in loss of photons and we can therefore model it with a BS coupled to the environment. Effectively the mode reflected at the BS will represent all the modes lost in various filtering processes. The PDC source that the Ultrafast group at Oxford University could potentially use shows indeed a small coupling efficiency to single mode fibres. Each PDC beam couples to a fibre with at most 30% efficiency and we will use this parameter in our mixed state calculations, exploring if lower or higher efficiencies are respectively tolerable or beneficial.

3.3 Ideal Photon Subtraction

Let us look at the optimal pure case with perfect detection in order to estimate the limits of this technique.

3.3.1 Starting State

We will assume that the parametric down conversion (PDC) crystal produces a pure two mode squeezed vacuum state described by $|\psi_\lambda\rangle = \sqrt{1 - \lambda^2} \sum_{n=0}^{\infty} \lambda^n |n, n\rangle$. Its entanglement can be evaluated with the entropy of entanglement (i.e. the Von Neumann

entropy of the reduced state):

$$\begin{aligned}
 S_{ini}(\lambda) &= - \sum_{k=0}^{\infty} \lambda^{2k} (1 - \lambda^2) \log_2 (\lambda^{2k} (1 - \lambda^2)) \\
 &= \frac{\lambda^2 \log_2(\lambda^2)}{(\lambda^2 - 1)} - \log_2(1 - \lambda^2)
 \end{aligned} \tag{3.1}$$

or with the Logarithmic Negativity as:

$$E_N(\rho) = \log_2 \| \rho^\Gamma \|_1 = \log_2 \left(\frac{1 + \lambda}{1 - \lambda} \right) \tag{3.2}$$

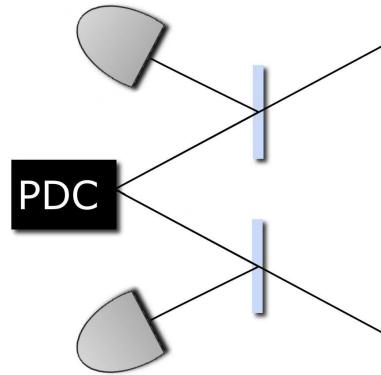


Figure 3.2: Perfect double photon subtraction setup

3.3.2 Perfect Double Photon Subtraction

After perfect Photon subtraction in the **two** branches of M photons:

$$\begin{aligned}
 |\psi'\rangle &= C \sum_{j=0}^{\infty} \binom{j+M}{j} \mu^j |j,j\rangle \\
 |\psi'_{M=1}\rangle &\sim |0,0\rangle + 2\lambda T^2 |1,1\rangle + 3\lambda^2 T^4 |2,2\rangle + 4\lambda^3 T^6 |3,3\rangle + \dots \\
 |\psi'_{M=2}\rangle &\sim |0,0\rangle + 3\lambda T^2 |1,1\rangle + 6\lambda^2 T^4 |2,2\rangle + 10\lambda^3 T^6 |3,3\rangle + \dots
 \end{aligned} \tag{3.3}$$

where $C = \sqrt{\frac{(1 - \mu^2)^{2M+1}}{\sum_{i=0}^M \binom{M}{i} \mu^{2i}}}$ and $\mu = \lambda T^2$.

Note that the increase in the entanglement is directly reflected in the coefficient transformation : $\lambda^n \longrightarrow (\lambda T^2)^n \binom{n}{M}$

The Logarithmic Negativity then becomes

$$E_N(|\psi'\rangle\langle\psi'|) = \log_2 \left(\frac{(1 + \mu)^{2M+1}}{\left(\sum_{i=0}^M \binom{M}{i} \mu^{2i} \right) (1 - \mu)} \right) \quad (3.4)$$

With the Von Neumann Entropy becoming:

$$S_{M=1}(|\psi'\rangle) = \sum_{j=0}^{\infty} C^2 \binom{j+M}{j}^2 \mu^{2j} \log_2 \left(C^2 \binom{j+M}{j}^2 \mu^{2j} \right) \quad (3.5)$$

$$\geq -\log_2 \left(\frac{(1 - \mu^2)^3}{\mu^2 + 1} \right) + 4\mu^2 (1 + \log_2(\mu)) \frac{\mu^2 + 2}{\mu^4 - 1} \quad (3.6)$$

(bound by (3.6) for M=1)

In the next page we plot the entanglement increase for a subtraction BS with transmittivity $T = 98\%$.

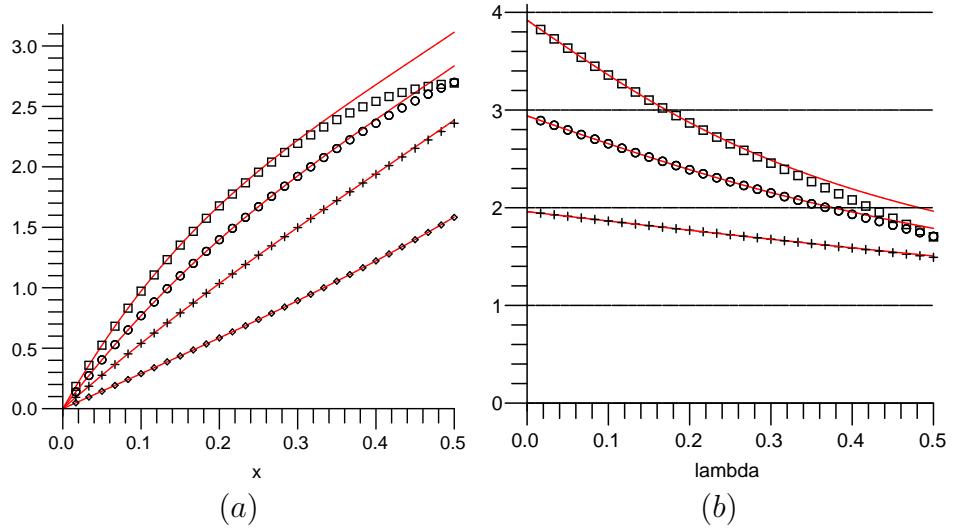


Figure 3.3: Logarithmic Negativity for the perfect subtraction of 1,2 and 3 photons vs. initial squeezing λ (or x .).

fig. 3.3 (a) shows from bottom to top the curves of the Logarithmic Negativity of the state before subtraction and after subtraction of $M=1$, $M=2$, $M=3$ photons. The points over the lines are simulations of these Log Negativities with Quantavo (truncating at 8 photons). They are plotted as a function of the squeezing parameter λ . Fig. 3.3 (b) shows the ratio $\frac{E_N(M=1)}{E_N(ini)}$, $\frac{E_N(M=2)}{E_N(ini)}$, $\frac{E_N(M=3)}{E_N(ini)}$ of these Logarithmic Negativities to find where the maximum increase can be achieved. We also compare it to the 8 photon Quantavo simulation.

Fig 3.4 shows: (c) From bottom to top: Von Neumann Entropy of the original state, analytic bound to the entropy of the subtracted state ($M=1$ photon) and numerical calculation of the entropy of the subtracted state. All as a function of λ . And (d), ratios of the Entropies. We must note that even though the highest increase in entanglement occurs at low λ -s, the magnitude is very small and could most likely not be measured. The best we can hope for is around a 3 fold increase of entanglement. Let us then address the problems of inefficient detectors, coupling losses and reasonable probabilities.

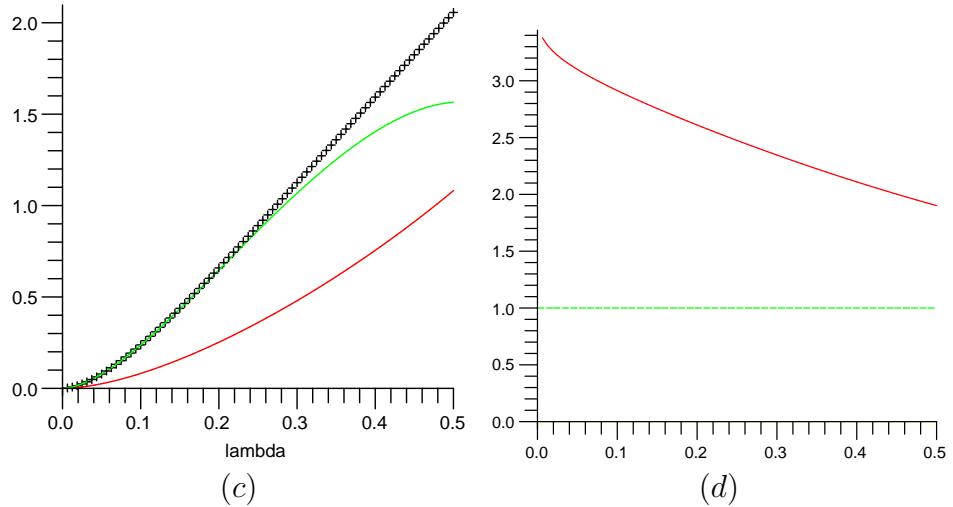


Figure 3.4: Entropy of Entanglement after subtraction of one photon in both branches of the parametric down conversion as shown in Fig. 3.3.

3.3.3 Inefficient fibre coupling and lossy imperfect detectors

To make a more realistic description I have explored various degrees of approximation and realism. I have also explored a variety of setups. All the following cases and combinations thereof have been studied:

- Using APDs or TMDs for subtraction.
- Recording one click, zero clicks, two or three clicks (in a single TMD), one and zero, etc
- modelling the BS as working perfectly in free space or as being in a fibre and therefore with an inefficient coupling from the crystal.
- Using ideal detectors, rough imperfect detector models or using a full characterization of both APDs and TMDs.

In more detail, if we want to couple the down converted photons to a single mode fibre, then many modes populated in the squeezed state will be filtered out. Since it is a passive linear optics element it will effectively make a mode transformation equivalent to a BS. Of course we have to make sure that this description of loss is appropriate. A

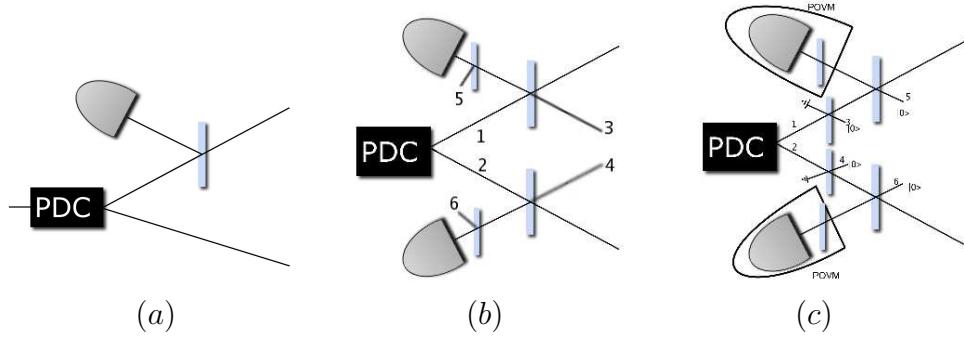


Figure 3.5: Different photon-subtraction setups. (a) Single subtraction in free space with perfect detectors. (b) Double subtraction in free space with lossy detectors. (c) Double subtraction in optical fibre with accurate model of the detector’s POVMs.

frequency dependent filter (therefore non-energy preserving and outside of linear optics) would introduce a different kind of loss. For the detection after the subtraction we used various models for the TMD and APD. More importantly due to the important role that their loss can play in mixing the subtracted state we performed a full characterization of them as discussed in chapter 4.

Describing the state which is the output of the circuit in figure (3.5-c) soon becomes difficult in the Fock basis. For instance, if we start with the truncated state $|00\rangle + \lambda|11\rangle + \lambda^2|22\rangle + \lambda^3|33\rangle$ then, in the output, the coefficient of $|00\rangle\langle 00|$ is a polynomial with 126 terms in T : transmittivity of the subtracting BS, η : efficiency of the fibre coupling, λ : squeezing parameter, α : inefficiency of the detector.

To describe this evolution we simply used the Maple Module **Quantavo** and evaluated the probabilities and Log Negativity with it. We have seen in the previous section that these simulations give good lower bounds for the squeezing range $0 < \lambda < 0.3$ which is our working range anyway.

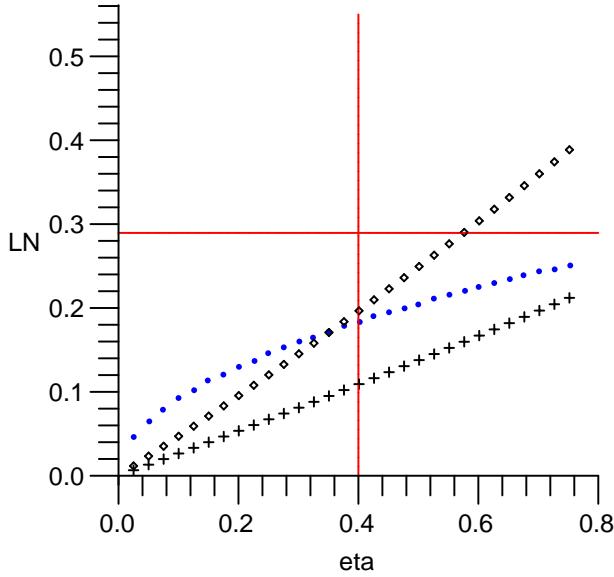


Figure 3.6: Logarithmic Negativity vs. coupling efficiency η (such that $\eta = 1/2$ is a 50/50 BS) for 3.5-c. The red horizontal line represents the entanglement of the original squeezed state with ($\lambda=0.1$). The vertical red line is a possible value of the coupling (40%). The lowest CROSSES curve is the exact Log-Negativity once the state is in the fibre. The DIAMONDS curve is the exact Log-Negativity after subtraction of one photon (the detector considered is the real detector described by the POVM provided by the tomography). The BLUE DOTS are the Log Negativity that a state with the same mean energy as the state in the fibre (before subtraction) would have, if it had the maximum achievable Log-Negativity.

Coupling to Fibre

An important lesson about coupling to fibres is that the initial entanglement will not increase for current efficiencies. The entanglement from the original state will be diminished by the filtering (and mixing) process the fibre introduces. Once in the fibre, the subtraction will however enhance this entanglement. A fair comparison would be to measure the mean energy a state has after coupling to the fibre deducing an upper bound to the entanglement. To obtain this upper bound consider $|fib\rangle$ to be the state in the fibre (prior to subtraction). If $\tilde{E}_{fib} = \langle fib|E|fib\rangle$ is its mean energy then one can define $|\psi_\epsilon\rangle = \sqrt{1 - \epsilon^2} \sum_{n=0}^{\infty} \epsilon^n |n, n\rangle$ such that $\tilde{E}_{fib} = \langle \psi_\epsilon|E|\psi_\epsilon\rangle$. This quantity is plotted in blue in fig. 3.7. It is precisely that quantity which could be compared with the lower bound on entanglement found after subtraction. Obtaining this lower bound will

be discussed further down (3.5).

3.4 Chosen Subtraction Setup

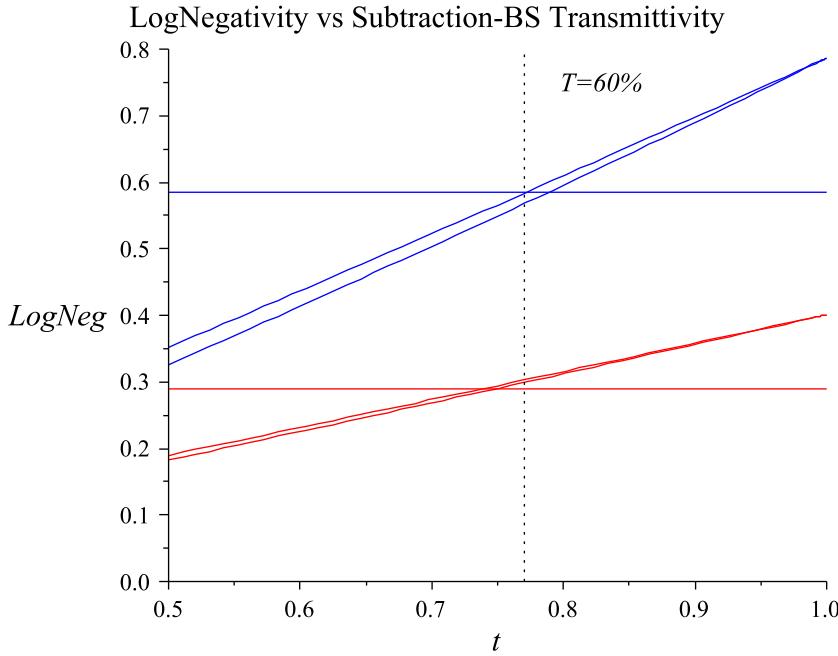


Figure 3.7: Logarithmic Negativity vs. the reflectivity t^2 of the subtracting BS. (single detection in one branch only). The blue curves show the Log-Negativity for initial squeezing of $\lambda = 0.2$. The horizontal blue one shows E_N for the squeezed state, while the upper one corresponds to a subtraction with detector efficiency of 90% and the lower one of 20%. The three red curves show respectively the Log-Negativity for initial squeezing of $\lambda = 0.1$. (Horizontal=initial, upper= 90% efficiency, lower= 20% efficiency).

In the near future it is our hope to be able to implement the setup we found to be the most advantageous: subtracting a single photon in free space. We would use a squeezed state with squeezing parameter near $\lambda = 0.2$. The fibre coupling to the APD detecting the subtracted photon has close to 30% efficiency, and the APD's efficiency close to 50%. Combined, this implies a detector with 85% loss. However, for a free space subtraction the efficiency of the detector or of the coupling to it will be largely irrelevant. That is, concerning entanglement increase but not probability. This can be interpreted

as follows: when a *click* occurs, it does not matter much if the implemented POVM is $\sum_{n=1}^{\infty} |n\rangle\langle n|$ or $\sum_{n=1}^{\infty} (1 - r^{2n})|n\rangle\langle n|$ (r^2 being the reflectivity of the interposed BS). Since the subtracting BS only diverted a small part of the state, and higher photon numbers have low amplitudes in the state, the mixing for any r^2 will be similar. Informally said our detector doesn't work very often, but when it works it does its job well enough. The efficiency will of course make itself apparent reducing the probabilities of a *click*. However, for a 250kHz pulsed laser, with an 80% transmitting BS and an 15% efficient detector we get of the order of 1300 successful subtractions per second. This rate is of course important for the next step: estimating the entanglement of the subtracted state.

3.5 Entanglement Quantification

In the continuous variable domain any reconstruction of the (infinite) density matrix will always be incomplete for obvious reasons. The so called *full reconstructions* performed with strong homodyne tomography are of course also limited by their finite phase space resolution and Hilbert space truncation [RHKL07, HMR06]. For these and other situations without full tomographic knowledge about the state a variety of tools have been developed [EBA07, GRW08, GRW07, AP06]. The spirit is the following: Given the statistics from a set of measurements on a state, what is the least entangled state consistent with this data. This contrasts with studying partial correlations or making a maximum likelihood estimate from incomplete data which can lead to an overestimate of the entanglement. Finding the least entangled state consistent with the measured statistics also called “quantitative entanglement witnesses” is therefore a very rigorous and sure way to assess the entanglement in a CV state providing a lower bound to it. The question can of course be recast as a simple optimization:

Given an entanglement measure E find the ρ which minimizes $E(\rho)$ subject to ρ being consistent with the measured data.

Of course this procedure can help us decide which observables to measure in order

to get the tightest lower bound for the entanglement generated. In more detail let us assume we measure certain probabilities $\{p_1, \dots, p_n\}$ for certain measurement operators M_1, \dots, M_n , meaning that

$$\langle M_i \rangle = \text{Tr} \{M_i \rho\} = p_i \quad (3.7)$$

For a bi-partite state, M_i is a witness if for all *separable states*

$$\text{tr}[M_i \rho] \geq 0, \quad (3.8)$$

and at least for a single entangled state ρ , one finds that [Ter00, HHH96]

$$\text{tr}[M_i \rho] < 0. \quad (3.9)$$

(which works because the separable states form a convex set). In our case we will be interested in using the Logarithmic negativity and therefore in solving the problem

$$\begin{aligned} E_{N,\min} &= \inf_{\rho} \quad \|\rho^{\Gamma}\|_1 \\ \text{subject to} \quad &\text{tr}[\rho M_i] = p_i, \quad \rho \geq 0 \quad \text{and} \quad \text{Tr} \{\rho = 1\} \end{aligned} \quad (3.10)$$

However, thanks to a property of the trace norm, $\|A\|_1 = \max_{\|X\|_{\infty}=1} [\text{Tr} \{X A\}]$ we can recast the minimization in the form

$$\begin{aligned} E_{N,\min} &= \inf_{\rho} \quad \max_{\|X\|_{\infty}=1} [\text{Tr} \{X \rho\}] \\ \text{subject to} \quad &\|X\|_{\infty} = 1, \quad \text{and} \\ &\text{tr}[\rho M_i] = p_i, \quad \rho \geq 0 \quad \text{and} \quad \text{Tr} \{\rho = 1\} \end{aligned} \quad (3.11)$$

Since the Log-Negativity is found maximizing over the X , any X observing $\|X\|_{\infty} = 1$

will provide a lower bound to it:

$$E_{N,\min} \geq \inf_{\rho} \quad \text{Tr } \{X \rho\} \quad (3.12)$$

$$\text{tr}[\rho M_i] = p_i, \quad \rho \geq 0 \text{ and } \text{Tr } \{\rho = 1\}$$

Now an interesting choice of a suitable X is a linear combination of the partial-transposed measured witnesses [EBA07],

$$X = \sum_{i=1}^n \alpha_i M_i^\Gamma + \alpha_{n+1} \mathbb{I}$$

with $\alpha_i \in \mathbb{R}$. This is so because,

$$\text{Tr } \{X \rho\} = \sum_{i=1}^n \alpha_i \text{Tr } \{M_i^\Gamma \rho^\Gamma\} + \alpha_{n+1} = \sum_{i=1}^n \alpha_i p_i + \alpha_{n+1}$$

and there is therefore nothing to minimize in \inf_{ρ} . Finding a good lower bound to $E_{N,\min}$ can then be translated to finding a set of real $\{\alpha_i\}$ such that:

$$\begin{aligned} \max & \quad \sum_{i=1}^n \alpha_i p_i + \alpha_{n+1}, \\ \text{subject to} & \quad -\mathbb{I} \leq X \leq \mathbb{I}, \\ & \quad X = \sum_{i=1}^n \alpha_i M_i^\Gamma + \alpha_{n+1} \mathbb{I}, \end{aligned} \quad (3.13)$$

which is a semi-definite-optimization problem (and therefore one that can be efficiently computed). This is then the tool we could use to characterize the entanglement in our subtracted state. Of course If we want to “prove rigorously” an increase in entanglement we should provide:

1. for the squeezed state before subtraction an upper bound on the Log-Negativity (or other Entanglement measure)
2. for the squeezed state after subtraction a lower bound on the Log-Negativity (or other Entanglement measure).

To obtain (1) all we need to do is measure the average photon number of the squeezed state. Indeed if our initial squeezed state is some unknown ρ_0 we can measure the expected value of the energy $\mu_0 = \text{Tr} \{E\rho_0\}$. Associated with this energy there is only one pure state which has maximum entanglement and that is the pure squeezed vacuum state $|Sq\rangle = \sqrt{1 - \lambda^2} \sum \lambda^n |n, n\rangle$. This state itself has $\langle Sq|E|Sq\rangle = \left(\frac{1+\lambda^2}{1-\lambda^2}\right) \frac{\hbar\omega}{2}$. Therefore, solving $\mu_0 = \left(\frac{1+\lambda^2}{1-\lambda^2}\right) \frac{\hbar\omega}{2}$ we find the maximum squeezing consistent with that energy and therefore the upper bound to the entanglement.

To obtain (2) we need to choose some appropriate POVMs, measure the state with them and from 3.13 find the lower bound. Of course there might not be enough entanglement to begin with in the mixed squeezed state due to various inefficiencies in which case the increase also would not be that dramatic and the bounds might not be far apart enough. However, given the simplicity of the method (measuring average photon number) and the possibility of such a robust result it should be the first approach to the problem. If this proves to be insufficient then one may have to do tomography of the state before subtraction together with some assumptions on the energy of the state. Of course it is impossible to do tomography beyond say 7 photons for which (in our case) detection of 7 photons would occur once or twice a day. Of course it is reasonable to make some assumptions about the decay of the probability amplitude beyond those Fock layers (and therefore about the Entanglement they might contain).

3.6 Characterization Strategy

As we have seen entanglement witnesses give us the advantage of not needing full tomography. This can make experiments faster, easier and result in rigorous outcomes. The advent of advanced photon number resolving detection has spurred interest in reconstructing directly the density operator of optical states. Beyond measuring quasi-probability distributions and reconstructing the density matrix there have been suggestions to measure directly the matrix elements [SV95]. That approach however requires superpositions of Fock states $|M\rangle, |N\rangle$ to characterize the elements $\rho_{N,M}$. Other more

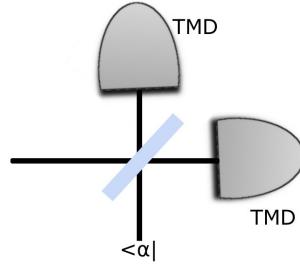


Figure 3.8: Weak-homodyning with a weak coherent local oscillator (providing the phase reference) and two time multiplexing detectors.

realistic proposals involve weak homodyning [PP02]. In this approach the state to be measured is mixed at a BS with a weak local oscillator. The photon number is measured in the two out-coming modes. Adapting the phase of the local oscillator and with the photon number measurements one can effectively reconstruct the $\rho_{N,M}$ elements in principle. Real TMDs, however won't implement exactly the $|n\rangle\langle n|$ POVM but some mixture of photon numbers. To know if the POVMs from such an imperfect detector are sufficient to build the appropriate entanglement witnesses we need an accurate description of them. This is thoroughly addressed in chapter 4.

Our findings reveal that the POVMs implemented by imperfect TMDs with a phase reference are sufficient to provide a tight bound within a few percent of the real Log-Negativity value. A setup as shown in fig. 3.9 should then represent a feasible experiment. This makes it a very promising avenue for non-Gaussian state preparation using LOCC as well as for CV distillation of entanglement with linear optics.

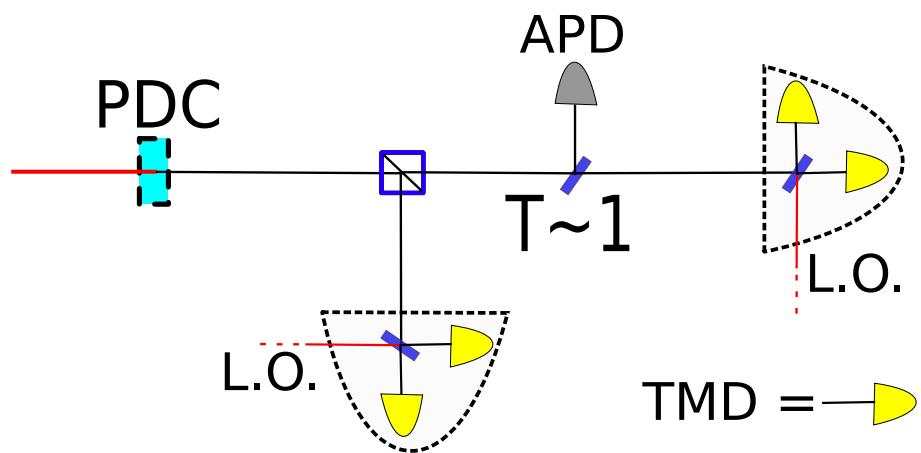


Figure 3.9: Subtraction setup with one subtraction (BS+APD) and the subsequent characterization. In this case we show on each side of the subtracted state weak-homodyning with a local oscillator and time multiplexing detectors).

4

Detector Tomography

Quantum measurement is our window to the microscopic quantum realm. Conversely, preparing a particular quantum state amounts to performing filtering measurements [Hel76]. Surprisingly – in the light of the central status of measurement in quantum mechanics – the characterisation of detectors has typically been based on partial calibrations or elaborate models invoking several assumptions. There exist recipes to create optical measurement apparatuses based on a few building blocks such as beam splitters, photon-number detectors and local oscillators [PP02, Pre05]. However there is no recipe for arranging matter (and eventually an apparatus) that implements those building blocks (and therefore a specific observable). In this chapter a direct and full characterisation tries to assume as little as possible about the detector presenting a substantially different strategy.

In our particular case, when implementing a photon subtraction and using entanglement witnesses, the role of well known detectors is crucial. By subtracting a number of photons we are preparing a state, and therefore characterising the detector tells us what state we are preparing. Furthermore, the use of entanglement witnesses as presented in chapter 3 can only be accurate if the operators describing those witnesses are correct.

Here we present the first experimental realisation of quantum detector tomography [LSS99, Fcv01, DMP04]. This result completes the triad of experimental state [VR89, SBRF93, BRWKan99], process[CN96, PCZ97, DM98, ABJ⁺03], and detector tomography [LFCR⁺08]. We therefore identify the physical positive-operator-valued measures closest to the experimental data without any assumptions on the functioning of the detectors. This is done for an avalanche photodiode and a photon number resolving detector able to detect up to 8 photons[ASS⁺03a]. This detector characterisation opens up more flexible and complex ways of detecting quantum states and accurately preparing non-classical light.

4.1 Introduction

The quantum properties of nature reveal themselves only to carefully designed measurement techniques[KJRPG⁺07, HBB⁺07]. In addition, most quantum information applications both computational and cryptographic, rely on a certain knowledge of the measurement apparatuses involved [RB02, Nie03]. But most importantly the assumption of a fully characterised detector completely underlies both quantum state tomography (QST) and quantum process tomography (QPT). In QST a given number of measurements on many copies of an unknown state reveal its density operator [VR89, SBRF93, BRWKan99]. Characterising the operators that govern an evolution or a channel – QPT – amounts to acting the process on a set of input states, and subsequently fully characterising the output states [CN96, PCZ97, DM98, ABJ⁺03].

In addition as quantum technology makes striking advances, detectors are becoming more complex calling for a black box approach to their characterisation. Photo-detection has seen the advent of single-carbon-nanotube detectors [FMM⁺03], charge integration photon detectors (CIPD) [SWM⁺04], Visible Light Photon Counters (VLPC) [KTYH99], quantum dot arrays [SOF⁺00], superconducting edge and picosecond sensors[MNMS03, GOC⁺01] or time multiplexing detectors based on commercial Si-APDs [ASS⁺03a, ASS⁺03b]. Certainly understanding in full detail the noise, loss and coherence charac-

teristics of these technologies is not trivial. Detector tomography is an answer to those challenges. With ever increasing non-classical states and channels studied and prepared with ever more elaborate detectors, detector tomography imposes itself. State and detector tomography evidently exhibit a dual role: Either the input is well-known and the detector is to be characterised, or the detector is well-known and the state is tomographically reconstructed. Building upon previous theoretical descriptions of detector tomography [LSS99, Fcv01, DMP04] and using methods of convex optimisation[BV04] we develop a framework in which this reconstruction can be carried out by means of efficient and simple numerics.

Any detector in quantum mechanics is described by a *positive operator valued measure* (POVM) $\{\pi_n\}$ where the outcomes are labelled by n . When these operators are orthogonal this generalisation simplifies to the projective measurement case, $\pi_n = |\psi_n\rangle\langle\psi_n|$ of which a perfect photon counter with $\pi_n = |n\rangle\langle n|$ is an example. For these operators to describe a physical measurement apparatus, they must be positive semi-definite, $\pi_n \geq 0$, and $\sum_n \pi_n = \mathbb{I}$, ensuring positive probabilities that add up to one. Given an input ρ , the probability p_n of obtaining output n is then

$$p_{\rho,n} = \text{tr}[\rho \pi_n]. \quad (4.1)$$

To recover the POVM elements $\{\pi_n\}$ from the measured statistics $p_{n,\rho}$ the probe states or input states must be carefully chosen. Indeed the set $\{\rho\}$ must be *tomographically complete*: the operators $\{\rho\}$ must form a basis for the operator space of π_n . When doing state tomography, one must perform a set of measurements $\{\pi_n\}$ spanning the space of the density operator to be reconstructed. If the state lives in N dimensions, then this space will have $N^2 - 1$ parameters (where one is subtracted to account for the normalisation). Conversely, for detector tomography, the reference states $\{\rho\}$ need to span the space of the POVM set. A spanning set will necessarily have N^2 elements or more in it, depending on how many outcomes the measurement has.

In principle this is sufficient to calculate the direct inversion of Eq. (4.1). However an experimental realisation carries additional requirements. On the one hand the probe states should be previously characterised, and large numbers of them should be easily generated. Coherent states are ideal candidates since a laser can generate them directly and we can create a tomographically complete set by transforming their amplitude through attenuation (for example with a beam splitter). Using input states $\{|\alpha\rangle\langle\alpha|\}$ one can then reconstruct the Q -function of the detector [LSS99] which is simply proportional to the measured statistics,

$$p_{n,\alpha} = \frac{1}{\pi^2} \langle \alpha | \pi_n | \alpha \rangle = \frac{1}{\pi} Q_n(\alpha). \quad (4.2)$$

Since $Q_n(\alpha)$ of each POVM contains the same information as the element π_n itself, predictions of the detection probabilities for arbitrary input states can then be calculated directly from the Q -function. However experimental errors and statistical fluctuations can cause a simple fit to the Q -function to be consistent with unphysical POVM elements. Due to this and because our detectors extract photon number information, we ultimately seek a physical and natural representation in the photon-number basis, i.e. the POVM elements.

4.1.1 The Experiment

We now turn to the description of the experimental realisation, shown in Fig. 4.1. The idea is simply to send coherent states with different amplitudes to the detectors and record all corresponding outcomes. To do so the coherent states were measured with a NIST calibrated power-meter. Part of the beam going to the power-meter was split and then attenuated with Neutral Density filters before entering the coupling fibre and the detector. The attenuation introduced by the different elements (BS, ND filters) was measured and the value of $|\alpha^2|$ derived.

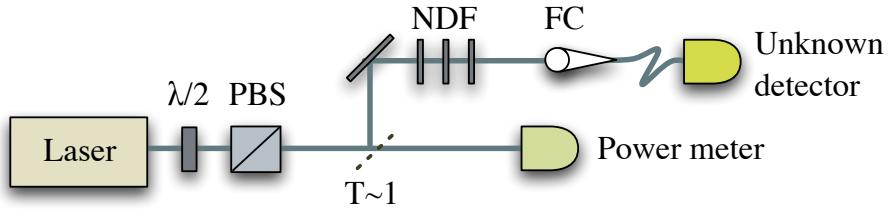


Figure 4.1: The experimental setup. A half-waveplate ($\lambda/2$) and Glan-Thompson polariser (PBS) are used to vary the amplitude of the probe coherent state, which is subsequently attenuated by Neutral Density Filters (NDF) and coupled into a fibre (FC).

4.2 The Detectors

APD

The first detector was a commercial single-photon counting module based on a silicon avalanche photodiode (APD). It has two detection outcomes, either outputting an electronic pulse (1-click) or not (0-clicks). A loss-free perfect version of it would implement the Kraus operators $\{|0\rangle\langle 0|, \mathbb{I} - |0\rangle\langle 0|\}$, distinguishing between the presence or absence of photons. However some photons are absorbed without triggering a pulse. This loss can be modelled placing a BS in front of the perfect detector [U.L03]. The POVMs describing a detector with a BS of transmittivity η can then be written as,

$$\text{NO CLICK} : \pi_0 = \sum_{n=0}^{\infty} (1 - \eta)^n |n\rangle \langle n|, \quad (4.3)$$

$$\text{CLICK} : \pi_1 = \mathbb{I} - \sum_{n=0}^{\infty} (1 - \eta)^n |n\rangle \langle n|. \quad (4.4)$$

disregarding after-pulsing or dark counts [CRS07]. Having only two outcomes, this detector cannot distinguish the number of photons present.

TMD

The second detector on the other hand has certain photon-number resolution. It obtains this resolution splitting the incoming pulse into many spatially or temporally separate bins, making unlikely the presence of more than one photon per bin. All the time bins are then detected with two APDs. Photon-number resolution results by summing the number of 1-click outcomes from all the bins. This *time-multiplexed detector* (TMD) is not commercially available but was constructed by the Ultrafast Group in Oxford [ASS⁺03a]. It had eight bins in total (four time bins in each of two output fibres) and thus nine outcomes – from zero to eight clicks. The added complexity and greater number of outcomes made detector tomography more interesting on this particular detector.

The theoretical description of this detector is a bit more involved since there is what we call the “binning problem”. Indeed, in addition to loss there is a certain probability that all photons will end up in a single time bin, or more generally that k incoming photons will result in less than k clicks. To account for the details describing these probabilities we use a recursive relation [Ple07]. Our goal is to describe the probability distribution:

$P^N(j/k)$: Probability of having j -Clicks given that there were k incident photons and that the detector has N-bins (or modes).

We will show how to calculate $P^2(j/k)$, $P^4(j/k)$ and then how to go from $P^N(j/k)$ to $P^{2N}(j/k)$. For a two-bin loopy detector as shown in Fig. 4.2 the relation is quite simple. If we consider a BS with transmittivity T and reflectivity R , then:

- $P^2(j, 0) = \delta_{j,0}$ (if no photons are present we will only register zero clicks).
- $P^2(1, k) = T^k + R^k$ (with probability T^k , k photons end up in the lower bin and the same holds for the upper bin with R^k . The probability of a single click is the sum of these independent probabilities).

- $P^2(2, k) = 1 - T^k + R^k$ (if $k \neq 0$ then only two events may happen: one click or two. This complementary event has therefore $P = 1 -$ (Probability of 1 click).)

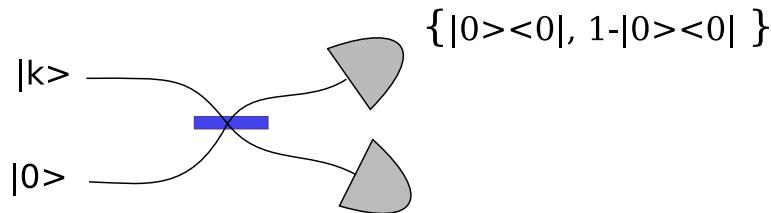


Figure 4.2: Diagram of a simplified 2 bin multiplexing detector.

In the case of a 4-bin detector shown in fig. 4.3, k incoming photons are distributed to two 2-bin detectors according to a binomial distribution. Now let us evaluate the

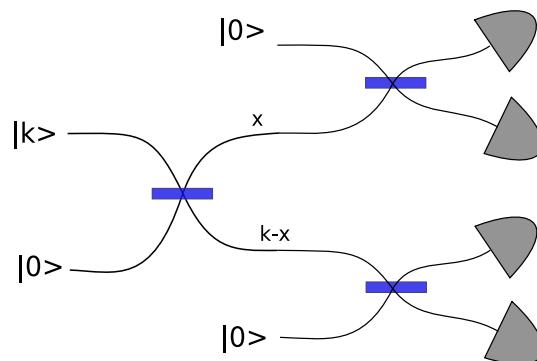


Figure 4.3: Diagram of a simplified 4 bin multiplexing detector. The first beam splitter distributes k photons according to a binomial distribution between the two 2-bin loopy detectors of the second stage.

probability for the upper 2-bin detector to register s counts if x photons entered while registering m clicks in the lower 2-bin detector if $k - x$ entered the lower port. This should be $P^2(s, x)P^2(m, k - x)$ weighted by the probability that x photons enter the upper branch and $k - x$ the lower one: $\binom{k}{x}T^{k-x}R^x$.

Now the probability that j counts are found overall is found summing the weighted probability over all possible ways that the detectors can find j counts (i.e. $m + s = j$)

and summing over all possible ways of distributing k photons:

$$P^4(j/k) = \sum_{x=0}^k \sum_{m+s=j} \binom{k}{x} T^{k-x} R^x P^2(s, x) P^2(m, k-x). \quad (4.5)$$

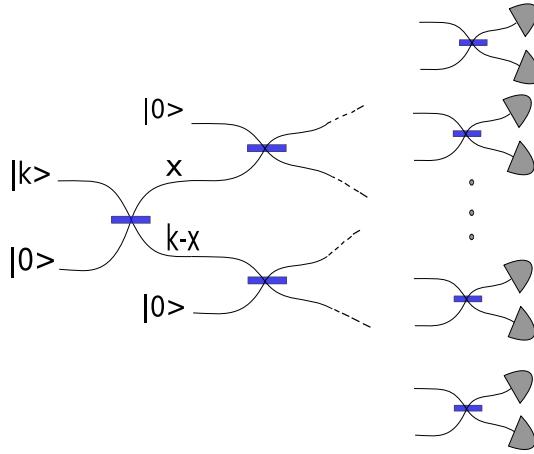


Figure 4.4: Diagram of a $2N$ bin multiplexing detector. The first beam splitter distributes k photons according to a binomial distribution between the two next N bin stages.

We can extend the same argument to $2N$. Imagine we know $P^N(j/k)$. Now, for that detector to become a $2N$ -bin detector all we need is to couple two of them to a beam splitter which will distribute the k photons as described above and as shown in Fig. 4.4. In that same fashion we can then define the recursive relation [Ple07]:

$$P^{2N}(j/k) = \sum_{x=0}^k \sum_{m+s=j} \binom{k}{x} T^{k-x} R^x P^N(s, x) P^N(m, k-x). \quad (4.6)$$

Based on this recursion and once we determine all T and R , we can write a simple program to generate the corresponding theoretical POVMs. For example a 5-outcome

detector would have a POVM which can be captured in the following matrix:

$$B = \begin{bmatrix} & \pi_0 & \pi_1 & \pi_2 & \pi_3 & \pi_4 & \pi_5 \\ |0\rangle\langle 0| & 1 & 0 & 0 & 0 & 0 & 0 \\ |1\rangle\langle 1| & 0 & 1 & 0 & 0 & 0 & 0 \\ |2\rangle\langle 2| & 0 & 0.12828 & 0.87172 & 0 & 0 & 0 \\ |3\rangle\langle 3| & 0 & 0.016857 & 0.33428 & 0.64886 & 0 & 0 \\ |4\rangle\langle 4| & 0 & 0.0022626 & 0.10096 & 0.49583 & 0.40094 & 0 \\ |5\rangle\langle 5| & 0 & 0.00030921 & 0.0283 & 0.26519 & 0.50875 & 0.19744 \\ |6\rangle\langle 6| & 0 & 0.000042885 & 0.0077213 & 0.12337 & 0.42183 & 0.44704 \\ |7\rangle\langle 7| & 0 & 0.000006018 & 0.002089 & 0.05363 & 0.29105 & 0.65323 \\ |8\rangle\langle 8| & 0 & 0.00000085224 & 0.00056503 & 0.022492 & 0.18197 & 0.79497 \end{bmatrix}$$

where $B_{k,j} = P^5(j/k)$ and j runs from $\{0 \rightarrow 5\}$ and k from $\{0 \rightarrow 8\}$. for example the 5-click event has a POVM element, $\pi_5 \simeq 0.2|5\rangle\langle 5| + 0.4|6\rangle\langle 6| + 0.6|7\rangle\langle 7| + 0.8|8\rangle\langle 8|$, etc. More generally, the measured statistics are related to the incoming photons by

$$p_j = \sum_k P^N(j/k) \rho_k$$

where p_j is the probability of detecting j counts and ρ_k the probability that k photons arrived to the TMD. This relation can be cast into matrix form becoming $\mathbf{p} = \mathbf{B} \cdot \boldsymbol{\rho}$ [ASS⁺03b].

Loss: TMD detectors have various sources of loss (meaning that photons are absorbed before triggering a detection event). The major sources of loss are the coupling to the fibres, the absorption and scattering in the delay fibres and the non-unit efficiency of the detectors [ASW06]. A full description of the effect of losses is certainly complex, since loss occurs at many stages of the detector. One would have to include a BS before the detector (fibre coupling), a BS at each stage of fibre and a BS in front of each APD, altering Eq. (4.6) accordingly. Instead we chose to give an effective description

which describes loss with a single BS in front of the detector. Disregarding the ‘binning problem’, a BS in front of a perfect number detector with $\pi_n = |n\rangle\langle n|$ would result in a relationship between statistics and incoming photons of the form,

$$\mathbf{p} = \mathbf{L} \cdot \rho$$

This time with $L_{k',k} = \binom{k}{k'} \eta^{k'} (1-\eta)^{k-k'}$ being the binomial distribution accounting for loss since $L_{k',k} = 0$ for all $k < k'$. Now combining both descriptions, we can decouple the loss from the binning, putting a BS coupled to the environment before the N-bin TMD resulting in:

$$\mathbf{p} = \mathbf{B} \cdot \mathbf{L} \cdot \rho. \quad (4.7)$$

This relationship expresses how the incoming photons experience loss and then are distributed among the available modes. This is the model we will use to describe the TMD sketched in Fig. 4.5. The transmittivities of the inner BS were measured independently, and from them we obtained the corresponding ‘binning matrix’ \mathbf{B} . Overall loss was estimated from the tomography data to achieve the best fit to the POVMs.

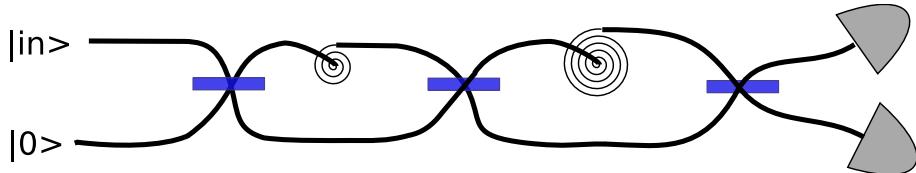


Figure 4.5: Diagram of a simplified 8 bin time multiplexing detector (TMD). The spirals represent a delay in the optical fibre.

4.3 Reconstruction

We now turn to the tomographic reconstruction. Since we adopt a ‘black box’ approach we need not assume any of the properties studied in the previous section. Only the accessible parts of the ‘black box’ will condition the description of our detector, i.e.

number of outcomes or control of phase. For both detectors we first allowed the phase of α to drift observing no variation in the outcome frequencies, as expected from a detector without a phase-reference. This simplifies the experimental procedure, allowing us to solely control the magnitude of α (as was done for tomography of a single photon [LHA⁺01]). A detector with no observed phase dependence will be described by POVM elements diagonal in the number basis,

$$\pi_n = \sum_{k=0}^{\infty} \theta_k^{(n)} |k\rangle\langle k|, \quad (4.8)$$

hence simplifying the reconstruction of π_n .

For a diagonal π_n , measuring the statistics for D different values of $\alpha, \alpha_1 \dots \alpha_D$, and truncating the number states at a sufficiently large M , we can rewrite Eq. 4.2 as,

$$\tilde{P} = F \Pi. \quad (4.9)$$

Where we have taken:

$$\begin{aligned} P^{(n)}(\alpha) &= \langle \alpha | \hat{\Pi}^{(n)} | \alpha \rangle \\ &= \langle \alpha | \left(\sum_k \theta_k^{(n)} |k\rangle\langle k| \right) | \alpha \rangle \\ &= e^{-|\alpha|^2} \sum_k \frac{|\alpha|^{2k}}{k!} \theta_k^{(n)} \\ &= \sum_k F_k(\alpha) \theta_k^{(n)} \\ &= \hat{F}(\alpha) \hat{\Pi}^{(n)} \end{aligned}$$

For an N outcome detector, the matrices will have dimensions $P_{D \times N}, F_{D \times M}$ and $\Pi_{M \times N}$. In addition $F_{i,k} = \frac{|\alpha_i|^{2k} \exp(-|\alpha_i|^2)}{k!}$ can easily be rewritten when the input state is a mixed state. This was done indeed to account for the laser's technical noise (as we will see in 4.4.2) but gave similar results. For such a detector, the physical POVM consistent with

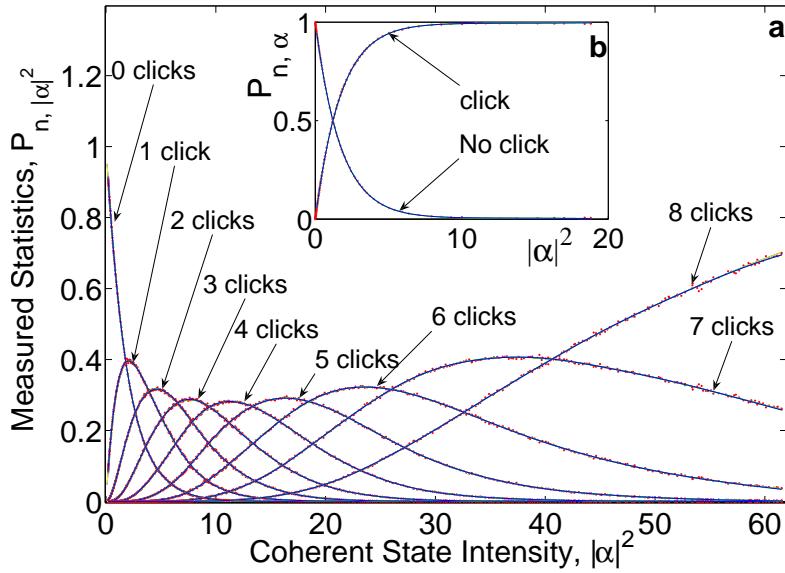


Figure 4.6: The measured probabilities for different photon numbers are shown (red dots) as a function of $|\alpha|^2 = \langle n \rangle$. The main plot shows the time multiplexed detector (TMD) with 9 time-bins and the upper right corner shows the measured probability distribution for the avalanche photodiode (APD). The statistical error vertically is too small to be seen and the jitter of $|\alpha|^2$ was estimated to be 2% of its value. An additional 5% systematic error in the calibration of the power meter is present but can be absorbed as loss. From the reconstructed POVM elements $\{\pi_n\}$ we generate the corresponding probability distributions $\text{Tr}[\rho_\alpha^{(\text{in})}\pi_n]$ (blue curves). These are generated for pure $|\alpha\rangle\langle\alpha|$ or mixed $\rho_\alpha^{(\text{in})}$ and for π_n reconstructed with the filter function in Eq. or without it. For all these options, the probability distributions (blue lines) are so similar that they are indistinguishable on this scale.

the data can be estimated through the following optimisation problem:

$$\begin{aligned} \min & \left\{ \|\tilde{P} - F\Pi\|_2 + g(\Pi) \right\}, \\ \text{subject to } & \pi_n \geq 0, \quad \sum_{n=1}^N \pi_n = 1, \end{aligned} \quad (4.10)$$

where the 2-norm of a matrix A is defined as $\|A\|_2 = (\sum_{i,j} |A_{i,j}|^2)^{1/2}$ and ensures it's a convex quadratic problem. Note that we also allow for convex quadratic filter functions g , related to the conditioning of the problem, which must not depend on the type of detector. For example, no symmetry or knowledge of the typical POVM structures in photo-detection can be assumed. If any, only general regularization functions that would work for any POVM should be chosen. Since this is a convex quadratic optimisation problem, and hence also a semi-definite problem (SDP) which can be efficiently solved numerically [BV04]. Moreover, in this case, there exists a dual optimisation problem whose solution coincides with the original problem. Thus, the dual problem provides a certificate of optimality since it provides a lower bound to the primal problem.

Care has to be taken that the optimisation problem is well conditioned in order to find the true POVM of the detector. In finding the number basis representation we are deconvolving a coherent state from our statistics which is intrinsically badly conditioned. Similar issues of conditioning have been discussed in the context of state and process tomography, see e.g. Refs. [BHP03, JcvFcvH03]. Due to a large ratio between the largest and smallest singular values of the matrices defining the quadratic problem, small fluctuations in the probability distribution can result in large variations for the reconstructed POVM. This can result in operators that approximate really well the outcome statistics and yet do not exhibit a smooth distribution in photon-number. We will discuss how to treat this problem in 4.4.3.

The measured probabilities for each outcome as a function of $|\alpha|^2$ are displayed in Fig. 4.6. The probability distributions (equivalent modulo $1/\pi$ to the Q-function of the

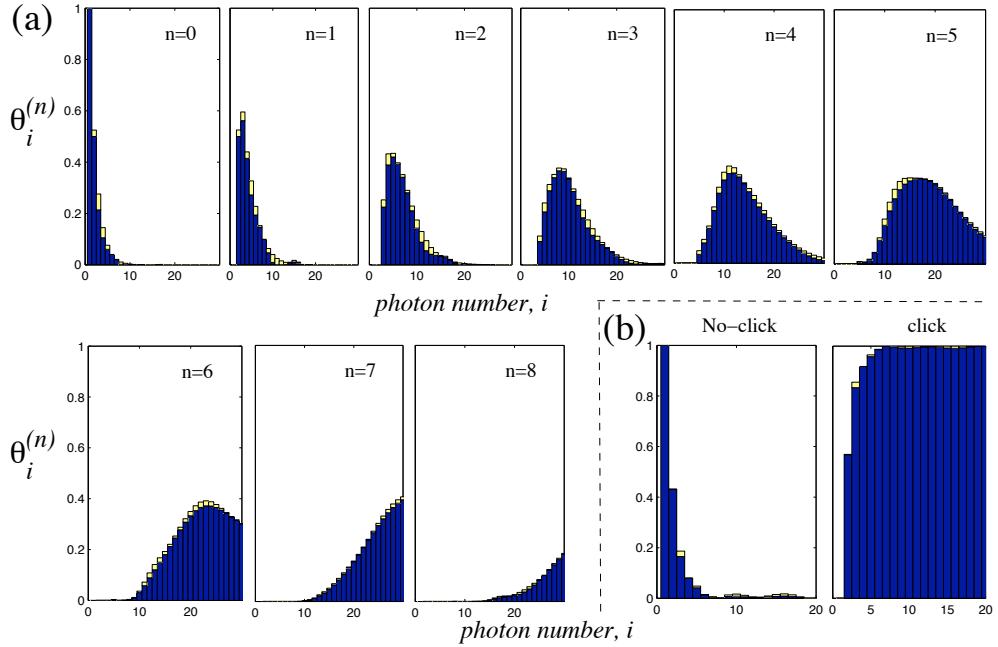


Figure 4.7: Reconstructed POVMs for (a) the photon-number resolving TMD and (b) the APD “yes/no” detector. TMD POVMs were obtained up to element $|60\rangle\langle 60|$ (therefore $M=60$), but are shown up to $|30\rangle\langle 30|$ for display purposes. APD POVMs are shown in full. Stacked on top of each $\theta_i^{(n)}$ where n is the number of clicks we show $|\theta_i^{(rec)} - \theta_i^{(teo)}|$ in yellow. “rec” stands for reconstructed and (teo) is the theoretical POVM expected from (a) a TMD modelled with 3 beam splitters of measured reflectivities and 52.1% overall loss (b) a theoretical APD with 43.2% loss respectively.

detector) show smooth profiles and distinct photon number ranges of sensitivity for increasing number of *clicks* in the detector. Fig. 4.7 shows the POVMs that result from the Eq. 4.10 optimisation which we will discuss later. A first remarkable property is that π_n , being the POVM for n clicks, shows zero amplitude for detecting less than n photons. That is, the detector shows essentially no dark counts. This however was not assumed anywhere and is purely the result of the optimization. This sharp feature gives the detector its discriminatory power where n clicks means at least n photons in the input pulse.

To assess the performance of our method we compare it to the model described in the previous section. This time however, the BS used in the model are not 50/50 but its reflectivities ($R=[0.5018, 0.5060, 0.4192]$) were measured experimentally. The yellow bars in Fig. 4.7, show the absolute value of the difference between the theoretical and the reconstructed POVM elements. The magnitude

$$\Delta_\theta^{(n,i)} = |\theta_i^{n,\text{teo}} - \theta_i^{n,\text{rec}}|$$

is shown stacked on top of each coefficient of the POVM elements where *teo* stands for theoretical and *rec* for reconstructed. The small yellow bars reveal a good agreement with the model. We also calculate the fidelity finding that

$$F = \text{Tr} \left\{ \left(\sqrt{\pi_n^{\text{teo}}} \pi_n^{\text{rec}} \sqrt{\pi_n^{\text{teo}}} \right)^{\frac{1}{2}} \right\}^2 \geq 98.7\%$$

holds for all n , indicating excellent agreement between the two.

In addition, one can reconstruct a probability distribution: from the found POVMs to fit the data. The reconstruction is plotted as blue curves in Fig. 4.6. It is the equivalent of the Q-function had our probe states $|\alpha\rangle\langle\alpha|$ been strictly pure. In fact, although formally distinct, the probability distribution associated with the reconstructed POVM using mixed or pure states are practically indistinguishable and are plotted together in

Fig. 4.6 for comparison.

4.3.1 Detector Wigner Functions

An alternative representation of the detectors which can give us more insight about their structure comes from the quasi-probability distributions such as the Wigner Function [Leo97, Sch01]. Since the POVM elements π_n are self adjoint positive-semi-definite operators the Wigner function W_n can be calculated in the standard way from the POVM element π_n :

$$W_n(x, p) = \frac{1}{\pi\hbar} \int_{-\infty}^{\infty} dy \langle x - y | \pi_n | x + y \rangle e^{2ipy/\hbar}. \quad (4.11)$$

However, since the POVMs do not have trace one, this detector Wigner function will not be normalised,

$$\int_{-\infty}^{\infty} dx \int_{-\infty}^{\infty} dp W_n(x, p) < 1. \quad (4.12)$$

We should note that the marginals cannot be interpreted as probability distributions but we can still use W_n to calculate probabilities according to:

$$P_{n,\rho} = \text{Tr}(\hat{\rho}\pi_n) = \int_{-\infty}^{\infty} dx \int_{-\infty}^{\infty} dp W_\rho(x, p)W_n(x, p). \quad (4.13)$$

We plot the Wigner functions of the ‘one click’ Operator for the TMD and the ‘click’ operator for the APD in Fig. 4.8. It is remarkable to see the similarities of the ‘one click’ POVM with its state counterpart: the single photon Wigner function. The APD however displays the contributions from higher photon numbers but still exhibits negative values at the origin. Since none of the detectors have phase sensitivity their Wigner functions are rotationally symmetric around a vertical axis through the origin.

In Figures 4.9 and 4.10 we display a cut of the TMD Wigner function for the following POVM elements: $\{\pi_0, \pi_1, \pi_2, \pi_3, \pi_4, \pi_5\}$. The interesting feature about the plot is the comparison with the theoretical TMD Wigner functions from section 4.2 and the effect

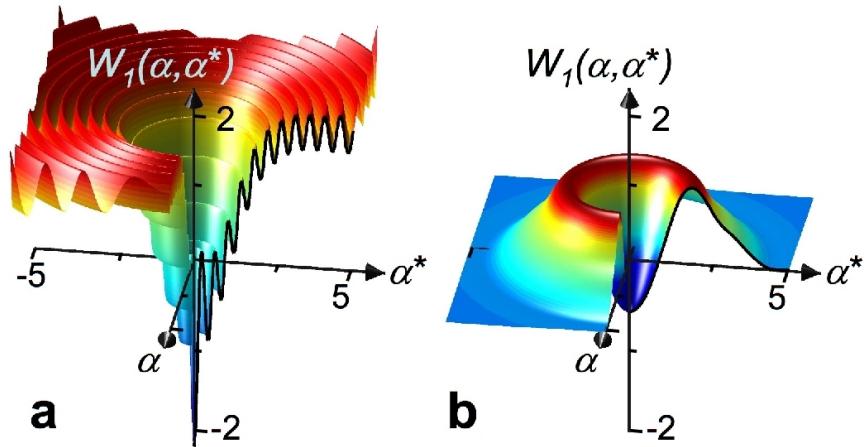


Figure 4.8: Wigner quasi-probability distributions for the ‘one click’ event on a TMD (a) and ‘click event’ on the APD (b) [LFCR⁺08]. From the diagonal POVM elements of the detectors one can easily generate the Wigner representation. An APD detector is sometimes regarded as a ‘single photon detector’ but here we can see the striking difference between the Wigner function of a ‘one click’ event on a TMD and the Wigner function of the ‘click event’ on the APD. Indeed the POVM element corresponding to (a) has a fidelity of 99% with a single photon having experienced loss on a 48% beam splitter. It is worth noting that both functions have negative values near the origin, emphasizing the absence of a classical analogue.

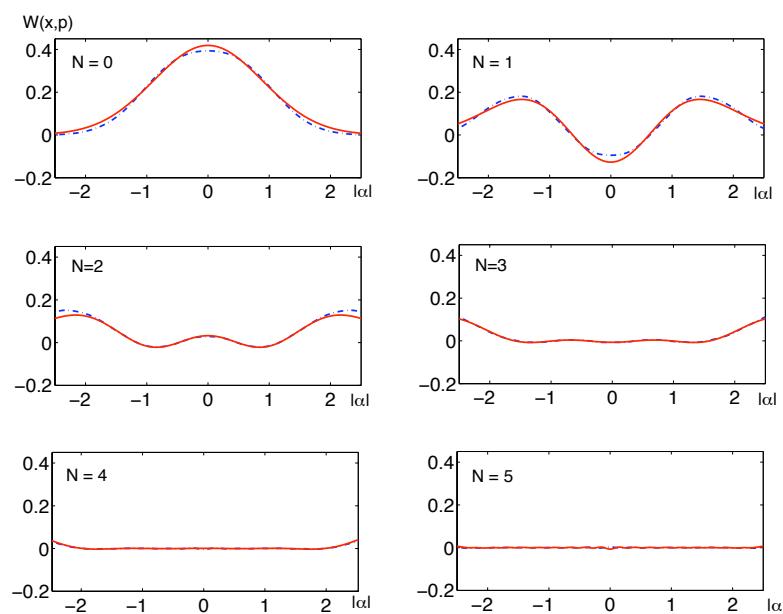


Figure 4.9: Wigner function of the first POVM elements of the TMD. Since the detectors have no phase reference, their Wigner functions are rotationally symmetric with respect to their center and a cut contains all the information. The dotted blue curve represents the Wigner function of the reconstructed POVMs from 0 to 5 clicks. In red we can see the theoretical Wigner function for a theoretical TMD with 52% loss.

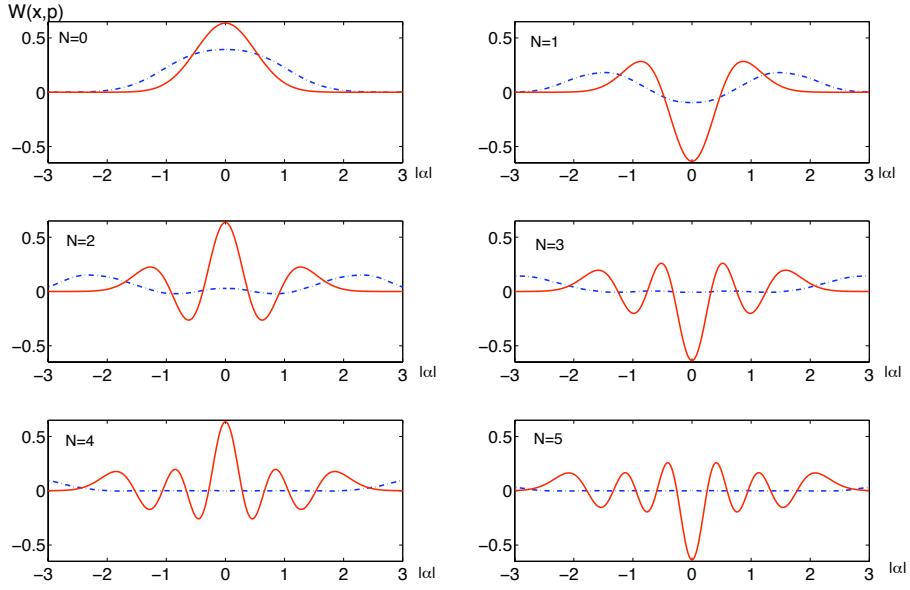


Figure 4.10: Wigner function of the first POVM elements of the TMD. Since the detectors have no phase reference, their Wigner functions are rotationally symmetric with respect to their center and a cut contains all the information. The dotted blue curve represents the Wigner function of the reconstructed POVMs from 0 to 5 clicks. In red we can see the theoretical Wigner function for a theoretical TMD without loss. Paying attention to the scale we observe how dramatic the effect of loss is at damping the ripples in the Wigner function.

of loss. Indeed comparing a theoretical loss-less TMD with the measured one we see how the amplitude of the Wigner function decreases rapidly for higher photon numbers. On the other hand, comparison with the lossy theoretical model reveals a good agreement.

4.4 Ill Conditioning and Regularisation

One of the main problems encountered in the tomographic characterisation of the detectors has to do with the numerical reconstruction. Such problems are common in tomography [BHP03, JcvFcvH03]. Consider for example the transformations involved in the inverse Radon transform and their inherent instabilities . Note also how going from

the Q-function to the P-function is not always well defined [Sch01]. Multiple tools exist to bridge the link between homodyne tomography and the density matrix description [LR05]. One of them involves the use of pattern functions [DLP95, LPD95, Wün97]. That is, finding some functions $G_k(\alpha)$ such that,

$$\int Q^{(n)}(\alpha) G_k(\alpha) = \theta_k^{(n)}.$$

However, finding the appropriate $G_k(\alpha)$ involves the irregular wave functions [LMK⁺96] and proving them to be appropriate is as hard as estimating the error. The use of maximum likelihood has also been explored and particularly for detector tomography [DMP04, Fcv01]. However, the speed of the convergence is not generally guaranteed to be high, becoming exponential for certain problems. Our approach, following the spirit of maximum-likelihood, translates the problem into a quadratic optimisation one allowing for efficient semi-definite programming (SDP) (cf. Eq. 4.10). We discuss here the details, approximations and filters that lead to our solution of the problem.

4.4.1 Truncation

The data was measured up to $|\alpha|^2 = 150$ but was truncated at lower values to avoid noisy behaviour and the emergence of new regimes requiring a larger POVM space. This was hinted at (but not specifically proven) by the detector's lack of saturation. That is, The probability of having 8 clicks (the maximum) for $|\alpha|^2 = 150$ was not close enough to 1. And the probability of having 7 clicks was still about 0.2 away from zero for such high average photon number. We found $|\alpha|^2 < 60$ to be a fair truncation since it is sufficient to characterise a 9 outcome detector up to reasonably high Fock layers and we wanted to avoid the possibility of the loopy detector + electronics entering a different regime. Among the plausible explanations for non saturation we count the possibility of the FPGA (field-programmable gate array) failing to count all 8 present pulses when 8 or more occur. This could be due to incorrect time-gating but was not yet been clarified.

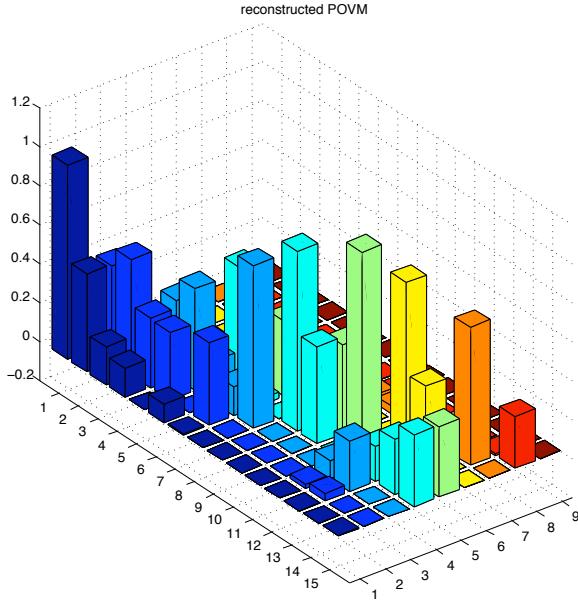


Figure 4.11: POVM reconstruction, using only minimisation from eq. 4.16. Dark blue: $\pi_0 \text{ clicks} = \sum_{i=0}^{15} \theta_i^{(0)} |i\rangle\langle i|$, lighter blue, $\pi_1 \text{ click} = \sum_{i=0}^{15} \theta_i^{(1)} |i\rangle\langle i|$, etc..

4.4.2 Pure vs. Mixed

The Q-function of our detector (directly measured) is proportional to,

$$p_{n,\alpha} = \text{tr}[|\alpha\rangle\langle\alpha| \pi^{(n)}]. \quad (4.14)$$

Since we have no phase reference, the POVM elements will be diagonal $\pi^{(n)} = \sum_k \theta_k^{(n)} |k\rangle\langle k|$.

Measuring the statistics for D different values of $\alpha, \alpha_1 \dots \alpha_D$, and truncating the number states at a sufficiently large M , we can rewrite Eq. 4.14 as

$$\tilde{P} = F \Pi. \quad (4.15)$$

In particular, for an N outcome detector, the matrices will have dimensions $P_{D \times N}$, $F_{D \times M}$ and $\Pi_{M \times N}$. We tried to invert Eq. 4.15 using a semi-definite solver such as Yalmip (therefore making a simple minimisation of a quadratic problem). We imposed $\pi_n \geq 0$ and $\sum_n \pi_n = \mathbb{I}$ and optimised:

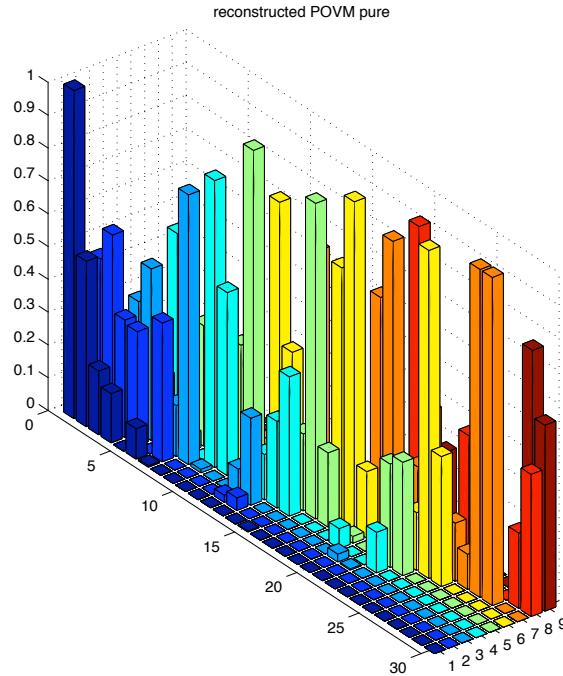


Figure 4.12: POVM reconstruction, using only minimisation from eq. 4.16. Dark blue: $\pi_0 \text{ clicks} = \sum_{i=0}^{15} \theta_i^{(0)} |i\rangle\langle i|$, lighter blue, $\pi_1 \text{ click} = \sum_{i=0}^{15} \theta_i^{(1)} |i\rangle\langle i|$, etc... displayed up to Fock layer 30.

$$\min \| \tilde{P} - F \Pi \|_2. \quad (4.16)$$

The obtained POVMs $\{\pi_n\}$ showed irregular dips and a structure very dissimilar from what a TMD was expected to do.

The Fig. 4.11 shows a typical result data, and Fig. 4.12 shows it for higher photon numbers revealing an even more irregular structure. A first approach was to consider that a certain uncertainty in the intensity of the coherent states $x = |\alpha|^2$ existed. If D values of x were measured then the real $\bar{x} = (x_1, x_2, \dots, x_D)$ might have actually been, $\bar{x}_{\bar{\delta}} = (x_1(1 + \delta_1), x_2(1 + \delta_2), \dots, x_D(1 + \delta_D))$. To address the effect of this uncertainty on our minimisation we can artificially introduce noise and then average over many runs of the optimisation. That is: Since F in Eq. 4.16 depends on x , for a family of $\{\bar{\delta}_j\}$ randomly distributed we run the optimisation and obtain a family of $\{\pi_n^{(\bar{\delta}_j)}\}$. To choose the random $\{\bar{\delta}_j\}$ we take a Gaussian distribution with $\sigma = 2\%|\alpha|^2$ since that was the

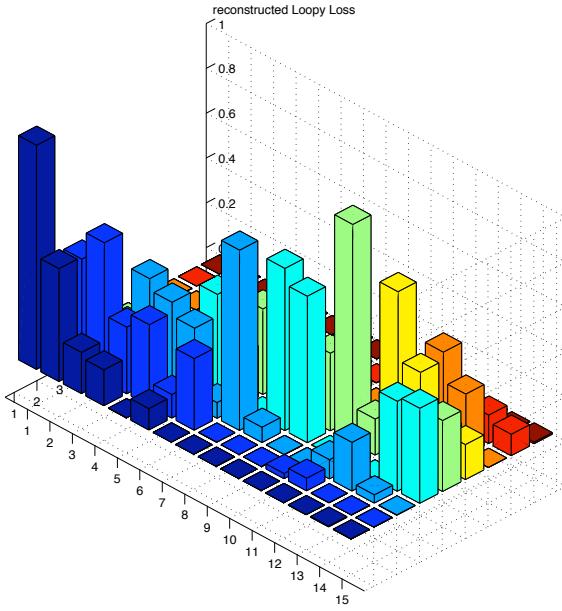


Figure 4.13: POVM reconstruction, using direct averaging (150 runs with 1% Gaussian noise).

magnitude of the measured jitter. Subsequently we average over the POVMs obtained with different “jitters” $\bar{\delta}_j$ in \bar{x} , obtaining:

$$\pi_{average}^{(n)} = \sum_j \pi_n^{(\bar{\delta}_j)} / N.$$

Making 200 runs of this averaging corrected some irregularities in the POVMs but barely solved the “dips” observed. Fig. 4.13 and Fig. 4.14 are an example of this lack of success.

A key objection is that this was not the proper quantum mechanical treatment of uncertainty in x . Each probe state would be best described by a mixture of coherent states,

$$\rho_x = \int d^2\beta |\beta\rangle\langle\beta| f_x(\beta) \quad (4.17)$$

$$= \sum_{n,m=0}^{\infty} R_{n,m,x} |n\rangle\langle m|, \quad (4.18)$$

where $f_x(\beta)$ is some distribution centered around x in phase space. We can integrate this

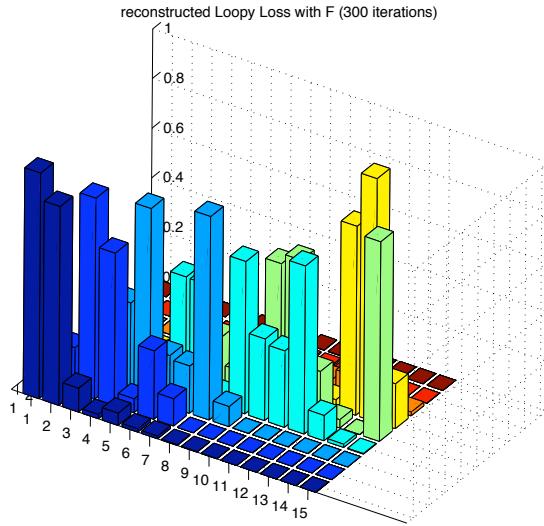


Figure 4.14: POVM reconstruction, using direct averaging (300 runs with 2% Gaussian noise).

state ρ_x over the complex phase since we have no phase reference and focus solely on the amplitude of the coherent states or mixtures thereof. Measurements reveal that the intensity of the laser varies from pulse to pulse following a distribution that looks like a Lorentzian with a tail. A good approximation can however be made using a Gaussian distribution with standard deviation $\sigma = 0.02|\alpha|^2$ implying

$$R_{n,m,\alpha} = \frac{1}{\sigma\sqrt{2\pi}\sqrt{n!m!}} \int \beta^{n+m} e^{-\beta^2} f_x(\beta) d\beta.$$

with $f_x(\beta) = e^{-(\beta^2 - x)^2/(2\sigma^2)}$. The detection probability for outcome n is then

$$p(\alpha, n) = \sum_{k=0}^{\infty} R_{k,k,\alpha} \theta_k^{(n)}. \quad (4.19)$$

To simplify these calculations we can write a distribution in $\sqrt{x} = |\alpha|$,

$$\rho_{|\alpha|} = \int d^2\beta |\beta\rangle\langle\beta| g_{|\alpha|}(\beta) \quad (4.20)$$

with $g_{\alpha}(\beta) = e^{-(\beta-\alpha)^2/(2\Gamma^2)}$. In this case Γ is chosen such that the approximation $f_x(\beta) \simeq g_{\alpha}(\beta)$ holds. These subtleties however do barely alter our results and POVMs

are as irregular as previously.

To evaluate the difference introduced by the pure ($|\alpha\rangle\langle\alpha|$) or mixed state ($\rho_{\langle\alpha\rangle}$) approach we studied their influence on the reconstructed POVMs. In the regularised optimisation (i.e. for our final results), we compared the POVMs obtained with each description finding that:

$$\frac{\|\Pi_{\text{pure}} - \Pi_{\text{mixed}}\|_2}{\|\Pi_{\text{mixed}}\|_2} \leq 0.7\% \quad (4.21)$$

and the largest relative difference between any two $\theta_k^{(n)}$ coming from a mixed state or a pure state derivation was 1.3%. Furthermore the reconstructed probability distributions are so close that they are indistinguishable on the scale of Fig. 4.6. This reinforces our earlier expectation that technical noise in the laser will be negligible when using single-photon-level coherent states. This differs from homodyne tomography where technical noise can shift a strong local oscillator to a nearly orthogonal state.

However, since the problem of the irregular POVMs is not solved by the mixed state description we need to look further into the origin of our strange results. One first remarkable (but expected) property is that large variations in the photon number degree of freedom of the POVMs result in minuscule differences in the probability distributions (see Fig. 4.7). Since one convolutes the photon number distribution with a Gaussian in α to obtain the Q-function this behaviour was expected. Conversely this means that small errors or statistical fluctuations in the Q-function can result in large errors in the POVM elements. Consider for example that if instead of

$$\min \| \tilde{P} - F \Pi \|_2$$

we try to minimise

$$\min \| F^{-1} \tilde{P} - \Pi \|_2$$

the SDP solver finds no sensible solution. This is because using the Moore-Penrose pseudo-inverse we find $F^{-1}F \neq \mathbb{I}$ due to its inherent ill conditioning.

Various methods exist to try and regularise these problems. Whatever the chosen method it should assume as little knowledge as possible about the specific form of the sought POVM. For example since F has very small values for high photon numbers one could enhance those values while preserving the minimisation target. For example we could run the optimisation

$$\min \| \tilde{P}\mathcal{D} - F \Pi\mathcal{D} \|_2$$

where \mathcal{D} is a diagonal matrix aimed at regularising the problem. This can be shown to introduce some improvement. It is however hard to find the exact form of $\mathcal{D}_{i,j} = f(i, j)$ that yields ‘good’ results without any prior knowledge about the expected POVMs. In addition (roughly speaking) it is hard to find a balance between having good results for low photon numbers and high photon numbers.

Another approach is to introduce a sort of damping or specific penalisation. For

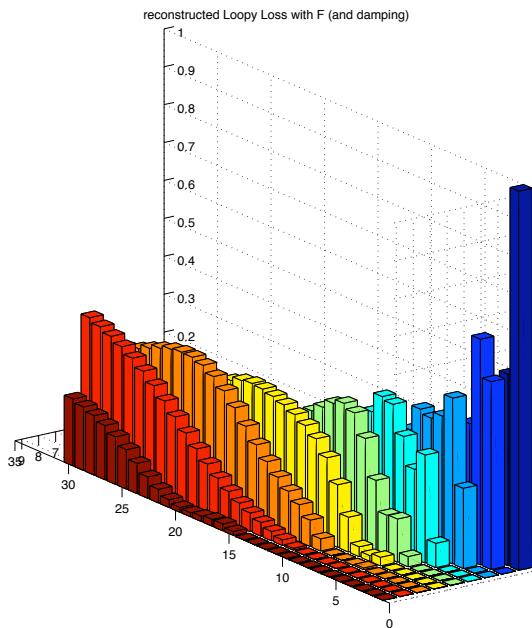


Figure 4.15: Minimisation using damping method on Eq. 4.22. Note that the point of view is opposite that of the previous plots. We see some dips around the 5th and 7th Fock layer.

example one could define a diagonal matrix M such that $M(i, j) = \delta_{i,j} \frac{1}{i}$, and use it to redistribute the weight of each POVM element, avoiding unreasonably large POVM element amplitudes (that compensate for low values in F). The optimisation could be recast as,

$$\min \{ \| \tilde{P} - F \Pi \|_2 + 0.03 \| M\Pi \|_2 \} \quad (4.22)$$

A result of this can be seen on Fig. 4.15. This method has the same shortcomings as the previous one: it is sensitive to the choice of parameters and the exact form of $M(i, j)$ is hard to determine without detailed prior assumptions.

A more reasonable method is to capture the relative smoothness of the POVM from a lossy detector. This method is also called smoothing regularisation [BV04]. In this case one single assumption needs to be made. The POVMs should exhibit a certain degree of “smoothness”.

4.4.3 Smooth or Not?

Let us first define what we mean by smooth. Smooth will mean in this context that the difference $\theta_k^{(n)} - \theta_{k+1}^{(n)}$ is small. And in the optimisation context we will mean that our minimisation is defined as follows:

$$\min \{ \| \tilde{P} - F \Pi \|_2 + y\mathcal{S} \}. \quad (4.23)$$

with

$$\mathcal{S} = \sum_{k,n} [\theta_k^{(n)} - \theta_{k+1}^{(n)}]^2.$$

Should the POVMs of our detector be smooth at all? Should the POVMs of an arbitrary detector be smooth? One simple argument is that any lossy detector should have some degree of smoothness. Indeed if an optical detector has a POVM element with non-zero amplitude in $|n\rangle\langle n|$, then if it is lossy, it will have a positive amplitude in $|n+1\rangle\langle n+1|$, $|n+2\rangle\langle n+2|$, etc decreasing with n but different from zero. In fact, in general, if the detector has a finite efficiency η which can be modelled with a BS, it will impose

some smoothness on the distribution $\theta_k^{(n)}$. That is because if $G(r)$ is the probability of registering r photons and $H(q)$ is the probability that q were present then the loss process will impose [Pre05]:

$$G(r) = \sum_q \binom{q}{r} \eta^r (1 - \eta)^{q-r} H(q).$$

Consequently, if $\theta_k \neq 0$ then θ_{k+1} , θ_{k+2} etc. cannot be zero, but will have some relatively smooth distribution. This simple physical argument calls for smoothness (but still should allow sharp transitions for $m < n$).

For this detector (and for any photodiode based detector) assuming loss is reasonable and can make the ‘smoothness’ requirement plausible. Let us however see if without looking at the specific shape of our POVM we can find an optimal *smoothing coefficient* y and justify further the use of the smoothing regularisation. One way to test

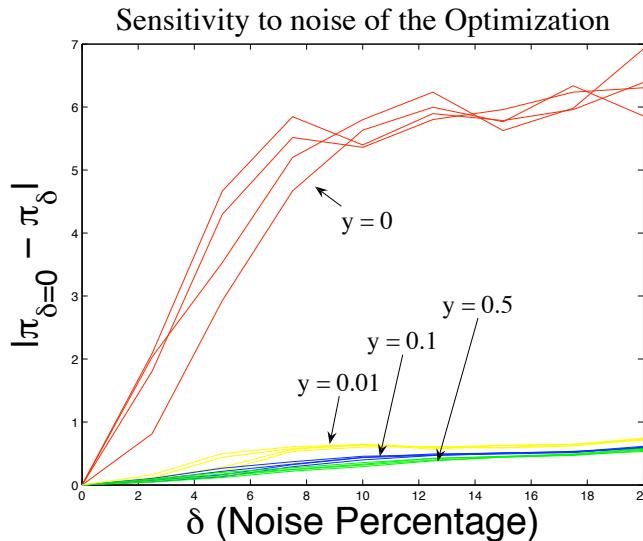


Figure 4.16: Illustration of the sensitivity to noise for two different minimisation methods ($y = 0$: No regularisation and $y \neq 0$: with smoothing regularisation). For each value of y and δ we ran the optimisation 4 times and displayed the results here to illustrate this variation

this method is to quantify how resilient it is to noise in the data. To do so we intro-

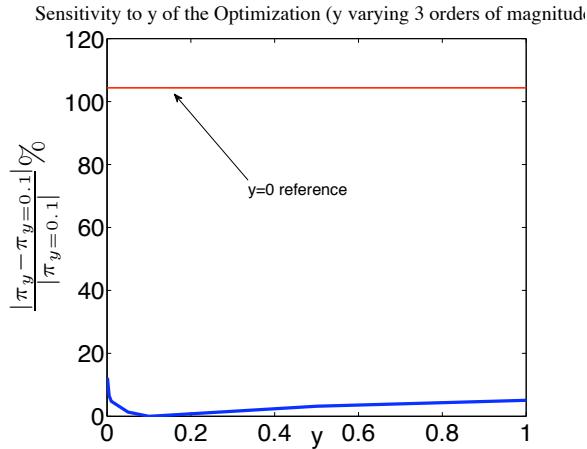


Figure 4.17: Illustration of how sensitive the optimisation is to the specific choice of y . This plot shows the relative error with respect to the POVM elements obtained using $y = 0.1$, as a function of y . In red, and only for reference (since it does not change with y), the value of the relative error for $y = 0$ (no smoothing) is shown. We vary y in the range [0.001-1]. It is remarkable that a 10000% variation in y results in only a 12% variation. For $y \in [0.05 - 0.2]$ the relative error is less than 2%. in Π .

duce additional noise in $x = |\alpha|^2$ to the measured data. For example we can alter x in $\tilde{P}_{i,n} = P(x_i(1 + \delta_i\%), n)$ where δ is a random variable distributed around zero with a Gaussian distribution. This simulates a statistical uncertainty in the measurement of the coherent state. To see its effect on the reconstruction we use the figure of merit $\|\Pi_\delta - \Pi_{\delta=0}\|_2$. This quantity should evaluate how POVMs differ from the one without noise. It is seen that the additional smoothing penalty makes the optimisation more robust, largely independent of the value of y (we can multiply y by a 100 and stay in the same regime). Using this smoothing regularisation with noisy data seems therefore a good choice.

Another question we might want to ask is how sensitive this optimisation is to the exact choice of y . To do so we may use the following procedure: compare the POVM obtained using $y = 0.1$ with that obtained varying y over 4 orders of magnitude. On Fig. 4.17 we plot the relative error $100 * |\Pi_y - \Pi_{y=0.1}| / |\Pi_{y=0.1}|$. Remarkably doubling the value of y results in an overall relative error in the POVM of less than 1%. Multiplying

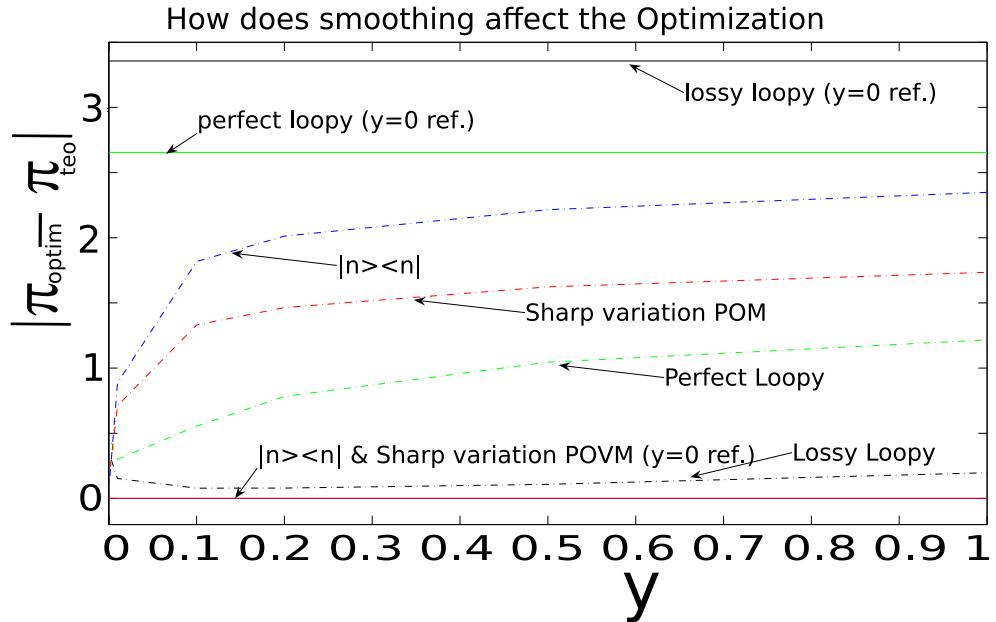


Figure 4.18: Illustration of how too much smoothing can fail to capture the sharp variations of a POVM. (note that *loopy* stands for *loopy detector*. We define Π_{teo} as the matrix containing the POVM elements of the theoretical POVMs. From them we generate a probability distribution and reconstruct the POVMs Π_{optim} with the smoothing regularised optimisation. The dotted lines represent $\|\Pi_{optim} - \Pi_{teo}\|_2$ for different values of y and for a variety of POVMs (see following plots). The horizontal lines represent that same difference for $y = 0$ and are plotted for reference.

(or dividing) y by 10 gives a variation below 5% and 100 fold variation results in a 12% variation. If we compare how this differs from the $y = 0$ case which is 110 % different then we can conclude that the optimisation is quite insensitive to the exact choice of the *smoothing parameter* y . The following table provides some values for reference.

y	y variation	Π relative error
0.0001	x 1000	27.3%
0.001	x 100	12.2%
0.01	x 10	4%
0.05	x 2	1%
0.5	x 5	3%
1	x 10	5%

4.4.4 Sharp and Smooth

These findings however prompt the following question: Is it possible for the smoothing regularisation to wash out all the sharp features of the POVM, thus smoothing in excess? This of course is a legitimate question that further restricts the reasonable range for y . To study that effect we analyse four cases:

1. A theoretical loss-less TMD, based on the model described in Eq. 4.6.
2. a lossy TMD, based on the above with added loss from an $R = 52\%$ BS.
3. a perfect photon number detector, therefore with $\pi_n = |n\rangle\langle n|$.
4. an artificial POVM with sharp variations. (having terms such as $\pi_0 = |0\rangle\langle 0| + |2\rangle\langle 2|$ while keeping $\sum_i \pi_i = \mathbb{I}$).

To study the smoothing we generate the POVM elements $\{\pi_n\}$ numerically, build a probability distribution $\text{Tr} \{\rho_\alpha \pi_n\}$ and retrieve the π_n using the optimisation from Eq. 4.23 for an increasing range of y -s. then compare this results with the theoretical POVMs we defined in order to generate the PD. All optimisations are done using the mixed-state approach from Eq. 4.20. Broadly speaking we find two behaviours: POVMs with terms that decay slowly in photon number need regularisation and are quite insensitive to the precise y . For sharp POVMs (without loss) the range $0 < y < 0.01$ preserves their shape quite well, but further smoothing hides their true shape. These properties are further illustrated in the figures that follow.

Lossy TMD:

Fig. 4.19 presents the evolution of the ‘4 click’ POVM element as we add more smoothing (or increase y in Eq. 4.23). This element is chosen as an illustrative example but the full details can be seen in appendix A.0.1. The figure shows in blue the coefficients $\theta_i^{(4)}$ in $\pi_4^{\text{rec}} = \sum_{i=0}^{60} \theta_i^{(4)} |i\rangle\langle i|$, where rec means reconstructed. In yellow, stacked on top

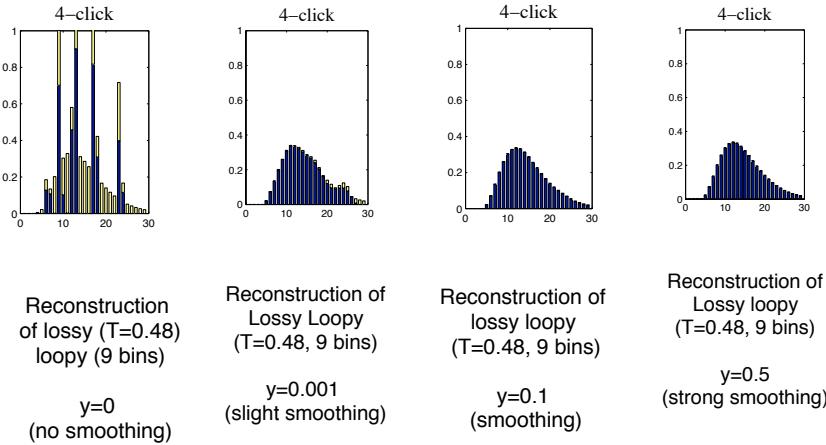


Figure 4.19: Smoothing Evolution for a lossy TMD detector (loss=52%). We show as an example the evolution of the POVM element $\pi_{4 \text{ clicks}} = \sum_{i=0}^{60} \theta_i^{(4,rec)} |i\rangle\langle i|$ as we increase the amount of smoothing (in y). The yellow bars display $|\theta_i^{(4,teo)} - \theta_i^{(4,rec)}|$ stacked on top of $\theta_i^{(4,rec)}$.

of $\theta_i^{(4)}$ we display $|\theta_i^{(4,rec)} - \theta_i^{(4,teo)}|$, where *teo* refers to the original POVM we used to generate the probability distribution. Clearly the smoothing improves the result and the exact value of y is rather unimportant. A sharp feature that is preserved however is $\theta_i^{(4)} = 0$ for $i < 4$ proving a good agreement with the model.

Loss-Less TMD: Fig 4.20 shows also the ‘4 click’ event and the error associated with the reconstruction (yellow). This TMD shows in its distribution the finite number of bins as we described earlier. The distribution is not as broad as that of the lossy-loopy-detector and the smoothing is therefore not so effective. The raw SDP (with $y = 0$) performs quite well, and the POVM is quite insensitive to the smoothing, although, when given 1/2 of the weight in the optimisation ($y = .5$) the smoothing starts to become too important.

Perfect Number Detector: Fig 4.21 shows also the ‘4 click’ event which in this case is simply $\pi_4 = |4\rangle\langle 4|$. A very interesting feature is that the simple SDP with $y = 0$ achieves a perfect result. This happens in spite of using a mixed state as a probe state (mixture of amplitudes $|\alpha|$ around $|\alpha\rangle\langle\alpha|$). The reconstruction is then robust for very

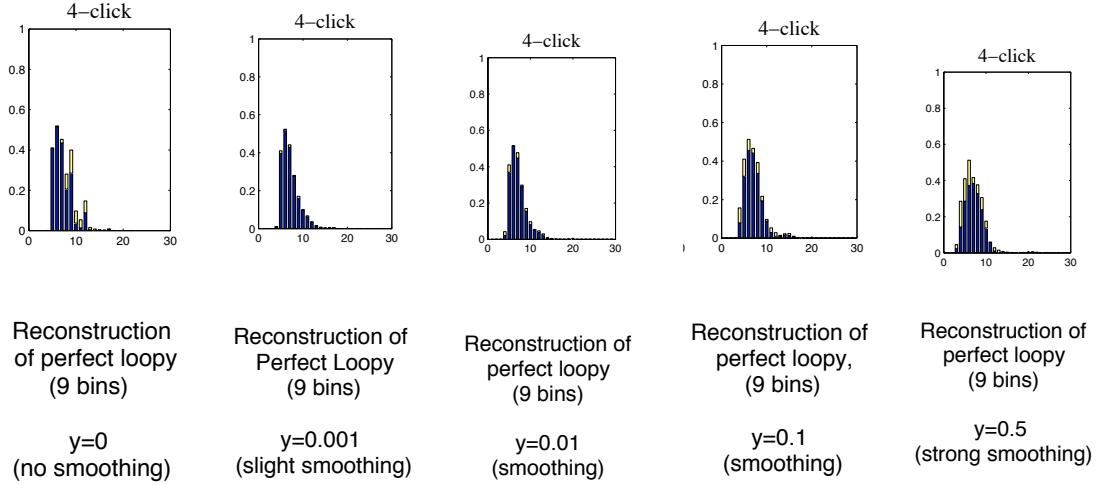


Figure 4.20: Smoothing Evolution for a perfect TMD detector (no loss). We show as an example the evolution of the POVM element $\pi_4 \text{ clicks} = \sum_{i=0}^{60} \theta_i^{(4,rec)} |i\rangle\langle i|$ as we increase the amount of smoothing (in y). The yellow bars display $|\theta_i^{(4,teo)} - \theta_i^{(4,rec)}|$ stacked on top of $\theta_i^{(4,rec)}$.

well defined and sharp features, where the higher decaying coefficients don't introduce instabilities.

Sharp POVM: This element has no claim of realism but was artificially generated to push the limit of the smoothing regularisation. The element displayed in Fig. 4.22 is $\pi_4 = |7\rangle\langle 7| + |9\rangle\langle 9|$ and we can see that $y = 0.1$ is already too much smoothing. Certainly to reconstruct a completely loss-less detector with such a structure smoothing is not an appropriate strategy. We must remember however that all current photon-number detectors that count particles do exhibit loss, and have therefore some degree of smoothness in them.

4.5 Detailed Assumptions

The strategy adopted here to characterize states raises fundamental questions about the kind of information we can extract from Nature. We have highlighted how state tomography performed without an accurate characterization of the detectors can lead to

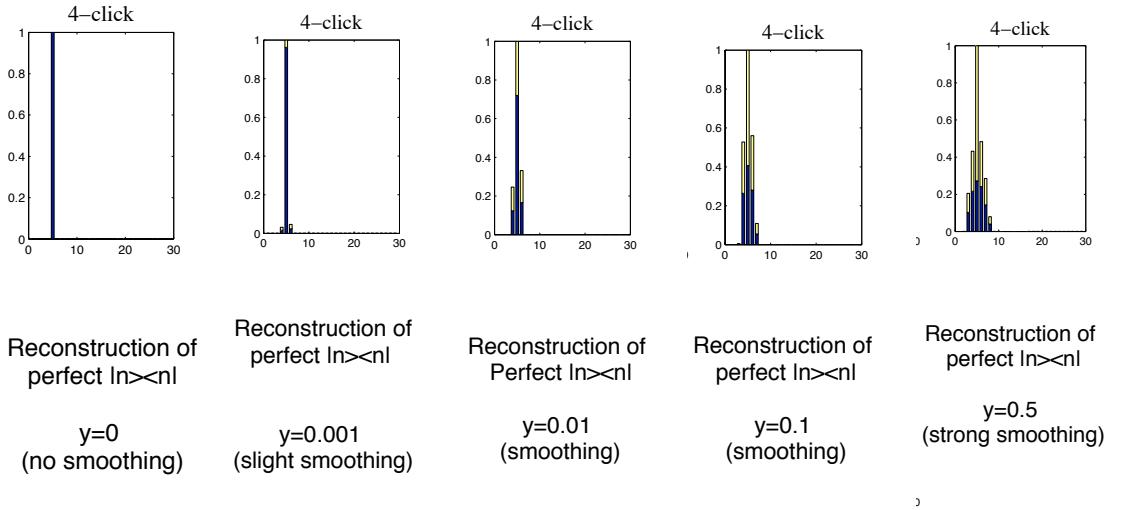


Figure 4.21: Smoothing Evolution for a perfect photon number detector, that is one with $\pi^{(n)} = |n\rangle\langle n|$. We show as an example the evolution of the POVM element π_4 clicks = $\sum_{i=0}^{60} \theta_i^{(4,rec)} |i\rangle\langle i|$ as we increase the amount of smoothing (in y). The yellow bars display $|\theta_i^{(4,teo)} - \theta_i^{(4,rec)}|$ stacked on top of $\theta_i^{(4,rec)}$.

unwanted errors but the converse is obviously true. Either our state is well known or our detector well characterized. In our case a few implicit assumptions are made. Some relate to the standard quantum mechanics formalism and others to the specifics of the experiments in question. In our case we assume that the states produced by the laser are mixed single mode coherent states. The pulses produced have a varying amplitude from one pulse to the next one but are assumed to be spectrally identical (although it's not clear that minor spectral differences would affect their detection probabilities).

Additionally it is assumed that the calibrated power meter *can* measure to within 5% error the intensity of the pulses produced. If it indeed measures the intensity of coherent states we can build the mixed state distribution mentioned above. Following standard quantum optics formalism we also assume that coherent states keep their photon number distribution and coherence under neutral density filter and beam splitter attenuation. The next assumptions have to do with the detector. That it can be described with a nine element POVM (corresponding to nine outcomes) seems a reasonable assumption.

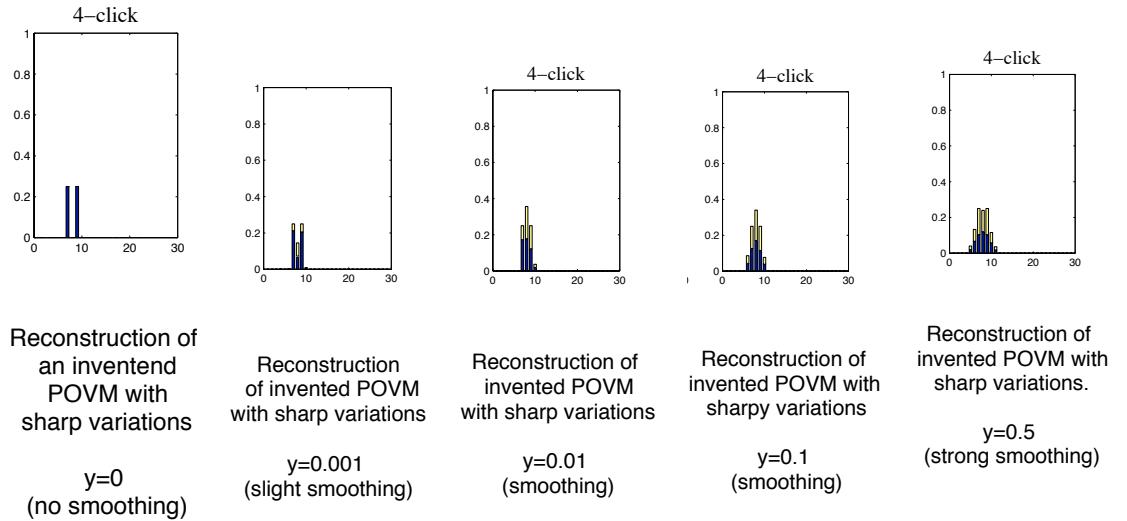


Figure 4.22: Smoothing Evolution for an invented POVM with sharp variations. Displayed is $\pi_4 = |7\rangle\langle 7| + |9\rangle\langle 9|$. We show as an example the evolution of the POVM element $\pi_4 = \sum_{i=0}^{60} \theta_i^{(4,rec)} |i\rangle\langle i|$ as we increase the amount of smoothing (in y). The yellow bars display $|\theta_i^{(4,teo)} - \theta_i^{(4,rec)}|$ stacked on top of $\theta_i^{(4,rec)}$.

Underlying this hypothesis lies the memory-less detector. In other words, the assumption that previous measurements don't modify the result of future measurements. Some of these hypothesis could be further explored if we knew well some detectors or some states. For example the response of BS and neutral density filters to single photons could be explored (granted good single photons and reliable single photon detectors). The time independence of the POVMs could also be studied with well known states. The excellent fit of the data to the Q-function suggests however that a complex conspiracy should occur to modify the quantum states sent to the detector at each shot while varying the POVMs accordingly.

In an attempt to generalize the concept of tomography we could simply take a black box approach: prepare a collection of unknown states, measure them and try to draw conclusions about both the detector and the states.

4.5.1 Without assumptions, can we have the pie and eat it too?

In the most general tomographic setting presented above, we could have some classical controls to prepare quantum states characterized by the index $\{\lambda\}$ and some classical pointer to indicate the possible outcomes $\{n\}$. Minimizing the set of assumptions would constrain us to draw our conclusions exclusively from the probability distribution $p_{\lambda,n}$. Of course many assumptions need to be made to take us beyond this stage. As an example consider the setup where $\{\lambda\} = \{1\}$ and $\{n\} = \{1, 0\}$ with $\{p_{1,0}, p_{1,1}\} = \{a, b\}$. This black box would make a random variable describing a biased coin indistinguishable from, say the quantum states $\sqrt{a}|2011\rangle + \sqrt{b}|0438\rangle$ or $a|0\rangle\langle 0| + b|1\rangle\langle 1|$ measured with the appropriate detector. In a more extreme case, we could prepare a million different states with different λ 's and obtain always the same result n_1 gaining no information at all.

To further constrain the problem we can add the standard assumptions: Known dimensionality of the state space, \mathcal{H}_d , a family of states $\left\{\rho_\lambda / \text{Tr } \{\rho_\lambda\} = 1, \rho_\lambda^\dagger = \rho_\lambda, \rho_\lambda \geq 0\right\}$ and for an N outcome detector $\sum_n \pi_n = \mathbb{I}$ in addition to $\pi_n \geq 0$. These assumptions together with their relationship $p_{\rho,n} = \text{Tr } \{\rho\pi_n\}$ leave the pair detector/state still far from determined. Staying within the finite dimensional setup each ρ_λ has $d^2 - 1$ parameters and the POVM elements have $(N - 1)d^2$. More problematic is however the relation of the parameters in λ to the $d^2 - 1$ parameters which is in principle unknown. The values λ_i of all our $d^2 - 1$ knobs in our preparation machine could simply be preparing a state with $\rho_{0,0} = \sum_i \lambda_i$ and a random (or conspiratory) distribution of all other parameters.

This discussion simply highlights the inherent difficulties that a fully general inference (or tomographic) scheme entails. Reasonable assumptions are thus needed but the question of general tomography remains an interesting one to be explored. In this direction, some progress has been made in self-testing maps. In this context states are prepared with classical recipes and families of unitary gates are revealed with few assumptions about the quantum states (however known measurements in the computational basis are

allowed) [DMMS99].

4.5.2 Model of the detector

Coming back to the case where all our reasonable assumptions are granted, we could wonder how accurate our model of the detector might be. One could even wonder if any information can be obtained from the discrepancy between our model and the reconstructed POVMs. For instance one could think that if a model assuming linear optics and linear loss cannot match the reconstruction this could reveal new forms of loss in the detector. However the current fit is so precise that any discrepancy is well beyond the error bars. The error in this kind of tomographic reconstruction has been studied only in very recent developments [AS08] and is well beyond the 98% fidelity we achieved.

4.6 Conclusion

As quantum information and computation implementations evolve, detectors are becoming more complex. This, as we have seen, calls for a black-box characterisation of the operators they implement. We have seen the first implementation of this type of tomography. The reconstruction methods are simple and efficient. However one has to pay close attention to the subtleties behind the ill conditioning of such reconstructions whether it's state or detector tomography. Fully characterising a detector with this method can help get rid of complex or erroneous assumptions in the modelling. Furthermore, once they are fully characterised, one can re-design or alter the detectors with a direct feedback on their performance.

Detector tomography of course benefits state tomography or metrology. For instance it enables the use of detectors that are noisy, non-linear or that operate in different ranges. As long as we know the exact POVMs, we can describe the rest of our experiment accordingly. More specifically, our well characterized TMD provides a unique tool for

performing the non-Gaussian operations described in previous chapters. This method will also allow the benchmarking of similar detectors making performance comparisons possible.

Knowledge of the POVMs allows one to ask precise quantitative questions about the power of preparing non-classical states. This opens a path for the experimental study of yet unexplored concepts such as the non-classicality of detectors. A promising avenue for future research will be to transfer well-established techniques from homodyne tomography to optical detector tomography (e.g. balanced noise-reduction, direct measurement of the Wigner function or pattern functions). Naturally an immediate next step would involve characterizing detectors with off diagonal terms and phase sensitivity.

5

Conclusion

5.1 Summary of Thesis Achievements

In this thesis we have presented three main lines of investigation. The first one concerns both the optimization and implementation of entanglement distillation in continuous variables with linear optics. The second one concerns the experimental challenges involved in proof-of-principle distillation and the third one the full characterization of detectors operating in the single photon regime.

As presented in chapter 2, small modifications to current distillation protocols can reduce resource consumption and increase probabilities by many orders of magnitude. Even so, technological breakthroughs in optical storage and switching could increase the performance of this family of protocols many fold.

Chapter 3 has shown the feasibility of increasing entanglement in continuous variables experiments by means of LOCC. Simulations using reasonable experimental parameters suggest that photon subtraction in this context can achieve entanglement increases. Furthermore, tools such as entanglement witnesses make it easy to perform rigorous

demonstrations of entanglement increases. Carrying out such first steps should pave the way towards more involved CV entanglement distillation.

Finally, in chapter 4, we have shown how to go beyond the partial calibration of complex photon-number resolving detectors. The technique of detector tomography has been shown to be a reliable way to reconstruct the operators that describe quantum detectors. To do so the thesis details the assumptions and methods used. Among the tools used we find mixed probe states, regularized quadratic optimization and various ways to test the regularization without explicitly looking at the resulting POVMs.

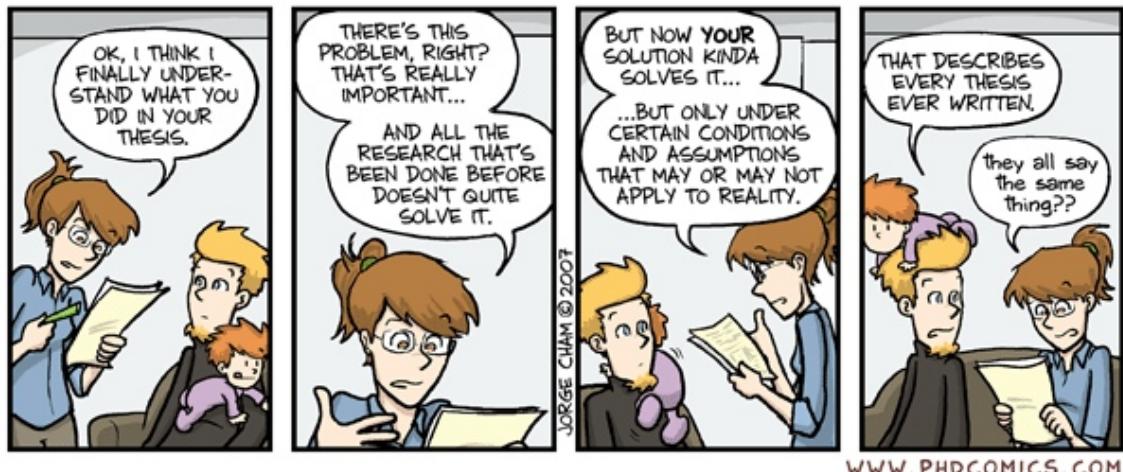
5.2 Future Work

Many of the results in this thesis have either prompted new questions or opened the door for new research areas. The distillation protocols studied have lead to new variants I was unable to explore due to time constraints. It would however be very interesting to follow the evolution of unsuccessful distillation results described in Ch. 2 when more iterations are applied. More details concerning the possibilities optical storage and switching offer should also be explored.

Concerning photon subtraction and practical distillation the collaboration with Oxford University should bear new results soon which will need careful analysis and interpretation. In the future a more accurate multimode description could also bring more insight into these experiments and the kind of sources and losses involved.

Finally, since the tomography experiments presented are the first ones of their kind there is a lot of room for investigation. Detector tomography with phase sensitive apparatuses presents an immediate challenge. To strengthen the motivation behind this method it is worthwhile researching what kind of errors a mis-characterization of detectors might lead to. What kind of errors one might incur in when doing tomography or using entanglement witnesses is a great area for exploration. From a more fundamental

perspective exploring the restrictions tomography imposes on states and POVMs is a yet unexplored and promising territory.



A

Tomography Appendix

A.0.1 Smoothing Regularization for Detector Tomography

The following plots illustrate in more detail the discussion from subsection 4.4.3.

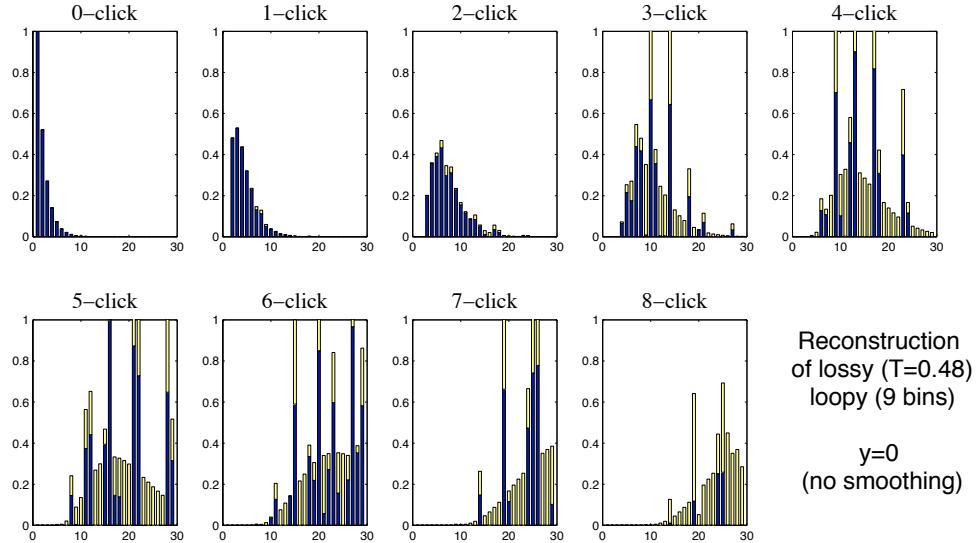


Figure A.1: Reconstruction of a theoretical lossy loopy detector with the $\|\tilde{P} - F\Pi\|_2$ optimization, where F uses the pure state description. We clearly observe that the optimization is ill conditioned for high photon numbers. In blue are the POVM element amplitudes ($\theta_i^{(n)}$) and in yellow we present the absolute value of the difference between the theoretical $\theta_i^{(n)}$ and the reconstructed ones.

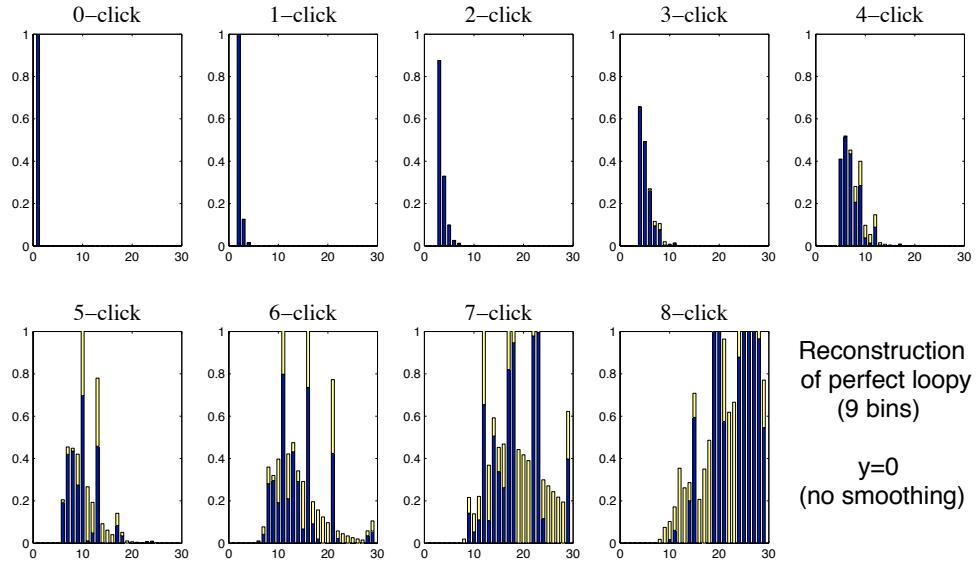


Figure A.2: Reconstruction of a theoretical 9-bin loopy detector with the $\|\tilde{P} - F\Pi\|_2$ optimization, where F uses the pure state description. We clearly observe that the optimization is ill conditioned for high photon numbers. In blue are the POVM element amplitudes ($\theta_i^{(n)}$) and in yellow we present the absolute value of the difference between the theoretical $\theta_i^{(n)}$ and the reconstructed ones.

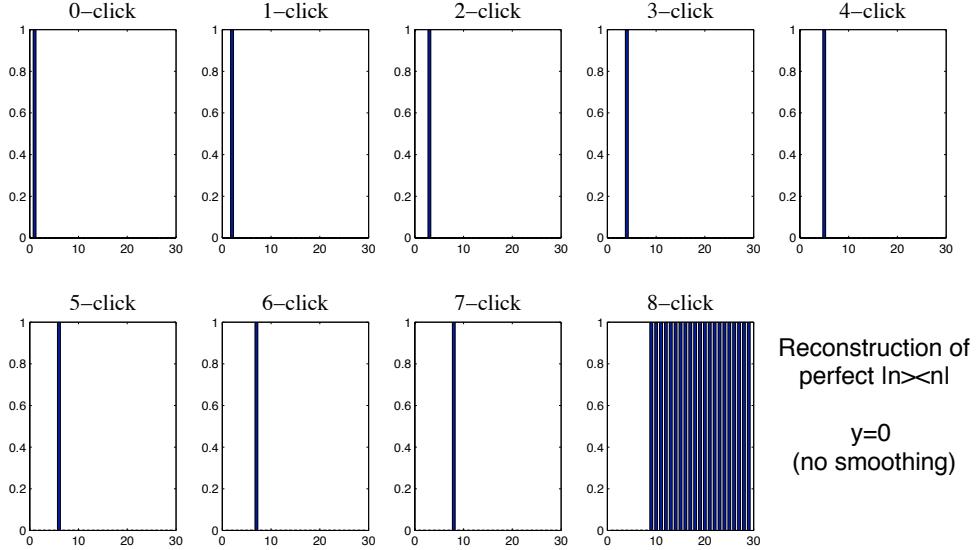


Figure A.3: Reconstruction of a theoretical detector with POVMs $\theta_i^{(n)} = |n\rangle\langle n|$ by means of the $\|\tilde{P} - F\Pi\|_2$ optimization, where F uses the pure state description. In blue are the POVM element amplitudes ($\theta_i^{(n)}$) and in yellow we present the absolute value of the difference between the theoretical $\theta_i^{(n)}$ and the reconstructed ones. In this case the error is so small that the yellow bars are imperceptible.

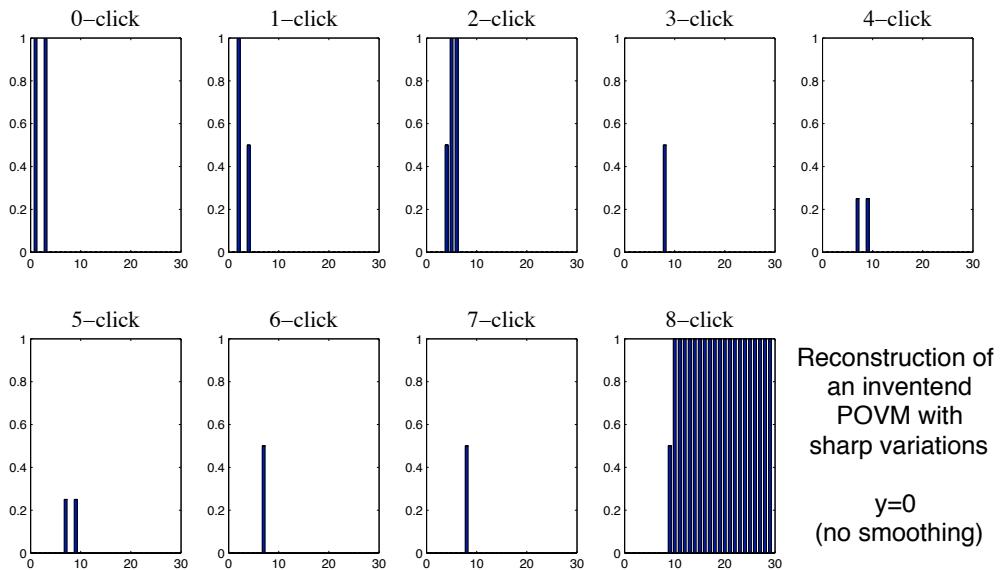


Figure A.4: Reconstruction of a theoretical detector with POVM elements specifically designed to test the smoothing regularization. We have for instance, $\pi_0 = |0\rangle\langle 0| + |2\rangle\langle 2|$, $\pi_1 = |1\rangle\langle 1| + \frac{1}{2}|3\rangle\langle 3|$, etc which have sharp variations. In this case the reconstruction is done by means of the $\|\tilde{P} - F\Pi\|_2$ optimization, where F uses the pure state description. In blue are the POVM element amplitudes ($\theta_i^{(n)}$) and in yellow we present the absolute value of the difference between the theoretical $\theta_i^{(n)}$ and the reconstructed ones. In this case the error is so small that the yellow bars are imperceptible.

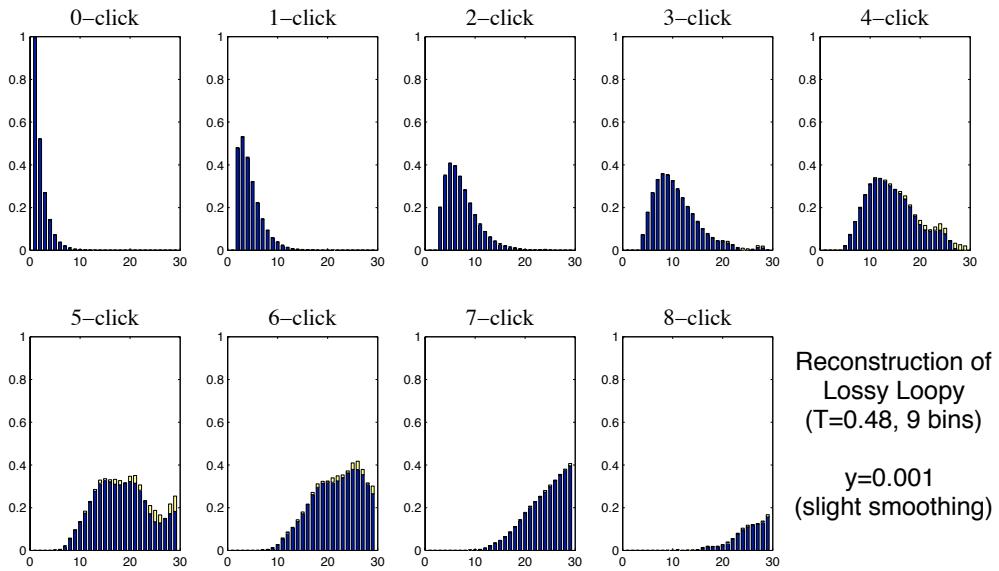


Figure A.5: Reconstruction of a theoretical lossy loopy detector with the $\|\tilde{P} - F\Pi\|_2$ optimization, where F uses the pure state description. We see that giving a small weight to the smoothing condition already improves the reconstruction greatly. In blue are the POVM element amplitudes ($\theta_i^{(n)}$) and in yellow we present the absolute value of the difference between the theoretical $\theta_i^{(n)}$ and the reconstructed ones.

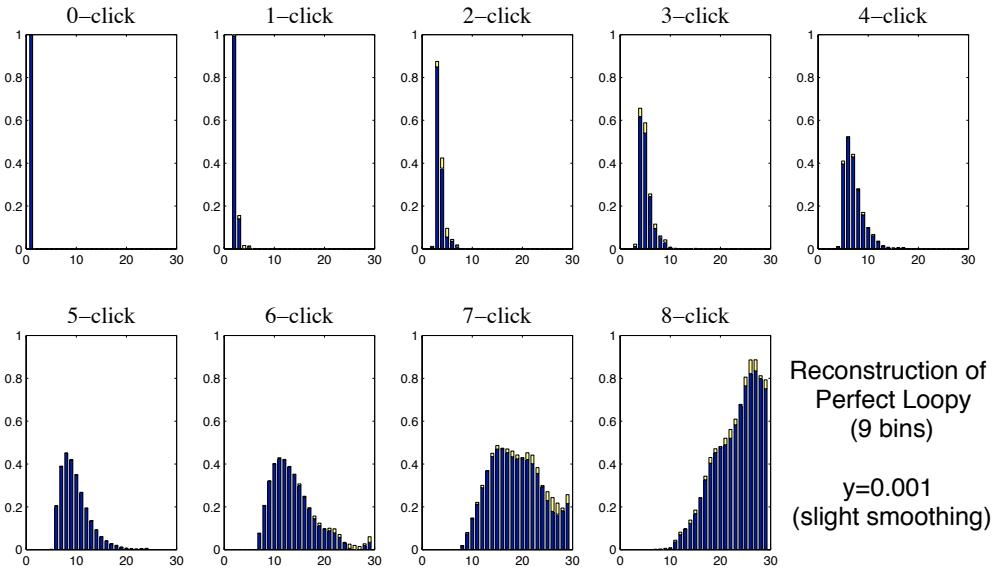


Figure A.6: Reconstruction of a theoretical 9-bin loopy detector with the $\|\tilde{P} - F\Pi\|_2$ optimization, where F uses the pure state description. Smoothing has little effect on the sharpness arising from the ‘no dark counts’. In blue are the POVM element amplitudes ($\theta_i^{(n)}$) and in yellow we present the absolute value of the difference between the theoretical $\theta_i^{(n)}$ and the reconstructed ones.

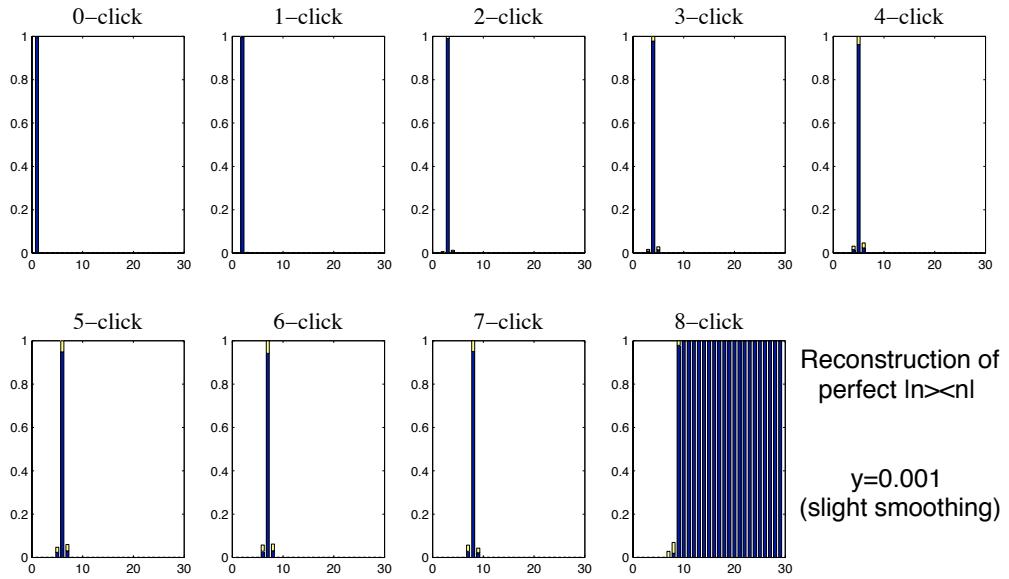


Figure A.7: Reconstruction of a theoretical detector with POVMs $\theta_i^{(n)} = |n\rangle\langle n|$ by means of the $\|\tilde{P} - F\Pi\|_2$ optimization, where F uses the pure state description. For this perfect photon number resolving detector the POVM elements are slightly altered by the $y = 0.001$ smoothing but not in any significant manner. In blue are the POVM element amplitudes ($\theta_i^{(n)}$) and in yellow we present the absolute value of the difference between the theoretical $\theta_i^{(n)}$ and the reconstructed ones.

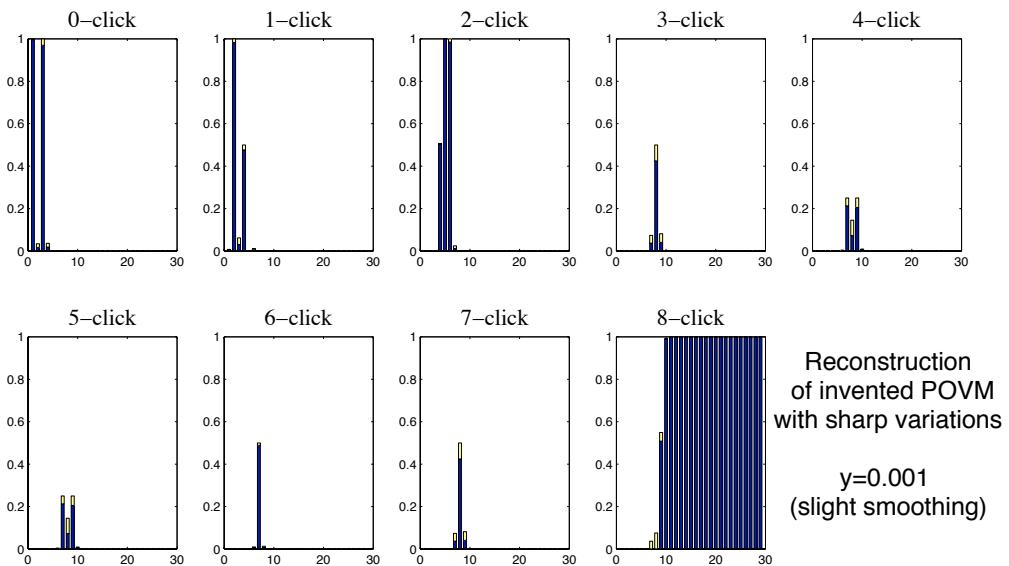


Figure A.8: Reconstruction of a theoretical detector with POVM elements specifically designed to test the smoothing regularization. We have for instance, $\pi_0 = |0\rangle\langle 0| + |2\rangle\langle 2|$, $\pi_1 = |1\rangle\langle 1| + \frac{1}{2}|3\rangle\langle 3|$, etc which have sharp variations. In this case the reconstruction is done by means of the $\|\tilde{P} - F\Pi\|_2$ optimization, where F uses the pure state description. In blue are the POVM element amplitudes ($\theta_i^{(n)}$) and in yellow we present the absolute value of the difference between the theoretical $\theta_i^{(n)}$ and the reconstructed ones. In this case the error arising from the smoothing begins to appear.

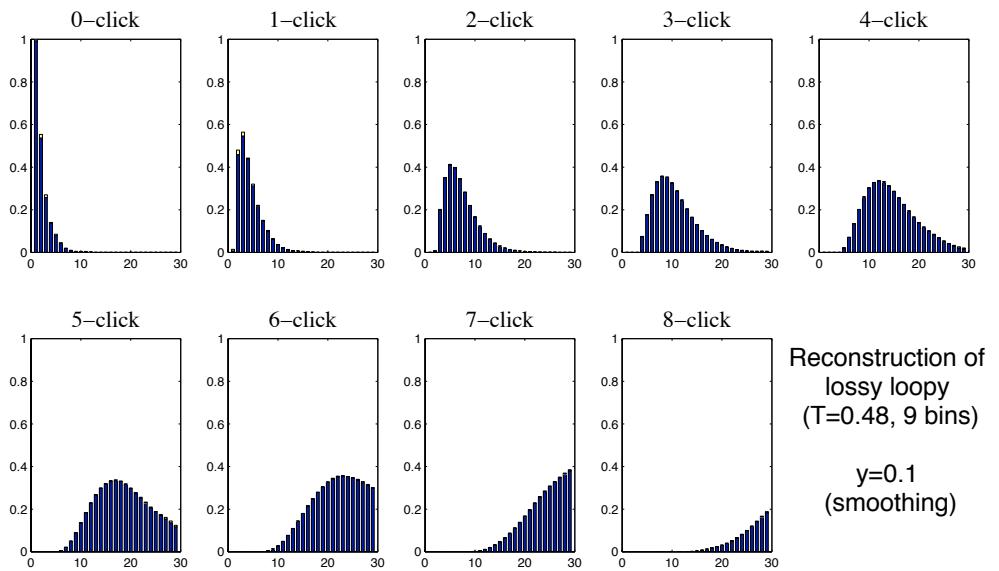


Figure A.9: Reconstruction of a theoretical lossy loopy detector with the $\|\tilde{P} - F\Pi\|_2$ optimization, where F uses the pure state description. We clearly observe that the optimization is ill conditioned for high photon numbers. In blue are the POVM element amplitudes ($\theta_i^{(n)}$) and in yellow we present the absolute value of the difference between the theoretical $\theta_i^{(n)}$ and the reconstructed ones. The agreement in this case is excellent.

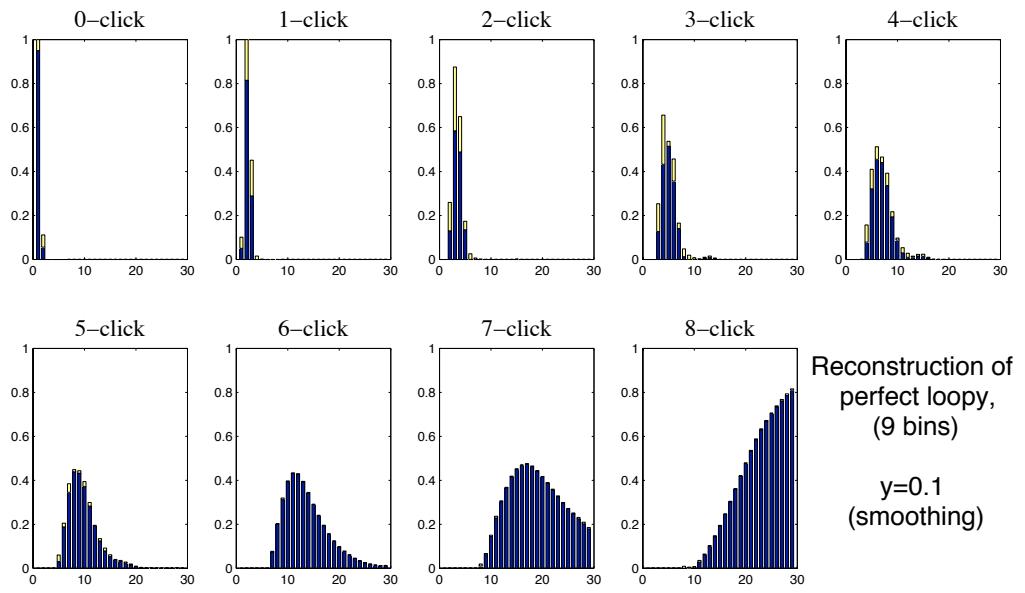


Figure A.10: Reconstruction of a theoretical 9-bin loopy detector with the $\|\tilde{P} - F\Pi\|_2$ optimization, where F uses the pure state description. We clearly observe that the optimization is ill conditioned for high photon numbers. In blue are the POVM element amplitudes ($\theta_i^{(n)}$) and in yellow we present the absolute value of the difference between the theoretical $\theta_i^{(n)}$ and the reconstructed ones. The agreement is still good especially for high photon numbers.

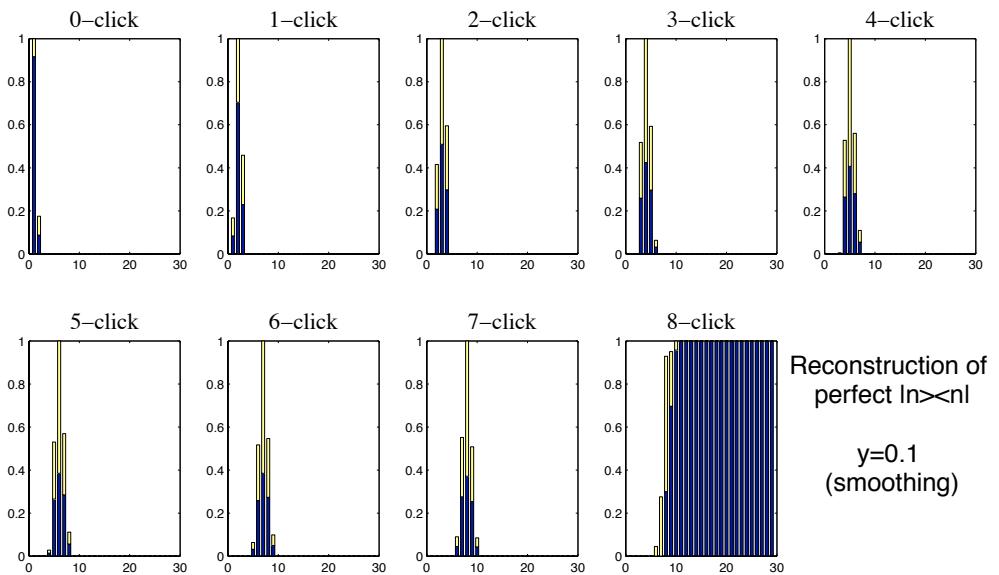


Figure A.11: Reconstruction of a theoretical detector with POVMs $\theta_i^{(n)} = |n\rangle\langle n|$ by means of the $\|\tilde{P} - F\Pi\|_2$ optimization, where F uses the pure state description. In blue are the POVM element amplitudes ($\theta_i^{(n)}$) and in yellow we present the absolute value of the difference between the theoretical $\theta_i^{(n)}$ and the reconstructed ones. In this case the error arising from the smoothing is significant.

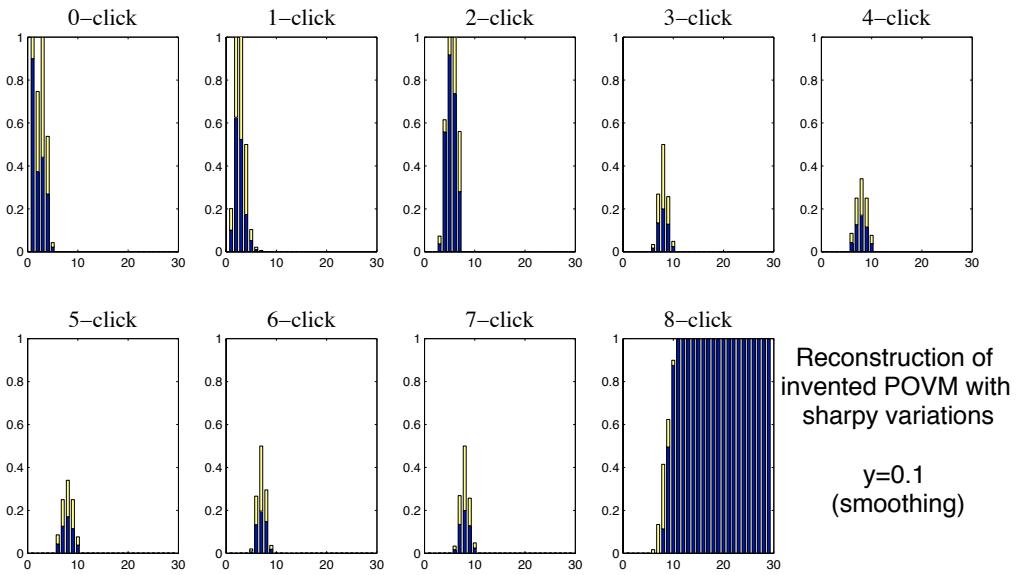


Figure A.12: Reconstruction of a theoretical detector with POVM elements specifically designed to test the smoothing regularization. We have for instance, $\pi_0 = |0\rangle\langle 0| + |2\rangle\langle 2|$, $\pi_1 = |1\rangle\langle 1| + \frac{1}{2}|3\rangle\langle 3|$, etc which have sharp variations. In this case the reconstruction is done by means of the $\|\tilde{P} - F\Pi\|_2$ optimization, where F uses the pure state description. In blue are the POVM element amplitudes ($\theta_i^{(n)}$) and in yellow we present the absolute value of the difference between the theoretical $\theta_i^{(n)}$ and the reconstructed ones. In this case ($y = 0.1$) the error for the reconstruction of a sharp POVM (with non exponentially decaying elements) is significant.

B

Homodyne Measurement Appendix

B.1 Explicit Calculation for the Gaussian Projective Measurement of a Gaussian State

We will explicitly calculate the result of performing a Gaussian destructive measurement on a Gaussian state in terms of its first and second moments. In particular we will evaluate the probability for a homodyne projective measurement described by $|\alpha\rangle\langle\alpha|$ performed on one of its modes.

B.1.1 Conventions and useful relations

- ρ will designate the density operator of an N mode Gaussian state of light.
- $\chi_\rho(\eta) = \text{Tr}[D_\eta \rho]$: characteristic function.
- $D_\eta = e^{i\eta^\sigma \hat{\mathbf{R}}}$: Displacement (or Weyl) operator.
- η or ξ : $2N$ real variables vectors.
- \hat{R}_i will be the canonical variables where $\hat{R}_{2i-1} = \hat{X}_i$ and $\hat{R}_{2i} = \hat{P}_i$ satisfy the canonical commutation relations.
- $\sigma_N = \bigoplus_{j=0}^N \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$ will be called symplectic matrix N (and will be of order $2N \times 2N$).
- $|\phi\rangle$ will be a k mode Gaussian state of light.
- $\mathcal{W}_\rho(\xi, s) = \frac{1}{(2\pi)^{2N}} \int e^{i\xi^\sigma \eta} \chi(\eta) e^{s/4||\eta||^2} d^{2N}\eta$:
- $\mathcal{W}_\rho(\xi, 0)$ = Wigner Function, $\mathcal{W}_\rho(\xi, -1)$ = Q function
- note that $\int |\alpha\rangle\langle\alpha| d^2\alpha = \pi$

B.1.2 Calculation

Our state before the measurement is ρ and the projection operator describing the projection of k modes onto a k mode Gaussian state $|\phi\rangle$ is:

$$P_k^\phi = |\phi_1\rangle\langle\phi_1| \otimes |\phi_2\rangle\langle\phi_2| \otimes \dots \otimes |\phi_k\rangle\langle\phi_k| \otimes \mathbb{I}^{\otimes N-k}.$$

After the measurement and since it is a destructive measurement leaving k modes inaccessible the state will be:

$$\text{Tr}_k[\rho'] = \text{Tr}_k \left[\frac{P_k^\phi \rho P_k^{\phi\dagger}}{\text{Tr}[P_k^\phi \rho P_k^{\phi\dagger}]} \right] = \frac{\text{Tr}_k[P_k^\phi \rho P_k^{\phi\dagger}]}{\text{Tr}[P_k^\phi \rho]} = \frac{\langle\phi|\rho|\phi\rangle}{\text{Tr}[P_k^\phi \rho]}$$

where Tr_k stands for tracing over modes from 1 to k and Tr stands for tracing over all modes (from 1 to N).

State after measurement

Let us then recall the relation between ρ and the characteristic function:

$$\rho = \frac{1}{(2\pi)^N} \int d^{2N}\eta \quad \chi_\rho(\eta) \quad D_\eta^\dagger \quad \text{in order to evaluate the operator:}$$

$$\begin{aligned} \langle\phi|\rho|\phi\rangle &= \frac{1}{(2\pi)^N} \int d^{2N}\eta \quad \chi_\rho(\eta) \langle\phi|D_\eta^\dagger|\phi\rangle \\ &= \frac{1}{(2\pi)^N} \int d^{2N}\eta \quad \chi_\rho(\eta) \langle\phi|D_{\tilde{\eta}_1}^\dagger|\phi\rangle \quad e^{-i\tilde{\eta}_2^\top \sigma_{N-k} \tilde{R}_2} \end{aligned}$$

Where we have introduced the notation:

$$\eta^\top = (\eta_1, \eta_2, \dots, \eta_{2N}) = (\tilde{\eta}_1^\top, \tilde{\eta}_2^\top)$$

with

$$\tilde{\eta}_1^\top = (\eta_1, \eta_2, \dots, \eta_{2k}), \quad \tilde{\eta}_2^\top = (\eta_{2k+1}, \dots, \eta_{2N})$$

and

$$\hat{R}^\top = (\hat{R}_1, \hat{R}_2, \dots, \hat{R}_{2N}) = (\tilde{\hat{R}}_1^\top, \tilde{\hat{R}}_2^\top)$$

with

$$\tilde{R}_1^\top = (\hat{R}_1, \hat{R}_2, \dots, \hat{R}_{2k}), \text{ and } \tilde{R}_2^\top = (\hat{R}_{2k+1}, \dots, \hat{R}_{2N})$$

which allows us to write

$$\begin{aligned} \langle \phi | \rho | \phi \rangle &= \frac{1}{(2\pi)^N} \int d^{2N} \eta \quad \chi_\rho(\eta) \operatorname{Tr} \left[|\phi\rangle \langle \phi| D_{\tilde{\eta}_1}^\dagger \right] \quad e^{-i\tilde{\eta}_2^\top \sigma_{N-k} \tilde{R}_2} \\ &= \frac{1}{(2\pi)^N} \int d^{2N} \eta \quad \chi_\rho(\eta) \chi_{|\phi\rangle \langle \phi|}(-\tilde{\eta}_1) \quad e^{-i\tilde{\eta}_2^\top \sigma_{N-k} \tilde{R}_2}. \\ &= \frac{1}{(2\pi)^N} \int d\tilde{\eta}_2 \quad e^{-i\tilde{\eta}_2^\top \sigma_{N-k} \tilde{R}_2} \underbrace{\int d\tilde{\eta}_1 \quad \chi_\rho(\eta) \chi_{|\phi\rangle \langle \phi|}(-\tilde{\eta}_1)}_{\Xi(\tilde{\eta}_2)}. \\ &= \frac{1}{(2\pi)^N} \int d\tilde{\eta}_2 \quad e^{-i\tilde{\eta}_2^\top \sigma_{N-k} \tilde{R}_2} \quad \Xi(\tilde{\eta}_2). \end{aligned}$$

We will now focus on Ξ , the scalar part of the integral. Recalling the form of the characteristic function of any k mode Gaussian function:

$$\chi_{|\phi\rangle \langle \phi|}(-\tilde{\eta}_1) = e^{-i\tilde{\eta}_1^\top \sigma_k d - \frac{1}{4}\tilde{\eta}_1^\top \sigma_k^\top \gamma \sigma_k \tilde{\eta}_1}$$

Where $d = (d_1, d_2, \dots, d_k) = (q_1, p_1, q_2, p_2, \dots, q_{2k}, p_{2k})$ is the displacement and γ is the covariance matrix (an arbitrary symmetric $k \times k$ matrix satisfying $\gamma - i\sigma \geq 0$ that contains the second moments. This describes both the projection onto a mixed state and the projection onto a pure one whose covariance matrix satisfies $(\gamma\sigma)^2 = -\mathbf{I}$). Note that in case of homodyning one mode and therefore of a projection onto a coherent state $|\alpha\rangle$, then $\gamma = \mathbb{I}_{2 \times 2}$ and $\alpha = \frac{q_0 + ip_0}{\sqrt{2}}$.

Coming back to the general case and since ρ is a Gaussian function too we can rewrite:

$$\Xi = \int d\tilde{\eta}_1 \quad e^{i\eta^\top \sigma g - \frac{1}{4}\eta^\top \sigma^\top G \sigma \eta} \quad e^{-i\tilde{\eta}_1^\top \sigma_k d - \frac{1}{4}\tilde{\eta}_1^\top \sigma_k^\top \gamma \sigma_k \tilde{\eta}_1} \quad (\text{B.1})$$

B.1 Explicit Calculation for the Gaussian Projective Measurement of a Gaussian State

Where g and $G = \begin{pmatrix} A & C \\ C^\top & B \end{pmatrix}$ are respectively the first moments and the covariance matrix whose terms have the following dimensions:

$A_{2k \times 2k}, B_{(2N-2k) \times (2N-2k)}, C_{2k \times (2N-2k)}$. We introduce additionally the notation:

$g = (\tilde{g}_1^\top; \tilde{g}_2^\top) = (g_1, g_2, \dots, g_{2k}; g_{2k+1}, \dots, g_{2N})$, and reorder the terms in the exponential:

$$\begin{aligned}\Xi &= \int d\tilde{\eta}_1 \exp \left(i \begin{pmatrix} \tilde{\eta}_1 \\ \tilde{\eta}_2 \end{pmatrix}^\top \sigma_N \begin{pmatrix} \tilde{g}_1 - d \\ \tilde{g}_2 \end{pmatrix} - \frac{1}{4} \begin{pmatrix} \tilde{\eta}_1 \\ \tilde{\eta}_2 \end{pmatrix}^\top \sigma_N^\top \begin{pmatrix} A + \gamma & C \\ C^\top & B \end{pmatrix} \sigma_N \begin{pmatrix} \tilde{\eta}_1 \\ \tilde{\eta}_2 \end{pmatrix} \right) \\ &= \int d\tilde{\eta}_1 \exp \left(i \tilde{\eta}_2^\top \sigma_{N-k} \tilde{g}_2 + i \tilde{\eta}_1^\top \sigma_k (\tilde{g}_1 - d) - \left(\tilde{\eta}_1^\top A' \tilde{\eta}_1 + \tilde{\eta}_1^\top C' \tilde{\eta}_2 + \tilde{\eta}_2^\top C^\top \tilde{\eta}_1 + \tilde{\eta}_2^\top B' \tilde{\eta}_2 \right) \right) \\ &= \exp \left(i \tilde{\eta}_2^\top \sigma_{N-k} \tilde{g}_2 - \tilde{\eta}_2^\top B' \tilde{\eta}_2 \right) \underbrace{\int d\tilde{\eta}_1 \exp \left(-\tilde{\eta}_1^\top A' \tilde{\eta}_1 + \tilde{\eta}_1^\top p \right)}_{\Theta(G, g, \gamma, d)} \\ &= \exp \left(i \tilde{\eta}_2^\top \sigma_{N-k} \tilde{g}_2 - \tilde{\eta}_2^\top B' \tilde{\eta}_2 \right)\end{aligned}$$

Where we have successively defined:

$$A' = \frac{1}{4} \sigma_k^\top (A + \gamma) \sigma_k$$

$$C' = \frac{1}{4} \sigma_k^\top C \sigma_{N-k} = \frac{1}{4} \sigma_k C \sigma_{N-k}^\top$$

$$B' = \frac{1}{4} \sigma_{N-k}^\top B \sigma_{N-k}$$

$$p = [i \sigma_k (\tilde{g}_1 - d) - C' \tilde{\eta}_2] + C' \tilde{\eta}_2 = i \sigma_k (\tilde{g}_1 - d) - 2C' \tilde{\eta}_2.$$

Let us then focus on the calculation of Θ for which we will use Appendix A.

$$\begin{aligned}
\Theta &= \int_{-\infty}^{+\infty} d\eta_1 \dots d\eta_{2k} \exp(-\tilde{\eta}_1^\top A' \tilde{\eta}_1 + \tilde{\eta}_1^\top p) \\
&= \frac{\pi^k}{\sqrt{|A'|}} \exp\left(\frac{p^\top (A')^{-1} p}{4}\right) \\
&= \frac{\pi^k}{\sqrt{|A'|}} \exp(p^\top \sigma_k^\top (A + \gamma)^{-1} \sigma_k p)
\end{aligned}$$

We can then come back to Ξ which turns out to be proportional to $\chi_{\text{Tr}_k(\rho')}(\tilde{\eta}_2)$:

$$\begin{aligned}
\Xi &= \exp(i \tilde{\eta}_2^\top \sigma_{N-k} \tilde{g}_2 - \tilde{\eta}_2^\top B' \tilde{\eta}_2) \frac{\pi^k}{\sqrt{|A'|}} \\
&\quad \exp([i \sigma_k(\tilde{g}_1 - d) - 2C' \tilde{\eta}_2]^\top \sigma_k^\top (A + \gamma)^{-1} \sigma_k [i \sigma_k(\tilde{g}_1 - d) - 2C' \tilde{\eta}_2]) \\
&= \frac{\pi^k}{\sqrt{|A'|}} \exp(i \tilde{\eta}_2^\top \sigma_{N-k} \tilde{g}_2) \exp\left(-\frac{1}{4} \tilde{\eta}_2^\top \sigma_{N-k}^\top B \sigma_{N-k} \tilde{\eta}_2\right) \\
&\quad \exp(-(\tilde{g}_1 - d)^\top \sigma_k^\top \sigma_k^\top (A + \gamma)^{-1} \sigma_k \sigma_k (\tilde{g}_1 - d)) \\
&\quad \exp\left(-i(\tilde{g}_1 - d)^\top \sigma_k^\top \frac{\sigma_k^\top (A + \gamma)^{-1} \sigma_k}{2} \sigma_k C \sigma_{N-k}^\top \tilde{\eta}_2\right) \\
&\quad \exp\left(-i \tilde{\eta}_2^\top \sigma_{N-k} C^\top \sigma_k^\top \frac{\sigma_k^\top (A + \gamma)^{-1} \sigma_k}{2} \sigma_k (\tilde{g}_1 - d)\right) \\
&\quad \exp\left(\tilde{\eta}_2^\top \sigma_{N-k}^\top C^\top \sigma_k \frac{\sigma_k^\top (A + \gamma)^{-1} \sigma_k}{4} \sigma_k^\top C \sigma_{N-k} \tilde{\eta}_2\right) \\
&= \frac{4^k \pi^k}{\sqrt{|A + \gamma|}} \exp(i \tilde{\eta}_2 \sigma_{N-k} \tilde{g}_2) \exp\left(-\frac{1}{4} \tilde{\eta}_2^\top \sigma_{N-k}^\top B \sigma_{N-k} \tilde{\eta}_2\right) \\
&\quad \exp(-(\tilde{g}_1 - d)^\top (A + \gamma)^{-1} (\tilde{g}_1 - d)) \\
&\quad \exp(-i \tilde{\eta}_2^\top \sigma_{N-k} C^\top (A + \gamma)^{-1} (\tilde{g}_1 - d)) \\
&\quad \exp\left(\tilde{\eta}_2^\top \sigma_{N-k}^\top C^\top \frac{(A + \gamma)^{-1}}{4} C \sigma_{N-k} \tilde{\eta}_2\right) \\
&= \Delta \exp(i \tilde{\eta}_2 \sigma_{N-k} \tilde{g}_2^{\text{out}}) \exp\left(-\frac{1}{4} \tilde{\eta}_2^\top \sigma_{N-k}^\top B^{\text{out}} \sigma_{N-k} \tilde{\eta}_2\right)
\end{aligned}$$

Where \tilde{g}_2^{out} and B^{out} are the new first and second moments defined as:

$$\tilde{g}_2^{\text{out}} = \tilde{g}_2 - C^\top (A + \gamma)^{-1} (\tilde{g}_1 - d) \quad (\text{B.2})$$

$$B^{\text{out}} = B - C^\top (A + \gamma)^{-1} C \quad (\text{B.3})$$

And the Δ factor is:

$$\Delta = \frac{(4\pi)^k}{\sqrt{|A + \gamma|}} \exp \left(-(\tilde{g}_1 - d)^\top (A + \gamma)^{-1} (\tilde{g}_1 - d) \right). \quad (\text{B.4})$$

Similarity transformation and the Schur Complement

An alternative and equivalent approach is to find a similarity transformation to solve the integral Ξ presented above. Indeed eq. (B.1) may be rewritten:

$$\Xi = \int d\tilde{\eta}_1 e^{i \eta^\top \sigma g' - \frac{1}{4} \eta^\top \sigma^\top G' \sigma \eta}$$

Were we have defined:

- $g'^\top = (g_1'^\top, g_2'^\top) = (\tilde{g}_1^\top, \tilde{g}_2^\top) - (d^\top, \mathbf{0})$

- $G' = \begin{pmatrix} A & C \\ C^\top & B \end{pmatrix} + \begin{pmatrix} \gamma & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} A' & C \\ C^\top & B \end{pmatrix}$

We note that $w^\top G' w = \begin{pmatrix} A' & 0 \\ 0 & B - C^\top A'^{-1} C \end{pmatrix} = \begin{pmatrix} A' & 0 \\ 0 & B' \end{pmatrix}$ where B' is the

Schur complement if we define $w = \begin{pmatrix} \mathbb{I} & D \\ 0 & \mathbb{I} \end{pmatrix}$ and $D = -A'^{-1}C$. This similarity transformation can be cast into the integral with the change of variables: $y = w^{-1}\sigma\eta$.

Since the Jacobian of the transformation $\left| \sigma \begin{pmatrix} \mathbb{I} & D \\ 0 & \mathbb{I} \end{pmatrix} \right| = 1$ then $dy^{2N} = d\eta^{2N}$. As before y_1 will be the vector with the first $2k$ variables, and y_2 will contain the remaining

ones. We may then directly write:

$$\begin{aligned}
 \Xi &= \int dy_1 \exp \left(-i y^\top w^\top g' - \frac{1}{4} y^\top \begin{pmatrix} A' & 0 \\ 0 & B' \end{pmatrix} y \right) \\
 &= \int dy_1 \exp \left(-i y_1^\top g'_1 - i y_2^\top (Dg'_1 + g'_2) - \frac{1}{4} y_1^\top A' y_1 - \frac{1}{4} y_2^\top B' y_2 \right) \\
 &= \int dy_1 \exp \left(-i y_1^\top g'_1 - \frac{1}{4} y_1^\top A' y_1 \right) \exp \left(i y_2^\top (C^\top A'^{-1\top} g'_1 - g'_2) - \frac{1}{4} y_2^\top B' y_2 \right) \\
 &= \frac{(4\pi)^k}{\sqrt{|A + \gamma|}} e^{-(\tilde{g}_1 - d)^\top (A + \gamma)^{-1} (\tilde{g}_1 - d)} e^{i y_2^\top (C^\top (A + \gamma)^{-1} g'_1 - g'_2) - \frac{1}{4} y_2^\top B' y_2}
 \end{aligned}$$

if we recall Δ from eq. (B.4), note that $y_2 = \sigma_{N-k} \tilde{\eta}_2$ and that $\sigma = -\sigma^\top$ we may finally rewrite:

$$\begin{aligned}
 \Xi &= \Delta \exp \left(i \tilde{\eta}_2^\top \sigma_{N-k} [\tilde{g}_2 - C^\top (A + \gamma)^{-1} (\tilde{g}_1 - d)] \right) \\
 &\quad \exp \left(-\frac{1}{4} \tilde{\eta}_2^\top \sigma_{N-k}^\top (B - C^\top (A + \gamma)^{-1} C) \sigma_{N-k} \tilde{\eta}_2 \right)
 \end{aligned}$$

recovering the result from eq. (B.2) and (B.3).

Moore-Penrose inverse

When $(A + \gamma)^{-1}$ is not well defined we will use the pseudo-inverse $(A + \gamma)^+$ to calculate the corresponding matrix.

Probabilities

The state after the measurement is $\rho' = \frac{\langle \phi | \rho | \phi \rangle}{\text{Tr}[P_k^\phi \rho P_k^{\phi\dagger}]}$. Also,

$$\begin{aligned}
 \langle \phi | \rho | \phi \rangle &= \frac{1}{(2\pi)^N} \int d\tilde{\eta}_2 e^{-i \tilde{\eta}_2^\top \sigma_{N-k} \tilde{R}_2} \Xi(\tilde{\eta}_2) \\
 &= \frac{\Delta}{(2\pi)^k} \left[\frac{1}{(2\pi)^{N-k}} \int d\tilde{\eta}_2 e^{-i \tilde{\eta}_2^\top \sigma_{N-k} \tilde{R}_2} \chi_{\rho'}(\tilde{\eta}_2) \right] \\
 &= \frac{\Delta}{(2\pi)^k} \rho'
 \end{aligned}$$

where we have required $\chi_{\rho'}(\mathbf{0}) = 1$ In order for ρ' to be normalized. It follows that $\text{Tr} \left[P_k^\phi \rho P_k^{\phi\dagger} \right] = \frac{\Delta}{(2\pi)^k}$. The probability density for projecting k modes onto the Gaussian state $\hat{\phi}$ characterized by (d, γ) will therefore be:

$$\text{Tr} \left[P_k^\phi \rho P_k^{\phi\dagger} \right] = \frac{2^k}{\sqrt{|A + \gamma|}} \exp \left(-(\tilde{g}_1 - d)^\top (A + \gamma)^{-1} (\tilde{g}_1 - d) \right). \quad (\text{B.5})$$

To check for consistency, this relation must obey the following particular statements:

- The probability of measuring the vacuum if our state is the vacuum is one:

$$\begin{aligned} \text{Tr}(|0\rangle\langle 0||0\rangle\langle 0|) &= \frac{2^k}{\sqrt{|A + \gamma|}} \\ &= \frac{2^k}{\sqrt{|2\mathbb{I}_{2k}|}} = 1 \end{aligned}$$

- The probability of measuring any coherent state is one:

$$\begin{aligned} 1 &= \frac{1}{\pi^k} \int \text{Tr}(|\alpha_1\rangle\langle\alpha_1| \otimes |\alpha_2\rangle\langle\alpha_2| \otimes \dots \otimes |\alpha_k\rangle\langle\alpha_k| \otimes \mathbb{I}^{\otimes N-k} \rho) d^2\alpha_1 \dots d^2\alpha_k \\ &= \frac{1}{\pi^k} \int \frac{2^k}{\sqrt{|A + \gamma|}} \exp \left(-(\tilde{g}_1 - d)^\top (A + \gamma)^{-1} (\tilde{g}_1 - d) \right) \frac{\partial d_1 \partial d_2}{2} \dots \frac{\partial d_{2k-1} \partial d_{2k}}{2} \\ &= \frac{1}{(2\pi)^k} \frac{2^k}{\sqrt{|A + \gamma|}} \sqrt{\frac{\pi^{2k}}{|A + \gamma|^{-1}}} = 1 \end{aligned}$$

- Probability $\leq 1 \iff |A + \gamma| \geq 2^{2k}$.

This property holds for covariance matrices [Bra07]. To prove that this is the case, remember that A and γ are positive semidefinite matrices. and that $-\log [\det(x)]$ is a convex function of x . We can therefore write:

$$\log \left[\det \left(\frac{1}{2} 2A + \frac{1}{2} 2\gamma \right) \right] \geq \frac{1}{2} \log [\det(2A)] + \frac{1}{2} \log [\det(2\gamma)] \quad (\text{B.6})$$

We Remind that symplectic transformations leave the determinant of covariance matrices unchanged. Symplectic diagonalization will therefore give for $|A|$ or $|\gamma|$ a determinant of the form $\prod_{i=1}^{2k} \lambda_i$ where $\lambda_i \geq 1$ are the symplectic eigenvalues.

This allows us to write $\det(2A) \geq 2^{2k}$. This property and (B.6) prove the above property.

Example

Another way to check the result is to evaluate the probability density for measuring $\frac{1}{\pi}|\alpha\rangle\langle\alpha|$ in the first mode:

$$\begin{aligned} P(\alpha_1) &= \frac{1}{\pi} \text{Tr} [\rho |\alpha\rangle\langle\alpha|_1 \otimes \mathbb{I}^{\otimes N-1}] \\ &= \frac{1}{\pi} \text{Tr} [\text{Tr}_{/1}(\rho) |\alpha\rangle\langle\alpha|_1] \end{aligned}$$

Where $\text{Tr}_{/1}(\rho)$ stands for tracing over all modes but the first one. Since ρ is a Gaussian function, $\rho_1 = \text{Tr}_{/1}(\rho)$ will have a covariance matrix which will be simply the submatrix of G corresponding to the first two canonical variables. Equally the first moments will be the first two elements of the general g vector. This way we can express the probability density for measuring α making use of the Q function. To this end note that $Q(\alpha)$ is normalized to 1 as follows: $\int Q(\alpha) d^2\alpha = \int \frac{1}{\pi} \langle \alpha | \rho | \alpha \rangle d^2\alpha = 1$. Therefore, when integrating over position and momentum like variables,

$$\int 2Q(\xi_1, \xi_2) \frac{d\xi_1}{\sqrt{2}} \frac{d\xi_2}{\sqrt{2}} = \int \frac{1}{\pi} \langle \alpha | \rho | \alpha \rangle d\xi_1 d\xi_2 = 1.$$

Therefore the probability density will be:

$$\begin{aligned} \text{Tr} \left(\frac{1}{\pi} |\alpha\rangle\langle\alpha| \rho \right) &= 2Q(\xi_1, \xi_2) \\ &= \frac{2}{(2\pi)^2} \int e^{i\tilde{\xi}_1 \sigma \tilde{\eta}_1} \chi_{\rho_1}(\tilde{\eta}_1) e^{-\frac{1}{4} \|\tilde{\eta}_1\|^2} d\eta_1 d\eta_2 \\ &= \frac{2}{(2\pi)^2} \int \left(e^{i\tilde{\xi}_1 \sigma \tilde{\eta}_1} e^{i \tilde{\eta}_1^\top \sigma \tilde{g}_1 - \frac{1}{4} \tilde{\eta}_1^\top \sigma^\top \tilde{G}_1 \sigma \tilde{\eta}_1} e^{-\frac{1}{4} \tilde{\eta}_1^\top \mathbb{I} \tilde{\eta}_1} \right) d\eta_1 d\eta_2 \\ &= \frac{2}{\pi \sqrt{|\tilde{G}_1 + \mathbb{I}|}} \exp \left(-(\tilde{\xi}_1 - \tilde{g}_1)^\top (\tilde{G}_1 + \mathbb{I})^{-1} (\tilde{\xi}_1 - \tilde{g}_1) \right) \end{aligned}$$

Which is indeed normalized so that $\int \text{Tr} \left(\frac{1}{\pi} |\alpha\rangle\langle\alpha| \rho \right) d^2\alpha = 1$. We have used the already introduced notation to distinguish the first two variables $\tilde{\xi}_1 = (\xi_1, \xi_2)$ from the $2N - 2$ remaining ones. We have also identified the Wigner function as the Fourier transform of our characteristic function with covariance matrix $\tilde{G}_1 + \mathbb{I}$. It is assumed that $\alpha = \frac{\xi_1 + i\xi_2}{\sqrt{2}}$ is the amplitude of the state we project onto, and (g_1, g_2) are the first moments of ρ_1 .

For further applications of these calculations see [Krü01, WGK⁺04, CG69].

B.1.3 Useful Integral

For the Gaussian integral of a bilinear operator plus the linear term we find:

$$\begin{aligned} I &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-\eta^\top A \eta + \eta^\top p} d\eta^N \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-x^\top D x + x^\top U p} \left| \frac{\partial d\eta}{\partial d\chi} \right| d\chi^N \end{aligned}$$

Where we have made the change of variables: $\eta^\top = \chi^\top U^\top$, $\eta = U\chi$, and therefore $\chi^\top D \chi = \chi^\top U^\top A U \chi = \eta^\top A \eta$, where D is the diagonal form of matrix A . Since A is symmetric it is diagonalized by an orthogonal matrix U which means that $\left| \frac{\partial d\eta}{\partial d\chi} \right| = |U^\top| = \pm 1$. If we call $p' = Up$ We can rewrite:

$$\begin{aligned} I &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp \left(\sum_{i=1}^N -d_i \chi_i^2 + \chi_i p'_i \right) d\chi^N \\ &= \sqrt{\frac{\pi^N}{\prod_i d_i}} \exp \left(\sum_{i=1}^N \frac{p'^2_i}{4d_i} \right) \\ &= \sqrt{\frac{\pi^N}{|A|}} \exp \left(\frac{1}{4} p^\top U^\top (U^\top A U)^{-1} U p \right) \\ &= \sqrt{\frac{\pi^N}{|A|}} \exp \left(\frac{1}{4} p^\top A^{-1} p \right) \end{aligned}$$

C

Quantavo Appendix

C.1 The Quantavo *Maple* Toolbox

The toolbox is made to be used with the following approach:

Declare an initial state, let it evolve through a quantum optical circuit involving linear optics (LO) and measurements and finally ask various questions about the structure, entanglement, and properties of the final state.

The toolbox intends to do so providing:

A) A framework in which to declare, manipulate and characterize quantum states of light (finite number of modes, and finite dimensional).

B) Procedures that implement linear operations or whole linear optics circuits on our states such as:

- Beam Splitters (BS)
- Phase Shifters (PS).
- arbitrary unitary transformations of the modes.

C) Procedures that implement arbitrary measurements (both projective or generalized positive operator valued measures (POVM)).

D) Procedures to determine probabilities and expected values for projective measurements and POVM measurements.

E) Procedures to trace out measured or inaccessible modes.

F) Procedures to calculate different entanglement measures such as:

- Entropy of entanglement
- partial trace, norm → Negativity, Logarithmic Negativity

G) Access to properties such as the mean Energy of given states.

H) Extract and display lists of coefficients along with their indexes like: $RT\sqrt{2} |001\rangle\langle 001|$

H) Tools to easily plot states and density matrices.

C.2 Getting Started

Some previous knowledge about *Maple* from MaplesoftTM is required to use this tool. However, the brief “Take a Tour of *Maple*” should suffice to get started. This toolbox works with Maple 9.5, Maple 10 and Maple 11.

C.2.1 Definitions and Notation

Procedures: Formally, in Maple, a procedure definition is a valid expression that can be assigned to a name. The procedures we will use can be thought of as a set of “rules” that generally receive one or more inputs and return one or more outputs. We will write them in typewriter face. For example the procedure `IsHermitian(M)` evaluates if a Matrix **M** describing a density matrix is hermitian. A dictionary with all the procedures involved in **Quantavo** can be found in the appendix.

Modules: Modules are repositories of procedures. By loading a module we can use its procedures in the Maple worksheet. For example **Quantavo** is the module containing the procedures we will use.

objects: The objects in which we will encode our quantum states will be written in bold face. These will include **vec**, **mat**, **matcol** and **poly** and will be introduced later.

d and K: Throughout the manual, ‘ K ’ will stand for the number of modes and ‘ d ’ for the dimension of each mode. If considering the photon number degree of freedom, then ‘ $d - 1$ ’ will be the maximum number of photons in any mode. It is important to keep track of the value of these two global variables throughout the worksheet as they play an important role in the labelling of the optical modes and translation procedures. These variables can always be updated and displayed with the procedure: `findKnd(State)`.

C.2.2 Loading the Modules

All necessary files can be found at,

<http://www.imperial.ac.uk/quantuminformation/research/downloads>

1. uncompress the file **Quantavo.zip**.
2. Save the folder QUANTAVO to a given directory. It should contain the files **Quantavo.mpl** and **Quantavo_Example_Worksheet.mw**.
3. One can start opening the worksheet **Quantavo_Example_Worksheet.mw**.
4. To use the module in a new worksheet execute the following commands ¹:

```
> with(LinearAlgebra);
> read "Quantavo.mpl";
> with(Quantavo);
```

It should return a list of all the procedures available ¹:

¹note that if the new file is in a different directory it should be,
 >read “/path-to-folder/QUANTAVO/Quantavo.mpl”;
 or under MS Windows,
 >read “C:\path-to-folder\QUANTAVO\Quantavo.mpl”;

```
[APD, BS, BuildUnitary, CoherentState, DP, Dbra, Dbraket,  
DeltaK, Dket, Dstate, Energy, Entropy, EvalState, IdentityState,  
IsHermitian, IsNormalized, LogNegativity, Negativity, POVMresult,  
PS, PlotState, Probability, Project, SqueezedVac, StateApprox,  
StateComplexConjugate, StateMultiply, StateNorm, StateNormalize,  
StatePartialTranspose, StateSort, StateTrace, TensorProduct,  
TensorVac, Traceout, Trim, UnitaryEvolution, Vac, findKnd,  
indexstate, mat2matcol, mat2poly, matcol2mat, matcol2poly,  
modesmatcol, myBS, poly2matcol, poly2vec, vec2mat, vec2matcol,  
vec2poly]
```

5. You are ready to use Quantavo !

Note that You may also save the module to your Maple library. To do so, visit the maple help on `module`, `savelibname` and `savelib`.

C.3 Toolbox

To run Quantavo one has to load the following modules:

- **LinearAlgebra** (Linear Algebra package from MaplesoftTM built in *Maple*)
- **Quantavo** (General toolbox)

Additionally, if we want to plot our states we will also need the module

- **geom3d** (geometry package from MaplesoftTM built in *Maple*)
- **plots** (plotting package from MaplesoftTM built in *Maple*)

C.3.1 Objects and Operations

When quantum optical states have a few modes and live in high dimensions, the matrices or vectors describing them very soon become intractable. To mitigate this difficulty, the procedures from **Quantavo** store and manipulate only the non-zero elements in the description of our states. Our main objects will be 2 column and 3 column matrices.

Pure States

Two column objects will describe vectors in Hilbert space of the form:

$$|\phi\rangle = \sum_{n_1, n_2, \dots, n_K}^{d-1} f(n_1, n_2, \dots, n_K) |n_1, n_2, \dots, n_K\rangle$$

Where all indices “ n_i ” range from $0 \rightarrow (d - 1)$. Pure state vectors will be encoded in 2 column matrices that we will call *trimmed vectors* containing only non-zero entries. They will only contain non-zero entries. These objects will be named as a short vector: “**vec**” and will have the following appearance:

$$\psi := \begin{bmatrix} 1 & [0, 0, 0] \\ \lambda & [1, 1, 0] \\ \lambda^2 & [2, 2, 0] \\ \lambda^3 & [3, 3, 0] \end{bmatrix}$$

For each row, the second column will be a list with the number of photons in each mode, This way $[0,1,2]$ stands for $|012\rangle$. the first column will contain the coefficient associated with this ket. The whole will describe the linear superposition of all these kets with their coefficients, therefore the above ψ describes the unnormalized quantum state:

$$|\psi\rangle = \sum_{n=0}^3 \lambda^n |n, n\rangle.$$

Mixed States

Two objects will be used to display density matrices. The first one is a square matrix, with as little zero entries as possible. This matrix will be called “**mat**” and will have for the above state the following form:

$$\rho = \begin{bmatrix} 0 & [0, 0, 0] & [1, 1, 0] & [2, 2, 0] & [3, 3, 0] \\ [0, 0, 0] & 1 & \bar{\lambda} & \bar{\lambda}^2 & \bar{\lambda}^3 \\ [1, 1, 0] & \lambda & \lambda \bar{\lambda} & \lambda \bar{\lambda}^2 & \lambda \bar{\lambda}^3 \\ [2, 2, 0] & \lambda^2 & \lambda^2 \bar{\lambda} & \lambda^2 \bar{\lambda}^2 & \lambda^2 \bar{\lambda}^3 \\ [3, 3, 0] & \lambda^3 & \lambda^3 \bar{\lambda} & \lambda^3 \bar{\lambda}^2 & \lambda^3 \bar{\lambda}^3 \end{bmatrix}$$

Another object that can also describe a density matrix will be a 3 column matrix, or trimmed density matrix. It will be named as a short column matrix: “**matcol**”. The second column will be a list with the number of photons in each mode of the ket. This way, [0,1] in the 2nd column describes $|01\rangle$. The third column will be a list with the number of photons in each mode of the bra; therefore [2,1] in the 3rd column describes $\langle 21|$. Finally The first column will have the coefficient associated with this $|\text{ket}\rangle\langle\text{bra}|$. The whole will describe the non zero elements of the density matrix. As an example consider the above state which will be:

$$\rho_{matcol} = \begin{bmatrix} 1 & [0, 0, 0] & [0, 0, 0] \\ \bar{\lambda} & [0, 0, 0] & [1, 1, 0] \\ \bar{\lambda}^2 & [0, 0, 0] & [2, 2, 0] \\ \bar{\lambda}^3 & [0, 0, 0] & [3, 3, 0] \\ \lambda & [1, 1, 0] & [0, 0, 0] \\ \lambda \bar{\lambda} & [1, 1, 0] & [1, 1, 0] \\ \lambda \bar{\lambda}^2 & [1, 1, 0] & [2, 2, 0] \\ \lambda \bar{\lambda}^3 & [1, 1, 0] & [3, 3, 0] \\ \lambda^2 & [2, 2, 0] & [0, 0, 0] \\ \lambda^2 \bar{\lambda} & [2, 2, 0] & [1, 1, 0] \\ \lambda^2 \bar{\lambda}^2 & [2, 2, 0] & [2, 2, 0] \\ \lambda^2 \bar{\lambda}^3 & [2, 2, 0] & [3, 3, 0] \\ \lambda^3 & [3, 3, 0] & [0, 0, 0] \\ \lambda^3 \bar{\lambda} & [3, 3, 0] & [1, 1, 0] \\ \lambda^3 \bar{\lambda}^2 & [3, 3, 0] & [2, 2, 0] \\ \lambda^3 \bar{\lambda}^3 & [3, 3, 0] & [3, 3, 0] \end{bmatrix}$$

Summarising, we will use mainly 3 objects: **vec**, **mat** and **matcol**.

A fourth object less commonly used is a polynomial representation of the state. It is a polynomial in the mode operators that define the state. A general state would then be:

$$\rho = \sum_{\substack{n_1, n_2, \dots, n_K \\ m_1, m_2, \dots, m_K}}^{d-1} f(n_1, n_2, \dots, n_K, m_1, \dots, m_K) a_1^{\dagger n_1} a_2^{\dagger n_2} \dots a_K^{\dagger n_K} |0\rangle\langle 0| b_1^{m_1} b_2^{m_2} \dots b_K^{m_K}$$

and its description as a **poly** object:

$$poly(\rho) = \sum_{\substack{n_1, n_2, \dots, n_K \\ m_1, m_2, \dots, m_K}}^{d-1} f(n_1, n_2, \dots, n_K, m_1, \dots, m_K) a_1^{n_1} a_2^{n_2} \dots a_K^{n_K} b_1^{m_1} b_2^{m_2} \dots b_K^{m_K}$$

(note that the commutation relations are not taken care of in **poly** objects).

C.3.2 Declare, Propagate, Measure and Ask

Quantavo contains various procedures that allow problems to be formulated as follows:

1. Declare the initial state.
2. Apply different transformations to it (Beam Splitter, Phase Shifter, Arbitrary Unitary, ...).
3. Measure certain modes (and trace out the inaccessible ones), find out probabilities.
4. Ask different questions about the properties of the state: Display, Plot, evaluate certain measures of Entanglement, etc..

Additionally, the order in which we use these procedures can be changed and the questions in item (4) can be formulated at any intermediate time. In addition, more states can be tensored or added at later times. Finally there are procedures to interconvert **mat**, **vec**, **matcol** and **poly**. Let us then give a more detailed description of these four basic steps.

Declaration

There are different ways to declare a state, all depending on its characteristics.

Pure States

For pure states of known functional form. That is, if we have a state of the form:

$$|\psi\rangle = \sum_{n_1, n_2, \dots, n_K}^d f(n_1, n_2, \dots, n_K) |n_1, n_2, \dots, n_K\rangle \quad (\text{C.1})$$

and we know explicitly $f(n_1, n_2, \dots, n_K)$ we may use the following structure:

Declare the number of modes “ K ”, the maximum number of photons “ d ”, and make a loop to declare the elements. For example:

```
d := 4;
K := 3;
V:=Matrix(d^K, 2);
```

There are now $2d^K$ elements to be specified.

```
for i from 1 to d do
  for j from 1 to d do
    for k from 1 to d do

      V[i,1]:=f(i,j,k);
      V[i,2]:=[i,j,k];

    end do;
  end do;
end do;
```

Where f is the function in eq.(C.1). We may use the procedure `deltaK(i,j)` if we need a Kronecker delta in our definition. Executing the above loop will declare a matrix \mathbf{V} of size $d^K \times 2$ that has hopefully many zero entries. To get rid of the zero entries and convert this object into a **vec** object (or trimmed vector) we will use the procedure `Trim`:

```
V1:=Trim(V);
```

Another way to proceed is to declare the object **vec** directly with an appropriate function. Consider as an example the state:

$$|\psi\rangle = \sum_{n=0}^3 \lambda^n |n, n, 0\rangle$$

In this case it is easy to declare the object **vec** directly as:

```
V1:=Matrix(4,2);
for i from 1 to 4 do
  V1[i,1]:=λ^{i-1};
  V1[i,2]:=[i-1,i-1,0];
end do;
```

However this is not always the case and if we have a pure state with no known $f(n_1, n_2, \dots, n_K)$, but we know which non-zero elements it contains, we can declare its elements one by one or declare the **vec** matrix at once. For example to declare

$$|\psi\rangle = |00\rangle + \lambda|11\rangle + \lambda^2|21\rangle$$

we can use:

```
d:=2:  
K:=2:  
V:=Matrix(3,2):  
V[1,1]:=1:  
V[1,2]:=[0,0]:  
V[2,1]:=λ:  
V[2,2]:=[1,1]:  
V[3,1]:=λ²:  
V[3,2]:=[2,1]:
```

or

```
d:=2:  
K:=2:  
V:=Matrix([ [1,[0,0]], [λ,[1,1]], [λ²,[2,1]] ]):
```

or

```
d:=2:  
K:=2:  
V:=<<(1, λ, λ²)> | <[0, 0], [1, 1], [2, 1]>>:
```

Adding `StateNormalize(V)` for normalization.

A special family of pure states are readily available in Quantavo:

Squeezed vacuum:

for a truncated, unnormalized, pure single mode squeezed vacuum state $|\phi\rangle \sim \sum_{n=0}^{d-1} \lambda^n |n\rangle$ type:

```
SqueezedVac(1, d, λ);
```

for a truncated, unnormalized, pure two mode squeezed vacuum state $|\phi\rangle \sim \sum_{n=0}^{d-1} \lambda^n |n, n\rangle$ type:

```
SqueezedVac(2, d, λ);
```

These states are given without normalization, since for finite d , $\sqrt{1 - \lambda^2}$ doesn't normalize them. Calculations and displays are easier this way.

Coherent States:

for a truncated single mode coherent state $|\phi\rangle = \sum_{n=0}^{d-1} \frac{\alpha^n}{\sqrt{n!}} |n\rangle$ type:

```
CoherentState(1, d, α);
```

The analytical normalization is rather lengthy so it is left unnormalized. Normalization can be done at a later time with `StateNormalize`.

It is also possible to tensor some vacuum modes to our state. The procedure `TensorVac(ψ, s)` effectively does the following transformation: $|\psi\rangle \longrightarrow |\psi\rangle \otimes |0\rangle^{\otimes s}$ when applied ei-

ther to a **vec**, **mat** or **matcol** (see appendix for more details).

Tensor Product:

The procedure `TensorProduct(A, ListA, B, ListB)` will make the tensor product between modes $[List_A]$ of state A and modes $[List_B]$ of state B. So for example,

```
Co:=CoherentState(1, 4, α);
Fock:=Matrix([1,[1]]);
State:=TensorProduct(Co,[1],Fock,[2]);
```

will result in the following **vec** object,

$$\text{State} = \begin{bmatrix} 1 & [0, 1] \\ \alpha & [1, 1] \\ 1/2 \alpha^2 \sqrt{2} & [2, 1] \\ 1/6 \alpha^3 \sqrt{6} & [3, 1] \end{bmatrix}$$

Mixed States

The generalized states (either pure or mixed) we are interested in can be written as:

$$\rho = \sum_{\hat{n}, \hat{m}}^{d-1} g(\hat{n}, \hat{m}) |\hat{n}\rangle \langle \hat{m}| \quad (\text{C.2})$$

where \hat{n} and \hat{m} stand for n_1, n_2, \dots, n_K and m_1, m_2, \dots, m_K respectively. The density matrix for these states is $d^K \times d^K$ dimensional which is in general too large for the computer to handle. We will therefore describe these states using **mat**, **matcol** or **poly**.

objects.

Two main strategies can be used to declare our states:

1. If our starting state is pure and will become mixed later, we can declare a pure **vec** object and then convert it into a **mat** or **matcol** when needed. All procedures to convert are named in the intuitive way: “object2object”. This way to convert **vec** into **mat** we have the procedure **vec2mat**, to convert **mat** into **matcol**, **mat2matcol** and so on for all objects and conversions. Therefore, once our pure state vector **V** has been declared we can do the following:

```
V1:=Trim(V): #eliminate non-zero entries
M1:=vec2mat(V1): #convert it to a density matrix
```

or

```
V1:=Trim(V):
M1col:=vec2matcol(M1):
```

2. if our starting state is mixed to begin with, we can declare our initial state as the object **mat** or **matcol**. To do so, if we know the functional form of $g(\hat{n}, \hat{m})$ in eq.(C.2) then we may directly declare our state. For example to declare the state:

$$\rho = \sum_{n,m=0}^4 \lambda^{n+m} |n, n, 0\rangle \langle m, m, 0|$$

We could use:

```

M1col:=Matrix(3,25):
for i from 1 to 5 do
  for j from 1 to 5 do

    M1col[i,1]:=λi+j-2;
    M1col[i,2]:=[i-1,i-1,0];
    M1col[i,3]:=[j-1,j-1,0];

  end do;
end do;

```

or

```

V1:=SqueezedVac(2, 4, lambda);
V1:=TensorVac(V1, 1);
M1col:=vec2matcol(V1);

```

or yet again, meaning $\rho = \left[\left(\sum_{n=0}^4 \lambda^n |n\rangle \right) \otimes \left(\sum_{m=0}^4 \lambda^m |m\rangle |0\rangle \right) \right] [c.c.]$

```

V1:=SqueezedVac(1, 5, lambda);
V2:=SqueezedVac(1, 5, lambda);
V2:=TensorVac(V2, 1);
V3:=TensorProduct(V1,[1],V2,[2,3]);
M1col:=vec2matcol(V3);

```

The basic conclusion is that there is no single way of declaring our state and that depending on its structure we have to find a clever way of declaring it. As a general guideline, small states without an obvious functional structure can be declared giving all elements. Medium sized pure states can be declared using a clever loop, ‘trimmed’

and transformed to **vec**, **mat** or **matcol**.

It is worth noting that when using the built-in declaration procedures such as `SqueezedVac`, `CoherentState`, `TensorVac`, `Vac`, or the inter-converting ones like `mat2matcol`, `mat2poly`, `vec2mat`, etc, the values of K and d are automatically updated. However, if the states are declared from scratch, the values of K and d should be explicitly declared. Applying to our state ρ the procedure `findKnd`(ρ) can help troubleshoot by reevaluating the value of these global variables.

Evolution

Beam Splitter

We can apply a beam splitter to a **vec** or **matcol** state **V** with the `BS` procedure:

```
V1:= BS( V , i , j ):
```

Where i , j specify which modes the beam splitter acts on. It is therefore essential to carefully label the modes of our quantum optical circuit. This will effectively do the following mode transformation:

$$\begin{pmatrix} a'_i \\ a'_j \end{pmatrix} = \begin{pmatrix} t & r \\ -r & t \end{pmatrix} \begin{pmatrix} a_i \\ a_j \end{pmatrix} \quad (\text{C.3})$$

Leaving ‘ t ’ and ‘ r ’ as unevaluated variables. If we want to use different reflectivities and transmittivities for our Beam Splitter (variables or numbers) we may use `myBS` and input ‘ $t = t_0$ ’ and ‘ $r = r_0$ ’ as follows:

```
V1:=myBS( V , i , j , t0 , r0 );
```

For the mixed state object **matcol** we may use the same procedures and the effective transformation will be:

$$\begin{pmatrix} a'_i \\ a'_j \end{pmatrix} = \begin{pmatrix} t & r \\ -r & t \end{pmatrix} \begin{pmatrix} a_i \\ a_j \end{pmatrix} \quad (\text{C.4})$$

$$\begin{pmatrix} b'_i \\ b'_j \end{pmatrix} = \begin{pmatrix} t^* & r^* \\ -r^* & t^* \end{pmatrix} \begin{pmatrix} b_i \\ b_j \end{pmatrix}. \quad (\text{C.5})$$

Further details about its use can be found in the example in section (C.4)

Phase Shifter

The phase shifter procedure (PS) can be used with mixed and pure states:

```
V1:=PS( vec/matcol , i , φ );
```

The input **vec/matcol** means that either **vec** or **matcol** objects can be given as inputs. This procedure makes the effective transformation:

$$a'_i = e^{i\phi} a_i \quad (\text{C.6})$$

$$b'_i = e^{-i\phi} b_i \quad (\text{C.7})$$

Build Unitary

If we wish to construct a unitary matrix that transforms the modes of light and describes a given linear optics (LO) circuit we may use the procedure `BuildUnitary`. Together with `UnitaryEvolution` it will evolve the state through a given LO circuit. To build a unitary consisting of Beam Splitters (BS) and Phase Shifters (PS) we will do the following:

```
U:=BuildUnitary([List]):
```

This will create a matrix \mathbf{U} of dimension $K \times K$, that later will transform the K modes. The list that is `BuildUnitary`'s input must have a precise format. It must be a list of lists. For example: `List:=[[1,2,t,r],[3,phi],[3,4,q]]`; means that first a BS with transmitivity t^2 and reflectivity r^2 will be applied to modes 1 and 2, then a PS will be applied to mode 3 and finally a BS with transmitivity q^2 and reflectivity $(1 - q^2)$ will be the last operation. If we have 4 modes, this will build the matrix

$$\begin{aligned} U &= \begin{pmatrix} t & r & 0 & 0 \\ -r & t & 0 & 0 \\ 0 & 0 & e^{i\phi}q & e^{i\phi}\sqrt{1-q^2} \\ 0 & 0 & -\sqrt{1-q^2} & q \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & q & \sqrt{1-q^2} \\ 0 & 0 & -\sqrt{1-q^2} & q \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{i\phi} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} t & r & 0 & 0 \\ -r & t & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{aligned}$$

In general, in our list of lists, lists with 4 elements, like [i, j, t, r] build BS transformations between modes i and j, lists with 3 elements like [i, j, t] build BS transformations for modes i and j such that $r = \sqrt{1 - t^2}$, and lists with two elements like [i, ϕ] build PS transformations on mode i.

UnitaryEvolution

Whether we have just built a unitary matrix with `BuildUnitary` or we have a $K \times K$ arbitrary unitary matrix to transform our modes $\{a_i\}$, we can use this procedure as follows:

```
V:= UnitaryEvolution(U, vec/matcol);
```

where U is the unitary matrix of dimension $K \times K$ and our state is described by a `vec` or a `matcol` object. This will effectively implement the mode transformation:

$$\bar{a}' = U\bar{a}$$

$$\bar{b}'^\dagger = U^\dagger\bar{b}^\dagger$$

Measurement

The Project Procedure:

If we wish to know the state after a projective measurement we may use the procedure `Project`. Depending on the inputs we give to the procedure it will do a projective measurement and return a density matrix (`matcol`) or state vector (`vec`). Below is a description of its different uses:

1. if given (`vec`₁, list, `vec`₂) and say `vec`₁ and `vec`₂ describe respectively $|\psi_1\rangle$ and

$|\psi_2\rangle$, `Project(vec1, list, vec2)` returns the **vec** (in principle unnormalized) corresponding to the expression:

$$|\psi'_2\rangle = (|\psi_1\rangle\langle\psi_1|_{list} \otimes \mathbb{I}_{rest}) |\psi_2\rangle$$

As an example consider `list=[2,3]` meaning that we want to measure modes two and three. The **vec** object that corresponds to the projector $|\psi_1\rangle\langle\psi_1|$ must therefore have kets with 2 modes.

For example, if $|\psi_2\rangle$ is

$$V2 := \begin{bmatrix} 1 & |0000\rangle \\ x & |1100\rangle \\ x^2 & |2200\rangle \end{bmatrix}$$

and $|\psi_1\rangle$ is

$$V1 := \begin{bmatrix} 1 & |10\rangle \end{bmatrix}$$

Then,

```
S := Project(V1, [2,3], V2);
```

will return:

$$S = \begin{bmatrix} x & |1100\rangle \end{bmatrix}$$

2. if given (**matcol 1**, `list`, **vec 2**) and say **matcol 1** and **vec 2** describe M_n and $|\psi_2\rangle$ then it will return the **matcol** object:

$$\rho = (M_{n_{list}} \otimes \mathbb{I}_{rest}) |\psi_2\rangle\langle\psi_2| (M_{n_{list}} \otimes \mathbb{I}_{rest})^\dagger$$

So for example, taking the same V2 as above and the Kraus Operator (or projector):

$$M := \begin{bmatrix} 1 & [1, 1] & [1, 1] \\ 1 & [0, 0] & [0, 0] \end{bmatrix}$$

Then,

```
S := Project(M, [1,2], V2);
```

will return the state:

$$S = \begin{bmatrix} 1 & [0, 0, 0, 0] & [0, 0, 0, 0] \\ \bar{x} & [0, 0, 0, 0] & [1, 1, 0, 0] \\ x & [1, 1, 0, 0] & [0, 0, 0, 0] \\ x\bar{x} & [1, 1, 0, 0] & [1, 1, 0, 0] \end{bmatrix}$$

which is the density matrix corresponding to:

$$S = \begin{bmatrix} 1 & [0, 0, 0, 0] \\ x & [1, 1, 0, 0] \end{bmatrix}$$

3. if given (**vec 1**, **list**, **matcol 2**) and say **vec 1**, **matcol 2** describe $|\psi_1\rangle$ and ρ respectively, then it will build $|\psi_1\rangle\langle\psi_1|$ and return the **matcol** object corresponding to:

$$\rho' = (|\psi_1\rangle\langle\psi_1|_{list} \otimes \mathbb{I}_{rest}) \rho (|\psi_1\rangle\langle\psi_1|_{list} \otimes \mathbb{I}_{rest})$$

4. if given (**matcol 1**, list, **matcol 2**) and say **matcol 1** and **matcol 2** describe M_n and ρ respectively, then it will return:

$$\rho' = (M_{n_{list}} \otimes \mathbb{I}_{rest}) \rho (M_{n_{list}} \otimes \mathbb{I}_{rest})^\dagger$$

(see further down for POVM measurements)

In a nutshell:

$$\begin{aligned} \text{Project}(|\psi_1\rangle, \text{list}, |\psi_2\rangle) &\rightarrow (|\psi_1\rangle\langle\psi_1|_{list} \otimes \mathbb{I}_{rest}) |\psi_2\rangle \\ \text{Project}(M_n, \text{list}, |\psi_2\rangle) &\rightarrow (M_{n_{list}} \otimes \mathbb{I}_{rest}) |\psi_2\rangle\langle\psi_2| (M_{n_{list}} \otimes \mathbb{I}_{rest})^\dagger \\ \text{Project}(|\psi_1\rangle, \text{list}, \rho) &\rightarrow (|\psi_1\rangle\langle\psi_1|_{list} \otimes \mathbb{I}_{rest}) \rho (|\psi_1\rangle\langle\psi_1|_{list} \otimes \mathbb{I}_{rest}) \\ \text{Project}(M_n, \text{list}, \rho) &\rightarrow (M_{n_{list}} \otimes \mathbb{I}_{rest}) \rho (M_{n_{list}} \otimes \mathbb{I}_{rest})^\dagger \end{aligned}$$

If we encounter a destructive measurement, it is possible to trace out the measured modes with the `Traceout` procedure (cf. appendix). Also `POVMresult` traces out the measured modes (see next section). We may otherwise calculate the full trace of a **mat** or **matcol** object with `StateTrace` or multiply **vec** and **matcol** in different orders thanks to `StateMultiply`.

The Probability Procedure:

This procedure will calculate the probability of a measurement result or an expected value. It considers the same cases and objects as the above `Project` procedure. It uses the definition,

$$P = \frac{\text{Tr} \{E_n \rho\}}{\text{Tr} \{\rho\}}$$

and assumes that $\{E_n\}$ safisfy $E_n \geq 0$ and $\sum_n E_n = \mathbb{I}$ to calculate probabilities. Therefore, one should verify that these conditions hold for the matrices describing E_n in order to obtain meaningful probabilities. Below we show more details for different inputs:

$$\text{Probability}(|\psi_1\rangle, \text{list}, |\psi_2\rangle) \rightarrow \frac{\text{Tr} \{(|\psi_1\rangle\langle\psi_1|_{list} \otimes \mathbb{I}_{rest}) |\psi_2\rangle\langle\psi_2|\}}{\text{Tr} \{|\psi_2\rangle\langle\psi_2|\}}$$

where $(|\psi_1\rangle, |\psi_2\rangle)$ are converted to **matcol** objects in an intermediate step. One should pay attention to the choice of the projection operator $|\psi_1\rangle\langle\psi_1|$. If it is not nomalized it can give unphysical values for the probability.

Now for a given POVM or Projector,

$$\text{Probability}(E_n, \text{list}, |\psi_2\rangle) \rightarrow \frac{\text{Tr} \{(E_n \text{list} \otimes \mathbb{I}_{rest}) |\psi_2\rangle\langle\psi_2|\}}{\text{Tr} \{|\psi_2\rangle\langle\psi_2|\}}$$

Or given a **vec** and a density operator **matcol**:

$$\text{Probability}(|\psi_1\rangle, \text{list}, \rho) \rightarrow \frac{\text{Tr} \{(|\psi_1\rangle\langle\psi_1|_{list} \otimes \mathbb{I}_{rest}) \rho\}}{\text{Tr} \{\rho\}}$$

where $|\psi_1\rangle$ has been converted to a **matcol** object.

And finally for a POVM and a density operator:

$$\text{Probability}(E_n, \text{list}, \rho) \rightarrow \frac{\text{Tr} \{(E_n \text{list} \otimes \mathbb{I}_{rest}) \rho\}}{\text{Tr} \{\rho\}}$$

POVM measurements

Quantavo possesses a procedure to describe POVM measurements. If our state before the measurement is $|\psi\rangle$ or ρ , and the POVM elements are described by the set $\{E_m\}$, satisfying $\sum_m E_m = \mathbb{I}$ and $E_m \geq 0$. Then the state after the measurement will be:

$$\rho' = \text{Tr}_{i,j,\dots,k} \{E_m \rho\}$$

or

$$\rho' = \text{Tr}_{i,j,\dots,k} \{E_m |\psi\rangle\langle\psi|\}$$

assuming $\text{Tr} \{\rho\} = 1$ or $\text{Tr} \{|\psi\rangle\langle\psi|\} = 1$.

This will be implemented by the procedure `POVMresult` which will take as inputs,

`POVMresult(matcol, List, matcol/vec)`

and implement the operation,

$$\text{POVMresult}(E_m, List, \rho) = \frac{\text{Tr}_{List} \{(E_m List \otimes \mathbb{I}_{rest}) \rho\}}{\text{Tr} \{\rho\}}$$

where ‘rest’ are all the indexes not included in ‘List’.

Declaring POVMs

The `matcol` objects that represent POVM operators need to be declared. One option is to declare them as standard states and convert them to `matcol`. Since avalanche photo diode detectors (APDs) are a standard tool in quantum optics, an interactive tool to de-

clare them is also provided. Executing,

```
APD();
```

Will bring up an interactive menu, where we can choose if we want the $|0\rangle\langle 0|$ or $\mathbb{I} - |0\rangle\langle 0|$ event, how many photons we will consider and what loss should be added to it. For example choosing the input **(0, r, 4)** will return,

$$\pi_0 = \begin{bmatrix} 1 & [0] & [0] \\ r^2 & [1] & [1] \\ r^4 & [2] & [2] \\ r^6 & [3] & [3] \\ r^8 & [4] & [4] \end{bmatrix}$$

and choosing **(1, r, 4)** will return,

$$\pi_1 = \begin{bmatrix} 1 - r^2 & [1] & [1] \\ 1 - r^4 & [2] & [2] \\ 1 - r^6 & [3] & [3] \\ 1 - r^8 & [4] & [4] \end{bmatrix}$$

recovering the expected $\pi_0 + \pi_1 = \mathbb{I}$. Note the convention for the BS in front of the detector for which $r = 0$ is a perfect detector and $r^2 + t^2 = 1$.

These are therefore the main tools to describe and simulate quantum measurements in this framework.

State Properties

Some questions that will interest us will concern hermiticity, normalization and entanglement measures. So far, Quantavo has the following useful procedures:

`IsHermitian` to check for selfadjointness,

`IsNormalized` to check for normalization,

`StateNormalize` to normalize `mat`, `matcol` or `vec`

`StatePartialTranspose` to partial transpose ρ^Γ

`Negativity` to calculate the negativity as the sum of all negative eigenvalues of the partial transposed density matrix.

`LogNegativity` to calculate the Logarithmic Negativity.

`StateApprox` to do symbolic or numeric approximations transforming our `matcol` or `vec` state.

For practical examples on how to use them one can refer to the following section. Otherwise, a detailed dictionary of procedures can be found in the appendix.

C.4 Practical Example

C.4.1 Squeezed state photon subtraction

Our initial state is a pure two mode squeezed state that can be described by $|\psi_\lambda\rangle = \sqrt{1 - \lambda^2} \sum_{n=0}^{\infty} \lambda^n |n, n\rangle$. We would like to declare it and propagate it through the circuit presented in fig. C.1. We observe that our initial state has three modes, one of which is a vacuum mode. We can work out as an example the state with up to 4 photons:

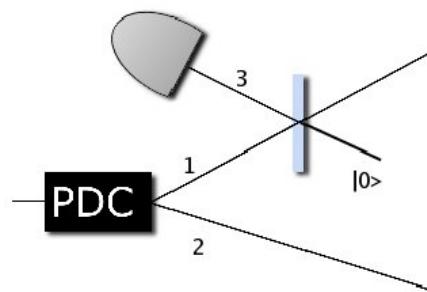


Figure C.1: Setup of the photon subtraction.

```
# Create a truncated two mode squeezed vacuum state
V := SqueezedVac(2, 5, lambda);

# Then add the vacuum mode:
V := TensorVac(V,1);
```

This will output state:

$$V := \begin{bmatrix} 1 & [0, 0, 0] \\ \lambda & [1, 1, 0] \\ \lambda^2 & [2, 2, 0] \\ \lambda^3 & [3, 3, 0] \\ \lambda^4 & [4, 4, 0] \end{bmatrix}$$

We then apply the corresponding beam splitter transformation (if “d” changes from the BS transformation, it is automatically recalculated after the BS operation and reset to its new value. In this case it doesn’t change).

```
V1:=BS(V,1,3);
```

which returns, if displayed with “Display State”, that is `Dstate(V1)`, the following output:

$$V1 = \begin{bmatrix} 1 & "|000 >" \\ \lambda t & "|110 >" \\ \lambda^2 t^2 & "|220 >" \\ \lambda^3 t^3 & "|330 >" \\ \lambda^4 t^4 & "|440 >" \\ \lambda r & "|011 >" \\ \sqrt{2}\lambda^2 tr & "|121 >" \\ \sqrt{3}\lambda^3 t^2 r & "|231 >" \\ 2\lambda^4 t^3 r & "|341 >" \\ \lambda^2 r^2 & "|022 >" \\ \sqrt{3}\lambda^3 t r^2 & "|132 >" \\ \sqrt{6}\lambda^4 t^2 r^2 & "|242 >" \\ \lambda^3 r^3 & "|033 >" \\ 2\lambda^4 t r^3 & "|143 >" \\ \lambda^4 r^4 & "|044 >" \end{bmatrix}$$

Measurements:

First we will consider the case of a perfect photon number resolving detector. The projector $|1\rangle\langle 1|$ can simply be introduced as:

```
Proj:=Matrix([1,[1]]);
```

And the state after the measurement will be obtained with:

```
V2:=Project(Proj,[3],V1);
```

returning:

$$V2 = \begin{bmatrix} \lambda r & [0, 1, 1] \\ \sqrt{2}\lambda^2 tr & [1, 2, 1] \\ \sqrt{3}\lambda^3 t^2 r & [2, 3, 1] \\ 2\lambda^4 t^3 r & [3, 4, 1] \end{bmatrix}$$

We may now trace-out the measured mode. Tracing out can then be done as follows:

```
M1:=Traceout(V2,3);
```

We may also simulate a measurement with an avalanche photo-diode described by the Kraus Operator: $\hat{O} = \mathbb{I} - |0\rangle\langle 0|$. That is, if the state “V1” represents the state ρ , we want to find the state ρ' resulting from the measurement:

$$\rho' \sim (\hat{O}_3 \otimes \mathbb{I}_{1,2}) \rho (\hat{O}_3 \otimes \mathbb{I}_{1,2})^\dagger$$

For this simple example we have at most 4 photons so we will approximate $\hat{O} \simeq |1\rangle\langle 1| + |2\rangle\langle 2| + |3\rangle\langle 3| + |4\rangle\langle 4|$. Therefore to measure mode 3 we construct the associ-

ated POVM which can then be expressed as a **matcol** object:

```
POVM := Matrix(4, 3);
```

```
for i from 1 to 4 do
    POVM[i,1]:=1;
    POVM[i,2]:=[i];
    POVM[i,3]:=[i];
od;
```

And the state after a ‘click’ in the detector (tracing out this inaccessible mode) will be:

```
M2:=Project(POVM,[3],V1);
M3:=Traceout(M2,3);
```

or directly

```
M3:=POVMresult(POVM,[3],V1);
```

Approximations:

M3 has 30 coefficients. We may want to use the approximation procedures to simplify our calculations. If for example we know that $r \ll 1$ and $\lambda \ll 1$ we may wish to delete all terms in the density matrix for which the coefficients containing $r^n \lambda^m$ sat-

isfy $n + m > 7$ (therefore getting rid of all the small terms up to the chosen order). The procedure `StateApprox` can be used for this purpose. This way,

```
M4:=RealDomain:-simplify(M3); # assume real variables for
simplicity
M5:=StateApprox(M4,[lambda,r], 7);
```

Will deliver

$$M5 = \begin{bmatrix} \lambda^2 r^2 & [0, 1] & [0, 1] \\ \sqrt{2}\lambda^3 t r^2 & [0, 1] & [1, 2] \\ \lambda^4 t^2 r^2 \sqrt{3} & [0, 1] & [2, 3] \\ 2\lambda^5 t^3 r^2 & [0, 1] & [3, 4] \\ \sqrt{2}\lambda^3 t r^2 & [1, 2] & [0, 1] \\ 2\lambda^4 t^2 r^2 & [1, 2] & [1, 2] \\ \sqrt{6}\lambda^5 t^3 r^2 & [1, 2] & [2, 3] \\ \lambda^4 t^2 r^2 \sqrt{3} & [2, 3] & [0, 1] \\ \sqrt{6}\lambda^5 t^3 r^2 & [2, 3] & [1, 2] \\ 2\lambda^5 t^3 r^2 & [3, 4] & [0, 1] \end{bmatrix}$$

See the dictionary in the appendix for details on how to use the `StateApprox` procedure.

Entanglement

We may now compute different properties of this state as for example the Negativity. We take the state after the detection of exactly one photon and obtain:

```
Neg:=simplify(Negativity(M1)) assuming t::positive, r::positive,
lambda::positive;
```

which will return:

$$Neg = \frac{\lambda t (2 \lambda^2 t^2 + \sqrt{2} + \sqrt{3} \lambda t + \sqrt{3} \lambda^2 t^2 \sqrt{2} + 2 \lambda^3 t^3 \sqrt{2} + 2 \lambda^4 t^4 \sqrt{3})}{4 \lambda^6 t^6 + 3 \lambda^4 t^4 + 2 \lambda^2 t^2 + 1}$$

The eigenvalues can be hard to find if our state has symbolic entries with complex conjugation. A previous simplification (assuming real variables) can help, but other solutions are possible. This explicit expression allows us to plot different values of the negativity for various ranges of parameters:

```
plot3d(Neg(\lambda,t,\sqrt{1 - t^2}), \lambda=0..1,t=0..1, axes=boxed);
```

will display the plot in fig C.2.:

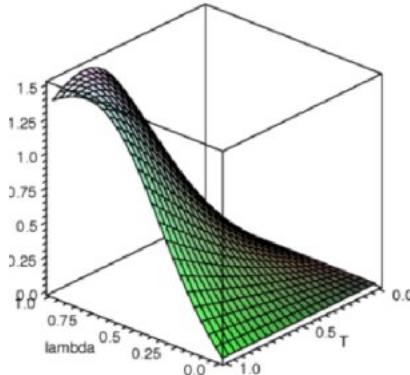


Figure C.2: Logarithmic Negativity (t, λ).

Further details, examples and documentation can be found on the website of the Imperial College London quantum information group,
<http://www.imperial.ac.uk/quantuminformation/research/downloads>

Acknowledgements

This project was supported by EPSRC grant EP/C546237/1 and by the Integrated Project Qubit Applications (QAP) supported by the IST directorate as Contract Number 015848. The author would also like to thank Alain Le Stang and the members of Mapleprimes.com for useful suggestions.

.1 Dictionary of Procedures:

A quick reference list is provided and an alphabetically ordered dictionary follows.

Objects	Appearance	Declaration
vec	$\psi = \begin{bmatrix} 1 & [0, 0, 0] \\ \lambda & [1, 1, 0] \\ \lambda^2 & [2, 2, 0] \\ \lambda^3 & [3, 3, 0] \end{bmatrix}$	<pre> vec:=CoherentState(K,d,α) vec:=SqueezedVac(K,d,λ) vac:=Vac(K); vec1:=TensorVac(vec,m) vec1:=TensorProduct(V,[1,2],W,[3,4]) vec:=Trim(V) </pre>
mat	$\rho = \begin{bmatrix} 0 & [0, 0, 0] & [1, 1, 0] \\ [0, 0, 0] & 1 & \bar{\lambda} \\ [1, 1, 0] & \lambda & \lambda \bar{\lambda} \end{bmatrix}$	<pre> mat:=vec2mat(vec) mat:=matcol2mat(matcol) or direct declaration of the Matrix </pre>
matcol	$\rho_{matcol} = \begin{bmatrix} 1 & [0, 0, 0] & [0, 0, 0] \\ \bar{\lambda} & [0, 0, 0] & [1, 1, 0] \\ \lambda & [1, 1, 0] & [0, 0, 0] \\ \lambda \bar{\lambda} & [1, 1, 0] & [1, 1, 0] \\ \lambda^2 \bar{\lambda}^2 & [2, 2, 0] & [2, 2, 0] \end{bmatrix}$	<pre> matcol:=vec2matcol(vec) matcol:=mat2matcol(mat) or direct declaration of the 3 column Matrix </pre>

$$\begin{aligned}
 \text{CoherentState}(K, d, \alpha) &\sim \sum_{n=0}^{d-1} \frac{\alpha^n}{\sqrt{n!}} |n\rangle^{\otimes K} \\
 \text{SqueezedVac}(K, d, \lambda) &\sim \sum_{n=0}^{d-1} \lambda^n |n\rangle^{\otimes K} \\
 \text{ready made states are:} \\
 \text{TensorVac}(\text{vec}, m) &: |\phi\rangle \rightarrow |\phi\rangle \otimes |0\rangle^{\otimes m}
 \end{aligned}$$

$$\text{IdentityState}(d, K) \sim \mathbb{I}_{d^K \times d^K}$$

A

APD(0/1,r,N): This procedure is interactive and is called by executing,

APD();

It declares the POVM describing a lossy avalanche photo diode detector (APD). The input “0” or “1” will select between the *no-click* or *click* events respectively. “r” will be the amplitude r of the reflectivity $R = r^2$ of the BS in front of the detector characterizing its loss, and N will be the maximum number of photons. For example choosing the input **(0, r, 4)** will return,

$$\pi_0 = \begin{bmatrix} 1 & [0] & [0] \\ r^2 & [1] & [1] \\ r^4 & [2] & [2] \\ r^6 & [3] & [3] \\ r^8 & [4] & [4] \end{bmatrix}$$

and choosing **(1, r, 4)** will return,

$$\pi_1 = \begin{bmatrix} 1 - r^2 & [1] & [1] \\ 1 - r^4 & [2] & [2] \\ 1 - r^6 & [3] & [3] \\ 1 - r^8 & [4] & [4] \end{bmatrix}$$

recovering the expected $\pi_0 + \pi_1 = \mathbb{I}$.

[Input: ($\{0, 1\}$, $r \in [0, 1]$, positive integer), Output: Matrix, Calls: Quantavo, LinearAlgebra]

B

BS(vec/matcol, i, j): This will effectuate the Beam Splitter transformation:

$$\begin{pmatrix} a'_i \\ a'_j \end{pmatrix} = \begin{pmatrix} t & r \\ -r & t \end{pmatrix} \begin{pmatrix} a_i \\ a_j \end{pmatrix} \quad (8)$$

$$\begin{pmatrix} b'_i \\ b'_j \end{pmatrix} = \begin{pmatrix} t^* & r^* \\ -r^* & t^* \end{pmatrix} \begin{pmatrix} b_i \\ b_j \end{pmatrix}. \quad (9)$$

for the chosen modes. That is eq. (8) for objects of type **vec** and eq. (8) and (9) for objects of type **matcol**. The value of ‘ d ’ and ‘ K ’ are evaluated and reset to the actual value after this operation.

[Input: (Matrix, integer, integer), Output: Matrix, Calls: Quantavo, LinearAlgebra, poly2matcol, matcol2poly, matcolBS, vecBS]

BuildUnitary(list of lists): This procedure will build a $K \times K$ unitary matrix corresponding to a linear optics circuit consisting of beam splitters (BS) and phase shifters (PS). To do so a list of lists must be provided. The lists inside the list can have 2, 3, or 4 elements, and should be in the same order as we want to apply those transformations in the circuit. Lists with 2 elements will be considered PS and lists with 3 or 4 elements as BS:

The list $[i, \xi]$ will implement a PS on mode i .

The list $[i, j, t]$ will implement a BS on modes i and j with transmittivity $t^2 = T$ and reflectivity $R = 1 - T = 1 - t^2$.

The list [i,j,t,r] will implement a BS on modes i and j with transmittivity $t^2 = T$ and reflectivity $r^2 = R$.

To implement the transformations one after another we can give for example the input

list = [[i_1, ξ], [i_2, j_2, t], [i_3, j_3, t, r], [i_3, ϕ]].

[Input: List, Output: Matrix, Calls: Quantavo, LinearAlgebra]

C

CoherentState($\mathbf{m}, \mathbf{d}, \alpha$): This procedure builds an object of type **vec** that describes the state $|\phi\rangle = \sum_{n=0}^{d-1} \frac{\alpha^n}{\sqrt{n!}} |n\rangle^{\otimes m}$. It is not normalized

[Input: (whole number, whole number), Output: Matrix, Calls: Quantavo, LinearAlgebra]

D

DP(Matrix**, **Matrix**)**: This procedure makes the Kronecker/Direct/Tensor product between any two matrices. As long as it is declared as a matrix, it can also make the Kronecker product between vectors and matrices. The vector, however would have to be declared as a Matrix. For example as V := Matrix([[1], [2], [0], [a]]).

[Input: Matrix, Output: Matrix, Calls: Quantavo, LinearAlgebra]

DeltaK(i**, **j**)**: This procedure takes the Kronecker Delta from two inputs. It works both symbolically and numerically. For example:

deltaK(1,1)=1, deltaK(1,2)=0,

deltaK(a,a)=1, (even if a is not specified).

deltaK(a,b); will return Quantavo:-deltaK(a, b)

However, if we execute “a:=b;” and evaluate it again it will be 1

[Input: (string or number, string or number), Output: 0,1, unevaluated string, Calls: Quantavo]

Dbra(List**)**: This is a Display procedure. When given a list which stands for a bra, for example List =[0,1,3] it builds a bra for display and returns $\langle 013|$.

[Input: List, Output: ⟨ string|, Calls: Quantavo].

Dbraket(List, List): This is a Display procedure. When given two lists which stand for a ket and a bra it returns a $|ket\rangle\langle bra|$. for example IList=[0,1,1] and JList =[0,1,3] then Dbraket(IList,JList)= $|011\rangle\langle 013|$.

[Input: (List, List), Output: |string⟩⟨ string|, Calls: Quantavo]

Dket(List): This is a Display procedure: When given a list which stands for a ket, for example List =[0,1,3] it builds a ket for display and returns $|013\rangle$.

[Input: List, Output: | string⟩, Calls: Quantavo]

Dstate(vec/mat/matcol): This is a Display procedure. When given either a **mat**, **vec** or **matcol** object it transforms the lists of modes into *bras* and *kets*. For example if it is given the object **mat**:

$$\rho = \begin{bmatrix} 0 & [0, 0, 0, 0] & [1, 1, 0, 0] \\ [0, 0, 0, 0] & 1 & \bar{\lambda} \\ [1, 1, 0, 0] & \lambda & \lambda \bar{\lambda} \end{bmatrix}$$

it will display bras and kets in the following way:

$$Dstate(\rho) = \begin{bmatrix} 0 & < 0000| & < 1100| \\ |0000 > & 1 & \bar{\lambda} \\ |1100 > & \lambda & \lambda \bar{\lambda} \end{bmatrix}$$

[Input: Matrix, Output: Matrix, Calls: Quantavo, LinearAlgebra]

E

Energy(vec/matcol): This will output the expected value of the energy of our state

defined as,

$$\langle \hat{E} \rangle = \langle \psi | \hat{E} | \psi \rangle$$

with

$$\langle \hat{E} \rangle = \hbar\nu \operatorname{Tr} \left\{ (a^\dagger a + 1/2)\rho \right\}$$

or for multipartite states,

$$\langle \hat{E} \rangle = \hbar\nu \operatorname{Tr} \left\{ (\hat{N}_1 + \hat{N}_2 + \dots + \hat{N}_K + K/2)\rho \right\}$$

[Input: Matrix, Output: Number or Analytic Expression, Calls: Quantavo, LinearAlgebra]

Entropy(vec/matcol): This will output the entropy of our state defined as,

$$S = \sum_i \lambda_i \log_2(\lambda_i)$$

where λ_i are the eigenvalues. The procedure transforms the objects **vec** or **matcol** into a **mat** density matrix and then finds the eigenvalues.

[Input: Matrix, Output: Number or Analytic Expression, Calls: Quantavo, LinearAlgebra]

EvalState(vec/matcol): This simply applies the Maple function `evalf` to the first column of our **vec** or **matcol** objects.

[Input: Matrix, Output: Number or Analytic Expression, Calls: Quantavo, LinearAlgebra]

F

findKnd(vec/matcol/mat): This will search through the given state to find the number of modes K and the dimension of the modes d . It updates these global variables with the K and d found and displays them as output

[Input: Matrix, Output: (K,d), Calls: Quantavo, LinearAlgebra] `findKnd`

I

IdentityState(Nr of photons, Nr of modes): This will declare an identity matrix represented as a **matcol** object. For example, for (2,2) as input we will obtain

$$\mathbb{I} = \begin{bmatrix} 1 & [0, 0] & [0, 0] \\ 1 & [0, 1] & [0, 1] \\ 1 & [0, 2] & [0, 2] \\ 1 & [1, 0] & [1, 0] \\ 1 & [1, 1] & [1, 1] \\ 1 & [1, 2] & [1, 2] \\ 1 & [2, 0] & [2, 0] \\ 1 & [2, 1] & [2, 1] \\ 1 & [2, 2] & [2, 2] \end{bmatrix}$$

[Input: (integer,integer), Output: Matrix, Calls: Quantavo, LinearAlgebra]

indexstate(vec/matcol): This procedure will transform the last two columns of a **vec** or **matcol** object. Each list containing the number of photons in each mode will be translated into a natural number with the procedure **VectorRow(List,d)**. This number will indicate the order in which the modes are ordered. In effect it does a number basis change from a d -base to a 10-base. For example,

$$\begin{bmatrix} 1 & [0, 0] \\ \xi & [1, 1] \\ \xi^2 & [2, 2] \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & 1 \\ \xi & 5 \\ \xi^2 & 9 \end{bmatrix}$$

[Input: 3 column Matrix, Output: 3 column Matrix, Calls: Quantavo, LinearAlgebra]

ISHermitian(matcol/mat): This procedure verifies if its input describes a Hermitian state. Therefore if the density matrix it represents verifies $\rho^+ = \rho$. If it is Hermitian the value returned will be 0, otherwise it will be 1. Maple might not recognize products of conju-

gated complex variables as equal, so one has to make sure they are simplified.

[Input: Matrix, Output: 0 or 1 and printed answer, Calls: Quantavo, LinearAlgebra]

IsNormalized(vec/matcol/mat): This procedure checks if the considered object is normalized. Therefore for **vec** if $\langle \psi | \psi \rangle = 1$, for **matcol** and **mat** if $\text{Tr}(\rho)=1$. If the state is not normalized it will return its norm or trace.

[Input: Matrix, Output: printed answer, Calls: Quantavo, LinearAlgebra]

L

LogNegativity(vec/matcol/mat): This procedure evaluates the Logarithmic negativity according to:

$$\text{LogNegativity}(M) := \log_2(2 * \text{Negativity}(M) + 1)$$

See **Negativity** for further details.

[Input: Matrix, Output: Expression or number, Calls: Quantavo, Negativity]

M

matcol2mat(matcol): This translation procedure takes an object of type **matcol** and converts it into an object of type **mat**. To do so, it sorts the **matcol** object, counts the number of different bras and kets needed and constructs the hermitian matrix **mat** with the least number of zeros containing all the elements of **matcol**

[Input: Matrix, Output: Matrix, Calls: Quantavo, LinearAlgebra, StateSort]

matcol2poly(matcol): This translation procedure takes an object of type **matcol** and converts it into an object of type **poly**. To do so it looks at the $|ket\rangle$ and $\langle bra|$ of each row of the Matrix and builds the associated monomial with its corresponding coefficient.

[Input: Matrix, Output: Polynomial, Calls: Quantavo, LinearAlgebra]

mat2matcol(mat): This translation procedure takes an object of type **mat** and con-

verts it into an object of type **matcol**

[Input: Matrix, Output: Matrix, Calls: Quantavo, LinearAlgebra, Trim]

mat2poly(mat): This translation procedure takes an object of type **mat** and converts it into an object of type **poly**

[Input: Matrix, Output: polynomial, Calls: Quantavo, LinearAlgebra]

modesmatcol(matcol with numbers): This procedure is the inverse of the procedure **indexmatcol**. It will transform a 3 column matrix, that has numbers in the 2nd and 3rd column to one that has the equivalent modes (lists with the number of photons)

[Input: 3 column Matrix, Output: 3 column Matrix, Calls: Quantavo, LinearAlgebra]

myBS(matcol/vec, i, j, t, r): This procedure is the same as **BS**, but it gives the user the option to choose the transmittivity $T = t^2$ and reflectivity $R = r^2$ of the beam splitter.

[Input: Matrix, Output: Matrix, Calls: Quantavo, LinearAlgebra, poly2matcol, matcol2poly, mymatcolBS, myvecBS]

N

Negativity(vec/matcol/mat): This procedure finds the eigenvalues of the partial transposed density matrix, and calculates the sum of all the corresponding non-negative eigenvalues (it also divides by the Trace of the state in case it is not normalized):

$$\text{Negativity} = \frac{1}{Tr(\rho)} \sum_i \frac{|\lambda_i| - \lambda_i}{2}$$

where λ_i are the eigenvalues of the partially transposed density matrix

[Input: Matrix, Output: Expression or number, Calls: Quantavo, LinearAlgebra, StatePartialTranspose]

P

PlotState(vec/matcol/mat, w, h): The coefficients of our states need to be real numbers before we can plot the state vector or density operator.

The procedure will plot a **vec** object as a bar diagram. In the abscissa, all possible kets between $|0, 0, \dots, 0\rangle$ and $|(d-1), (d-1), \dots, (d-1)\rangle$ are labeled from 1 to d^K . For example,

$$\begin{bmatrix} 1.0 & |0000\rangle \\ 0.5 & |1100\rangle \\ 0.25 & |2200\rangle \end{bmatrix}$$

with $K=2$ and $d=3$, will be displayed as shown in Fig. 3:

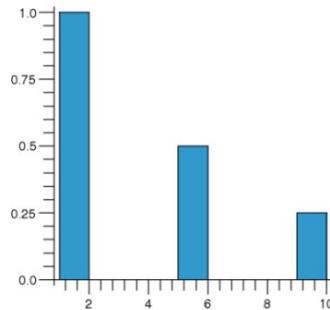


Figure 3: Plot for a pure state (**vec** object).

To plot a **matcol** object, the width w and height h of the bars in the bar diagram must be given as inputs ($w=0.5$ is the maximum width for the columns not to overlap). A 3D plot will be the output. Bras and kets will also be labeled from 1 to d^K . The density matrix for the above state ($w=0.5$, $h=10$) is shown in Fig. 4:

[Input: (Matrix, width, height), Output: , Calls: plots, geom3d, Quantavo, LinearAlgebra]

poly2matcol(poly): This procedure transforms a polynomial of the modes

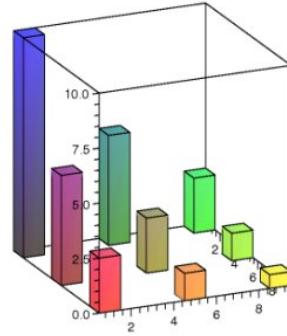


Figure 4: Plot for a density matrix (**matcol/mat** object).

Poly = $\sum_{m_1, m_2, \dots, m_K}^d \sum_{n_1, n_2, \dots, n_K}^d f(\hat{n}, \hat{m}) \prod_{j=1}^K a_j^{n_j} b_j^{m_j}$ into an object of type **matcol**
 [Input: polynomial, Output: Matrix, Calls: Quantavo, LinearAlgebra]

poly2vec(poly): This procedure transforms a polynomial of the modes

Poly = $\sum_{n_1, n_2, \dots, n_K}^d f(\hat{n}) \prod_{j=1}^K a_j^{n_j}$ into an object of type **vec**

[Input: polynomial, Output: Matrix, Calls: Quantavo, LinearAlgebra]

POVMresult(matcol, List, vec/matcol): If our state before the measurement is $|\psi\rangle$ or ρ , and the POVM elements are described by the set $\{E_m\}$ then the unnormalized state after the measurement will be:

$\text{POVMresult}(E_m, List, \rho) = \text{Tr}_{List} \{(E_m List \otimes \mathbb{I}_{rest}) \rho\}$

where ‘rest’ are all the indexes not included in ‘List’.

[Input: Matrix, List, Matrix, Output: Matrix, Calls: Quantavo, LinearAlgebra]

Probability(vec/matcol, List, vec/matcol): This procedure will calculate the probability of a measurement result or an expected value. It considers the same cases and objects as the **Project** procedure. It uses the definition,

$$P = \frac{\text{Tr} \{E_n \rho\}}{\text{Tr} \{\rho\}}$$

and assumes $E_n \geq 0$ and $\sum_n E_n = \mathbb{I}$ to calculate probabilities. Therefore, one should verify that these conditions hold for the matrices describing E_n in order to obtain meaningful probabilities. Below we show more details for different inputs.

$$\text{Probability}(|\psi_1\rangle, \text{list}, |\psi_2\rangle) \rightarrow \frac{\text{Tr} \{ (|\psi_1\rangle\langle\psi_1|_{list} \otimes \mathbb{I}_{rest}) |\psi_2\rangle\langle\psi_2| \}}{\text{Tr} \{ |\psi_2\rangle\langle\psi_2| \}}$$

where $(|\psi_1\rangle, |\psi_2\rangle)$ are converted to **matcol** objects in an intermediate step. One should pay attention to the choice of the projection operator $|\psi_1\rangle\langle\psi_1|$. If it is not normalized it can give unphysical values for the probability.

Now for a given POVM or Projector,

$$\text{Probability}(E_n, \text{list}, |\psi_2\rangle) \rightarrow \frac{\text{Tr} \{ E_n |\psi_2\rangle\langle\psi_2| \}}{\text{Tr} \{ |\psi_2\rangle\langle\psi_2| \}}$$

where E_n is assumed to be a **matcol** object.

Or given a **vec** and a density operator **matcol**:

$$\text{Probability}(|\psi_1\rangle, \text{list}, \rho) \rightarrow \frac{\text{Tr} \{ (|\psi_1\rangle\langle\psi_1|_{list} \otimes \mathbb{I}_{rest}) \rho \}}{\text{Tr} \{ \rho \}}$$

where $|\psi_1\rangle$ has been converted to a **matcol** object.

And finally for a POVM and a density operator:

$$\text{Probability}(E_n, \text{list}, \rho) \rightarrow \frac{\text{Tr} \{ (E_n \otimes \mathbb{I}_{rest}) \rho \}}{\text{Tr} \{ \rho \}}$$

[Input: (Matrix, list, Matrix), Output: Expression or number, Calls: Quantavo, LinearAlgebra, StateTrace, StateNormalize, vec2matcol]

Project(vec/matcol, list, vec/matcol): This procedure returns the state after a measurement in the following cases:

1. if given (**vec**₁, list, **vec**₂) and say **vec**₁ and **vec**₂ describe respectively $|\psi_1\rangle$ and $|\psi_2\rangle$, **Project(vec**₁, list, **vec**₂) returns the **vec** (in principle unnormalized) corresponding to the expression:

$$|\psi'_2\rangle = (|\psi_1\rangle\langle\psi_1|_{list} \otimes \mathbb{I}_{rest}) |\psi_2\rangle$$

Note that if for example list=[1,2,3] it means that we want to measure modes one, two and three. The **vec** object that corresponds to $|\psi_1\rangle$ must therefore have kets with 3 modes.

For example, if $|\psi_2\rangle$ is

$$V2 := \begin{bmatrix} 1 & |0000\rangle \\ x & |1100\rangle \\ x^2 & |2200\rangle \end{bmatrix}$$

and $|\psi_1\rangle$ is

$$V1 := \begin{bmatrix} 1 & |10\rangle \end{bmatrix}$$

Then,

```
S := Project(V1, [2,3], V2);
```

will return:

$$S = \begin{bmatrix} x & |1100\rangle \end{bmatrix}$$

2. if given (**matcol 1**, list, **vec 2**) and say **matcol 1** and **vec 2** describe M_n and $|\psi_2\rangle$ then it will return the **matcol** object:

$$\rho = (M_{n_{list}} \otimes \mathbb{I}_{rest}) |\psi_2\rangle\langle\psi_2| (M_{n_{list}} \otimes \mathbb{I}_{rest})^\dagger$$

So for example, taking the same V2 as above and the POVM:

$$M := \begin{bmatrix} 1 & [0, 0] & [1, 1] \\ 1 & [1, 1] & [0, 0] \\ 1 & [1, 1] & [1, 1] \end{bmatrix}$$

Then,

```
S := Project(M, [1,2], V2);
```

will return the state:

$$S = \begin{bmatrix} x\bar{x} & [0, 0, 0, 0] & [0, 0, 0, 0] \\ x + x\bar{x} & [0, 0, 0, 0] & [1, 1, 0, 0] \\ x\bar{x} + \bar{x} & [1, 1, 0, 0] & [0, 0, 0, 0] \\ x + x\bar{x} + \bar{x} + 1 & [1, 1, 0, 0] & [1, 1, 0, 0] \end{bmatrix}$$

which is the mixed state that corresponds to:

$$S = \begin{bmatrix} x & [0, 0, 0, 0] \\ x + 1 & [1, 1, 0, 0] \end{bmatrix}$$

3. if given (**vec 1**, list, **matcol 2**) and say **vec 1**, **matcol 2** describe $|\psi_1\rangle$ and ρ respectively,

then it will build $|\psi_1\rangle\langle\psi_1|$ and return the **matcol** object corresponding to:

$$\rho' = (|\psi_1\rangle\langle\psi_1|_{list} \otimes \mathbb{I}_{rest}) \rho (|\psi_1\rangle\langle\psi_1|_{list} \otimes \mathbb{I}_{rest})$$

4. if given (**matcol 1**, list, **matcol 2**) and say **matcol 1** and **matcol 2** describe M_n and ρ respectively, then it will return:

$$\rho' = (M_{n_{list}} \otimes \mathbb{I}_{rest}) \rho (M_{n_{list}} \otimes \mathbb{I}_{rest})^+$$

In a nutshell:

$$\begin{aligned}\text{Project}(|\psi_1\rangle, \text{list}, |\psi_2\rangle) &\rightarrow (|\psi_1\rangle\langle\psi_1|_{list} \otimes \mathbb{I}_{rest}) |\psi_2\rangle \\ \text{Project}(M_n, \text{list}, |\psi_2\rangle) &\rightarrow (M_{n_{list}} \otimes \mathbb{I}_{rest}) |\psi_2\rangle\langle\psi_2| (M_{n_{list}} \otimes \mathbb{I}_{rest})^\dagger \\ \text{Project}(|\psi_1\rangle, \text{list}, \rho) &\rightarrow (|\psi_1\rangle\langle\psi_1|_{list} \otimes \mathbb{I}_{rest}) \rho (|\psi_1\rangle\langle\psi_1|_{list} \otimes \mathbb{I}_{rest}) \\ \text{Project}(M_n, \text{list}, \rho) &\rightarrow (M_{n_{list}} \otimes \mathbb{I}_{rest}) \rho (M_{n_{list}} \otimes \mathbb{I}_{rest})^+\end{aligned}$$

[Input: (Matrix, list Matrix), Output: Matrix, Calls: Quantavo, StateComplexConjugate, LinearAlgebra, vec2matcol]

PS(vec/matcol, j, ϕ): PHASE SHIFTER: This procedure makes a phase shifter transformation to our state. It is implemented making the following mode transformation to the specified mode j :

$$a'_j = e^{i\phi} a_j \quad (10)$$

$$b'_j = e^{-i\phi} b_j \quad (11)$$

(that is transformation (10) for a **vec** and transformation (10) and (11) for **matcol** and **mat**.)

[Input: (Matrix, whole number, symbol or number), Output: Matrix, Calls: Quantavo, LinearAlgebra]

S

SqueezedVac(m,d,λ**)**: This procedure builds a **vec** describing the state $|\phi\rangle \sim \sum_{n=0}^{d-1} \lambda^n |n\rangle^{\otimes m}$ where m can be $m = 1, 2$. The state is not normalized.

[Input: (whole number, whole number, string or number), Output: 2 column Matrix, Calls: Quantavo, LinearAlgebra]

StateApprox(vec/matcol, list, n**)**: This procedure is used to reduce the size and complexity of the **vec** or **matcol** objects with an approximation.

Numerical Approximation: If the coefficients of our state are numbers then use as follows:
StateApprox(vec/matcol, [], n). All Rows for which the coefficient $< 10^{-n}$ will be deleted.

Symbolic Approximation: If the coefficients are symbolic polynomials (or monomials) and some variables are small, we may choose to delete all terms with a certain power in those variables. If we execute **S := StateApprox(M,[lambda,r],10)**, then all terms containing $\lambda^n r^m$ such that $n + m > 10$ will be deleted. For example if we have a state of the form:

$$M := \begin{bmatrix} 1 + y^4x^2 + yx^4 & [0, 0, 0, 0] \\ y^7x + x^3 + x^5 & [1, 1, 0, 0] \\ x^2 + x^4 + x^6 + y^5 & [2, 2, 0, 0] \end{bmatrix}$$

then,

```
S := StateApprox(M,[y,x],5);
```

will return the state:

$$\begin{bmatrix} 1 + yx^4 & [0, 0, 0, 0] \\ x^3 + x^5 & [1, 1, 0, 0] \\ x^2 + x^4 + y^5 & [2, 2, 0, 0] \end{bmatrix}$$

and

```
S := StateApprox(M,[y,x],1);
```

will return the state:

$$\begin{bmatrix} 1 & [0, 0, 0, 0] \end{bmatrix}$$

[Input: (Matrix, list, integer), Output: Matrix, Calls: Quantavo, LinearAlgebra]

StateComplexConjugate(matcol/mat): It will complex conjugate all the coefficients of an object of type **mat** or **matcol**.

[Input: Matrix, Output: Matrix, Calls: Quantavo, LinearAlgebra]

StateMultiply(number/matcol, matcol/vec): This procedure multiplies following the matrix multiplication rules the following objects:

k × matcol,

k × vec,

matcol × matcol

matcol × vec

where k is a number, a variable, a function or a string

[Input: (string/Matrix, Matrix), Output: Matrix, Calls: Quantavo, LinearAlgebra]

StateSort(vec/matcol): This procedure sorts by rows the objects **vec** or **matcol**. The order is determined in the following way. The modes are converted from the numeral base *d* to the numeral base 10 and are then sorted by increasing order. For **matcol** they are sorted

first according to the order of the “kets” and inside the same “ket” number, by the order of the “bras”. For example, with $d = 2$ and $K = 2$, $[\lambda, [0, 0]] < [\lambda^2, [1, 0]] < [\lambda, [0, 1]] < [\lambda^3, [1, 1]]$. The output is the ordered state.

[Input: Matrix, Output: Matrix, Calls: Quantavo, LinearAlgebra, Tribullesmatcol, Tribullesvec]

StateTrace(mat/matcol): This procedure performs the trace on **mat** or **matcol** states. It simply adds all diagonal elements.

[Input: Matrix, Output: Real Number or symbolic expression, Calls: Quantavo, LinearAlgebra]

StateNorm(vec): It evaluates the norm $\langle \phi | \phi \rangle$ of the state associated with the object **vec**

[Input: Matrix, Output: number or expression, Calls: Quantavo, LinearAlgebra]

StateNormalize(vec/matcol/mat): This procedure returns a normalized state such that $\langle \psi | \psi \rangle = 1$ for **vec** and $\text{Tr}(\rho)=1$ for **matcol** or **mat**. If the state was not normalized to begin with it will print: “the state is not normalized” and then return the normalized state.

[Input: Matrix, Output: Matrix and printed answer, Calls: Quantavo, LinearAlgebra, IsNormalized, LittleTrace]

StatePartialTranspose(vec/matcol/mat,s): This procedure works only for bipartite states (two modes only). If your state is not bipartite use **Traceout** to trace out the other modes. The parameter ‘**s**’ specifies if the partial transposition must be made with respect to the first or to the second mode. The output is the partially transposed **matcol** or **mat**.

[Input: (Matrix, 1 or 2), Output: Matrix, Calls: Quantavo, LinearAlgebra]

T

TensorProduct(vec/matcol, List, vec/matcol, List): This will return the tensor product between modes $[List_A]$ of state A and modes $[List_B]$ of state B. It therefore implements the operation, $(\rho, \text{listA}, \sigma, \text{listB}) \longrightarrow \rho_{listA} \otimes \sigma_{listB}$. Note that the dimension of each

state and its corresponding list must be equal. So for example we can declare two states,

```
Sq:=SqueezedVac(2, 3, ξ);
Fock2:=Matrix([[γ, [1,1]], [gamma, [2,2]]]);
```

$$\begin{bmatrix} 1 & [0, 0] \\ \xi & [1, 1] \\ \xi^2 & [2, 2] \end{bmatrix} \quad \begin{bmatrix} \gamma & [1, 1] \\ \gamma & [2, 2] \end{bmatrix}$$

and after using

```
TensorProduct(Sq,[1,4],Fock2,[2,3]);
```

we obtained the tensored **vec** object,

$$\begin{bmatrix} \gamma & [0, 1, 1, 0] \\ \gamma & [0, 2, 2, 0] \\ \xi\gamma & [1, 1, 1, 1] \\ \xi\gamma & [1, 2, 2, 1] \\ \xi^2\gamma & [2, 1, 1, 2] \\ \xi^2\gamma & [2, 2, 2, 2] \end{bmatrix}$$

[Input: Matrix,List,Matrix,List, Output: Matrix, Calls: Quantavo, LinearAlgebra]

TensorVac(vec/matcol/mat, m): This procedure tensors ‘m’ vacuum modes with an existing state. The new modes are in a product state with the original state. If it is a **vec** object it will therefore do the transformation $|\psi\rangle \longrightarrow |\psi\rangle \otimes |0\rangle^{\otimes m}$ and $\rho \longrightarrow \rho \otimes (|0\rangle\langle 0|)^{\otimes m}$ for **mat** and **matcol**. The procedure will also transform the global variable **K** \rightarrow **K** + **m**.

[Input: (Matrix, whole number), Output: Matrix, Calls: Quantavo, LinearAlgebra]

Traceout(matcol,i): This procedure takes the partial trace of a sparse density matrix with respect to mode i . Its input is an object of type **matcol**

[Input: (Matrix, whole number), Output: Matrix, Calls: Quantavo, LinearAlgebra]

Trim(Vector/vec/matcol): This procedure eliminates all the non zero entries of a **vector** of dimension d^K and converts it into an object of type **vec**. It also deletes all non-zero entries in objects of type **vec** and **matcol**

[Input: Vector/Matrix, Output: Matrix, Calls: Quantavo, LinearAlgebra]

U

UnitaryEvolution(Unitary Matrix, vec/matcol): Whether we have just built a unitary matrix with `BuildUnitary` or we have an arbitrary unitary matrix, we can use this procedure as follows:

```
V:= UnitaryEvolution(U, vec/matcol);
```

where U is the unitary matrix of dimension $K \times K$ and our state is described by a **vec** or a **matcol** object. This will effectively implement the mode transformation:

$$\bar{a}' = U\bar{a}$$

and return the **vec** or **matcol** after the transformation. (Note that for **matcol** it also does $\bar{b}' = U^\dagger \bar{b}$)

[Input: (Matrix, Matrix), Output: Matrix, Calls: Quantavo, LinearAlgebra]

V

Vac(Nr. of modes): This procedure returns a vacuum **vec** with the number of specified modes. For example `Vac(3)` will return $\begin{bmatrix} 1 & [0, 0, 0] \end{bmatrix}$

[Input: positive natural number, Output: Matrix, Calls: Quantavo, LinearAlgebra]

vec2mat(vec): It converts a **vec** object into a **mat** object. The procedures is independent of d and K .

[Input: 2 column Matrix, Output: Matrix, Calls: Quantavo, LinearAlgebra]

vec2matcol(vec): It converts a **vec** object into a **matcol** object effectively doing

$$\sum_{\bar{n}} \alpha_{\bar{n}} |\bar{n}\rangle \longrightarrow \sum_{\bar{n}\bar{m}} \alpha_{\bar{n}} \bar{\alpha}_{\bar{m}} |\bar{n}\rangle \langle \bar{m}|$$

Note that no normalization is implemented. The procedure is independent of d and K .

[Input: 2 column Matrix, Output: 3 Column Matrix, Calls: Quantavo, LinearAlgebra]

vec2poly(vec): It converts a **vec** object into a **poly** object

[Input: 2 column Matrix, Output: Polynomial, Calls: LinearAlgebra, Quantavo]

VectorModes(i): this procedure takes as input an integer between 1 and d^K and outputs a list with the equivalent number of photons in each mode. This way, **VectorModes(1) = [0,0,0]** if it is a 3 mode state and **VectorModes(2)=[1,0,0]** if $K = 3$ and $d = 2$ for example. (Note that d and K need to be defined)

[Input: integer, Output: List, Calls: Quantavo]

VectorRow(Indi, d): d (maximum number of photons) and K (number of modes) need to be specified. This is the inverse procedure of **VectorModes**. When a list with the number of photons in each mode (for instance **Indi = [0,1,0]**) and the value of d are given, it outputs the row number that corresponds to it in a **vector** type object). This way, for $d = 2$, $K = 3$ **VectorRow([0,1,0],2)=3**.

[Input: (List,d), Output: whole number, Calls: Quantavo]

Quantavo also uses the local procedures:

multiplymatcol, **multiplymatcolvec**, **Tribullesmatcol**, **Tribullesvec**, **VectorModes**, **VectorRow**,

indexvec, modesvec, vecBS, myvecBS, matcolBS, mymatcolBS, Projectvecvec, Projectmatcol,
Dvec, Dmat, Dmatcol, barra,histo;

.2 Copyright and Disclaimer

Copyright (c) 2008 Alvaro Feito Boirac.

This is the Module QUANTAVO, a toolbox for Quantum Optics calculations that can be used in
MapleTM (Waterloo Maple Inc.)

It is released under the GNU General Public License v3 which can be obtained at <http://www.gnu.org/licenses/gpl.html> . Please acknowledge its use if used to establish results for a published work. If you make any improvements or find any bugs the author will be thankful if you can let him know.

DISCLAIMER: This Module is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY, without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

.3 Maple Code

```

1 ##########
2 #####
3
4 #####          Algebra Operations      #####
5
6 #####
7 #####
8
9 ## tensor product between two Maple matrices ##
10 ### (not specifically used by Quantavo objects) ###
11 DP:=proc(A::Matrix, B::Matrix)
12   local M, P, i, j;
13
14   M := Matrix(LinearAlgebra:-RowDimension(A)*LinearAlgebra:-RowDimension(B),
15             LinearAlgebra:-ColumnDimension(A)*LinearAlgebra:-ColumnDimension(B));
16   P := Matrix(LinearAlgebra:-RowDimension(B), LinearAlgebra:-ColumnDimension(B));
17
18   for i to LinearAlgebra:-RowDimension(A) do
19     for j to LinearAlgebra:-ColumnDimension(A) do
20
21       P := LinearAlgebra:-ScalarMultiply(
22         B, A[i, j]);
23       M[1 + (i - 1)*LinearAlgebra:-RowDimension(B) .. .
24 (i - 1)*LinearAlgebra:-RowDimension(B) + LinearAlgebra:-RowDimension(B) ,
25           1 + (j - 1)*LinearAlgebra:-ColumnDimension(B) .. .
26 (j - 1)*LinearAlgebra:-ColumnDimension(B) + LinearAlgebra:-ColumnDimension(B)] := P;
27     end do;
28   end do;
29   M;
30 end proc;
31
32 #####      #####
33 ### tensor product between modes [listA] of state A
34 ### and modes [listB] of state B
35
36 TensorProduct:=proc(A, listA ,B, listB )
37   local i,j ,t ,dA,dB, Ket, Bra , AA,BB,C;
38 ##
39 ##### check if dimensions agree #####
40   if nops(listA)= nops(A[1,2]) and
41     nops(listB)=nops(B[1,2]) then

```

```

42
43         dA:=LinearAlgebra:-RowDimension(A):
44         dB:=LinearAlgebra:-RowDimension(B):
45
46 ##### VEC-VEC case #####
47         if LinearAlgebra:-ColumnDimension(A)=2 and
48             LinearAlgebra:-ColumnDimension(B)=2 then
49     ### declare the new matrix with the appropriate dimensions
50     C:=Matrix(dA*dB,2):
51
52         for i from 1 to dA do
53             for j from 1 to dB do
54
55     ##### build new ket #####
56             Ket:=[seq(0,s=1..nops(listA)+nops(listB))]:
57
58             for t from 1 to nops(listA) do
59                 Ket:=subsop(listA[t]=A[i,2][t],Ket);
60             od:
61
62             for t from 1 to nops(listB) do
63                 Ket:=subsop(listB[t]=B[j,2][t],Ket);
64             od:
65     ##### build new ket (end) #####
66     ## product of matrix elements ##
67             C[dB*(i-1)+j,1]:=A[i,1]*B[j,1]:
68             C[dB*(i-1)+j,2]:=Ket:
69
70             od:
71         od:
72         print("k and d are now", findKnd(C));
73         return C;
74
75 ##### matcol-matcol case #####
76         elif LinearAlgebra:-ColumnDimension(A)=3 and
77             LinearAlgebra:-ColumnDimension(B) = 3 then
78
79         C:=Matrix(dA*dB,3):
80
81         for i from 1 to dA do
82             for j from 1 to dB do
83
84     ##### build new ket and bra#####
85             Ket:=[seq(0,s=1..nops(listA)+nops(listB))];
86             Bra:=[seq(0,s=1..nops(listA)+nops(listB))];

```

```

87
88         for t from 1 to nops(listA) do
89             Ket:=subsop(listA[t]=A[i,2][t],Ket);
90             Bra:=subsop(listA[t]=A[i,3][t],Bra);
91             od;
92
93         for t from 1 to nops(listB) do
94             Ket:=subsop(listB[t]=B[j,2][t],Ket);
95             Bra:=subsop(listB[t]=B[j,3][t],Bra);
96             od;
97 ##### build new ket and bra (end) #####
98
99             C[dB*(i-1)+j,1]:=A[i,1]*B[j,1];
100            C[dB*(i-1)+j,2]:=Ket;
101            C[dB*(i-1)+j,3]:=Bra;
102            od;
103        od;
104        print("K and d are now", findKnd(C));
105        return C;
106
107 ##### MATCOL-VEC case #####
108     elif LinearAlgebra:-ColumnDimension(A)=3 and
109         LinearAlgebra:-ColumnDimension(B)=2 then
110         BB:=vec2matcol(B);
111
112         dB:=LinearAlgebra:-RowDimension(BB);
113         C:=Matrix(dA*dB,3);
114
115         for i from 1 to dA do
116             for j from 1 to dB do
117
118 ##### build new ket and bra#####
119             Ket:=[seq(0,s=1..nops(listA)+nops(listB))];
120             Bra:=[seq(0,s=1..nops(listA)+nops(listB))];
121
122             for t from 1 to nops(listA) do
123                 Ket:=subsop(listA[t]=A[i,2][t],Ket);
124                 Bra:=subsop(listA[t]=A[i,3][t],Bra);
125                 od;
126
127             for t from 1 to nops(listB) do
128                 Ket:=subsop(listB[t]=BB[j,2][t],Ket);
129                 Bra:=subsop(listB[t]=BB[j,3][t],Bra);
130                 od;
131 ##### build new ket and bra (end) #####

```

```

132          C[dB*(i-1)+j,1]:=A[i,1]*BB[j,1]:
133          C[dB*(i-1)+j,2]:=Ket:
134          C[dB*(i-1)+j,3]:=Bra:
135      od:
136  od:
137  print("K and d are now", findKnd(C));
138  return C;
139 ##### VEC-MATCOL case #####
140  elif LinearAlgebra:-ColumnDimension(A)=2 and
141  LinearAlgebra:-ColumnDimension(B)=3 then
142  AA:=vec2matcol(A):
143
144  C:=Matrix(dA*dB,3):
145
146  for i from 1 to dA do
147    for j from 1 to dB do
148
149 ##### build new ket and bra#####
150    Ket:=[seq(0,s=1..nops(listA)+nops(listB))];
151    Bra:=[seq(0,s=1..nops(listA)+nops(listB))];
152
153    for t from 1 to nops(listA) do
154      Ket:=subsop(listA[t]=AA[i,2][t],Ket);
155      Bra:=subsop(listA[t]=AA[i,3][t],Bra);
156    od:
157
158    for t from 1 to nops(listB) do
159      Ket:=subsop(listB[t]=B[j,2][t],Ket);
160      Bra:=subsop(listB[t]=B[j,3][t],Bra);
161    od:
162 ##### build new ket and bra (end) #####
163    C[dB*(i-1)+j,1]:=AA[i,1]*B[j,1]:
164    C[dB*(i-1)+j,2]:=Ket:
165    C[dB*(i-1)+j,3]:=Bra:
166  od:
167  od:
168  print("K and d are now", findKnd(C));
169  return C;
170
171 ##### otherwise #####
172  else
173  print("are your states VEC or MATCOL objects? are they well defined?"):
174  fi:
175
176  else

```

```

177 print("the number of modes in the state and the list don't coincide");
178 fi:
179
180 end proc:
181
182 #####      #####      #####      #####
183 ## kronecker delta #
184 DeltaK:=proc(j, k)
185     if not type(j - k, numeric)
186         then RETURN('procname(args)')
187     end if;
188     if j = k then 1 else 0 end if;
189 end proc;
190
191
192 #####      #####      #####      #####
193 StateMultiply:=proc(A, B)
194 local i, C;
195     if whattype(A) <> Matrix and B[1, 1] <> 0 and
196         LinearAlgebra:-ColumnDimension(B
197 ) = 3 then
198
199     C := Matrix(LinearAlgebra:-RowDimension(B), 3);
200     for i to LinearAlgebra:-RowDimension(B) do C[i, 1] := A*B[i, 1];
201     C[i, 2] := B[i, 2];
202     C[i, 3] := B[i, 3];
203     end do;
204     return C;
205     elif whattype(A) <> Matrix and B[1, 1] <> 0 and
206         LinearAlgebra:-ColumnDimension(B) = 2 then
207     C := Matrix(LinearAlgebra:-RowDimension(B), 2);
208     for i to LinearAlgebra:-RowDimension(B) do C[i, 1] := A*B[i, 1];
209     C[i, 2] := B[i, 2];
210     end do;
211     return C;
212     elif whattype(A) = Matrix and A[1, 1] <> 0 and B[1, 1] <> 0 and
213         LinearAlgebra:-ColumnDimension(A) = 3 and
214         LinearAlgebra:-ColumnDimension(B) =
215         3 then
216     C := multiplymatcol(A, B);
217     return C;
218     elif whattype(A) = Matrix and A[1, 1] <> 0 and B[1, 1] <> 0 and
219         LinearAlgebra:-ColumnDimension(A) = 3 and
220         LinearAlgebra:-ColumnDimension(B) =
221         2 then

```

```

222      C := multiplymatcolvec(A, B);
223      return C;
224  else
225      print("this procedure multiplies
226 MATCOL x VEC or MATCOL x MATCOL Number x VEC or Number x MATCOL");
227      print("are your states well defined?");
228  end if;
229 end proc;
230
231
232
233 #####      #####      #####      #####
234
235 multiplymatcolvec:=proc(M, V)
236 local i, j, s, halt, TempoV, W;
237 W := Matrix([0, 0]);
238 for i to LinearAlgebra:-RowDimension(M) do for j to
239     LinearAlgebra:-RowDimension(V) do if M[i, 3] = V[j, 2] then
240         s := 1;
241         halt := 0;
242         while s <= LinearAlgebra:-RowDimension(W)
243 and halt = 0 do
244             if M[i, 2] = W[s, 2] then
245                 W[s, 1] := W[s, 1] + M[i, 1]*V[j, 1];
246                 halt := 1;
247             else
248                 s := s + 1
249             end if;
250         end do;
251         if halt = 0 then
252             TempoV := LinearAlgebra:-Transpose(Vector([M[i, 1]*V[j, 1], [M[i, 2]]]
253                                         ]));
254             W := Matrix([[W], [TempoV]]);
255         else
256             end if;
257         end if;
258     end do;
259 end do;
260
261 W := simplify(expand(LinearAlgebra:-DeleteRow(W, 1)));
262 return W;
263 end proc;
264
265
266 #####      #####      #####      #####

```

```

267 multiplymatcol:=proc(AA, BB)
268   local i, j, dimiA, dimiB, A, B, V, M, s, halt;
269   dimiA := LinearAlgebra:-RowDimension(AA);
270   dimiB := LinearAlgebra:-RowDimension(BB);
271   M := Matrix([0, 0, 0]);
272   A := indexstate(AA);
273   B := indexstate(BB);
274   for i to dimiA do for j to dimiB do if A[i, 3] = B[j, 2] then
275     s := 1;
276     halt := 0;
277     while s <= LinearAlgebra:-RowDimension(M) and halt = 0 do
278
279     if M[s, 2] = A[i, 2] and M[s, 3] = B[j, 3] then
280       M[s, 1] := M[s, 1] + A[i, 1]*B[j, 1];
281       halt := 1;
282     else
283       s := s + 1
284     end if;
285   end do;
286   if halt = 0 then
287     V := LinearAlgebra:-Transpose(Vector([A[i, 1]*B[j, 1], A[i, 2],
288                                         B[j, 3]]));
289     M := Matrix([[M], [V]]);
290   else
291     end if;
292   else
293     end if;
294   end do;
295   end do;
296   M := LinearAlgebra:-DeleteRow(M, 1);
297   return modesmatcol(M);
298 end proc;
299
300
301 #####      #####
302 StateNorm:=proc(V)
303   local i, Norma;
304   if V[1, 1] <> 0 and LinearAlgebra:-ColumnDimension(V) = 2 then
305     Norma := simplify(sqrt(add(abs(V[i, 1])^2,
306                               i = 1 .. LinearAlgebra:-RowDimension(V))));
307     return Norma;
308   else
309     print("This procedure evaluates the norm of a VEC object");
310
311     print("is your object well defined?");

```

```

312         end if;
313     end proc;
314
315
316 ##### ##### ##### #####
317 ### take the trace of a state ###
318 StateTrace:=proc(M::Matrix)
319   local i, Tr;
320   if M[1, 1] = 0 then
321     Tr := simplify(add(M[i, i], i = 2 .. LinearAlgebra:-RowDimension(M)));
322     return Tr;
323   else
324     Tr := 0;
325     for i to LinearAlgebra:-RowDimension(M) do if M[i, 2] = M[i, 3] then
326       Tr := Tr + M[i, 1]
327     else
328       end if;
329     end do;
330     Tr := simplify(Tr);
331     return Tr;
332   end if;
333 end proc;
334
335
336 ##### ##### ##### #####
337 ### traceout a mode ##
338 Traceout:=proc(Ma::Matrix, r)
339   local M,j, Out, i,L;
340   global K, d;
341
342   if LinearAlgebra:-ColumnDimension(Ma)=2 then
343     M:=vec2matcol(Ma):
344   elif LinearAlgebra:-ColumnDimension(Ma)=3 then
345     M := Matrix(Ma);
346   else
347     print("is your state a well defined VEC or MATCOL object?")
348   fi:
349
350   ### keep the elements of the density matrix that have
351   ### |012x23><329x32| same element x at the position "r". ##
352
353 M:=M[[ seq(`if `((M[i,2][r]<>M[i,3][r]),NULL,i),
354 i=1..LinearAlgebra:-RowDimension(M))],[1..3]]];
355
356 print("K and d were", findKnd(M));

```

```

357 K:=K-1:
358 ##### delete the extra terms in the modes #####
359 M:=LinearAlgebra:-Map[(i,j)->evalb(j=2)](x->VectorRow(subsop(r=NULL,x),d),M);
360 M:=LinearAlgebra:-Map[(i,j)->evalb(j=3)](x->VectorRow(subsop(r=NULL,x),d),M);
361
362
363 ##### sort #####
364 M:=TribulleMatcol(M):
365
366
367 ##### Add the repeated elements #####
368
369 L:=[1,seq(`if `((M[i-1,2]<>M[i,2]) or M[i-1,3]<>M[i,3]),i,NULL),
370 i=2..LinearAlgebra:-RowDimension(M));
371 Out:=M[L,[1..3]]:
372
373
374 if LinearAlgebra:-RowDimension(Out)<>1 then
375 for j from 1 to LinearAlgebra:-RowDimension(Out)-1 do
376   for i from L[j]+1 to L[j+1]-1 do
377     Out[j,1]:=Out[j,1]+M[i,1];
378 od:
379 od:
380 ##### not to forget the last elements #####
381 if i< LinearAlgebra:-RowDimension(M) then
382   while i<LinearAlgebra:-RowDimension(M) do
383     if M[i,2]=M[i+1,2] and M[i,3]=M[i+1,3] then
384       Out[LinearAlgebra:-RowDimension(Out),1]:=-
385     Out[LinearAlgebra:-RowDimension(Out),1]+M[i+1,1];
386     i:=i+1;
387   else
388     fi:
389   od;
390 else
391   fi:
392 ##### in case it's only one element#####
393 elif LinearAlgebra:-RowDimension(Out)=1 then
394 for i from 2 to LinearAlgebra:-RowDimension(M) do
395   Out[1,1]:=Out[1,1]+M[i,1];
396 od:
397 else fi;
398
399
400 ### back to modes #####
401 M:=LinearAlgebra:-Map[(i,j)->evalb(j=2)](x->VectorModes(x),Out);

```

```

402 M:=LinearAlgebra:-Map[(i,j)->evalb(j=3)](x->VectorModes(x),M);
403
404 print("K and d are now", findKnd(M));
405 return M;
406
407
408 end proc;
409
410
411 #####      #####
412 StateNormalize:=proc(M)
413 local i, j, Tra, Nor, Kout;
414 if IsNormalized(M) = 1 then
415   return M
416 else
417   if M[1, 1] = 0 then
418     Tra := StateTrace(M);
419     for i from 2 to LinearAlgebra:-RowDimension(M) do for j from 2 to
420       LinearAlgebra:-RowDimension(M) do M[i, j] := (M[i, j])/(Tra)
421     end do;
422   end do;
423   return M;
424 elif M[1, 1] <> 0 and LinearAlgebra:-ColumnDimension(M) <= 2 then
425
426   Nor := sqrt(add(M[i, 1]^2, i = 1 .. LinearAlgebra:-RowDimension(M)));
427
428   Kout:=Matrix(M):
429   Kout:=LinearAlgebra:-Map[(i,j)->evalb(j=1)](x->x/Nor,Kout):
430   return Kout;
431
432 elif M[1, 1] <> 0 and LinearAlgebra:-ColumnDimension(M) = 3 then
433   Tra := StateTrace(M);
434   Kout:=Matrix(M):
435   Kout:=LinearAlgebra:-Map[(i,j)->evalb(j=1)](x->x/Tra,Kout):
436   return Kout;
437 else
438   print("is your state well defined?")
439   end if;
440   end if;
441 end proc;
442
443
444
445 #####      #####
446 StateComplexConjugate:=proc(M)

```

```

447 local i, j, dimi, dimj, V, K;
448   if M[1, 1]  $\diamond$  0 and LinearAlgebra:-ColumnDimension(M) = 3 then
449     dimi := LinearAlgebra:-RowDimension(M);
450     K := Matrix([0, 0, 0]);
451     for i to dimi do
452       if M[i, 2] = M[i, 3] then
453         V := LinearAlgebra:-Transpose(Vector([conjugate(M[i, 1]), [M[i, 2]],
454                                         [M[i, 3]]]));
455         K := Matrix([[K], [V]]);
456       else
457         V := LinearAlgebra:-Transpose(Vector([conjugate(M[i, 1]), [M[i, 3]],
458                                         [M[i, 2]]]));
459         K := Matrix([[K], [V]]);
460       end if;
461     end do;
462     K := LinearAlgebra:-DeleteRow(K, 1);
463     return K;
464   elif M[1, 1] = 0 then
465     dimi := LinearAlgebra:-RowDimension(M);
466     dimj := LinearAlgebra:-ColumnDimension(M);
467     K := Matrix(dimi, dimj);
468     for i to dimi do
469       for j to dimj do
470         if i = j then
471           K[i, j] := conjugate(M[i, j])
472         elif i  $\diamond$  j and (i = 1 or j = 1) then
473           K[i, j] := M[i, j]
474         elif i  $\diamond$  j and 1 < i and 1 < j then
475           K[i, j] := conjugate(M[j, i])
476         else
477           end if;
478         end do;
479       end do;
480       return K;
481     else
482       print("is your object MAT or MATCOL well defined?");
483     end if;
484   end proc;
485
486 ##### ##### ##### ##### #####
487 StatePartialTranspose:=proc(M, s)
488   local i, Matc, Kout;
489
490   ##### choose vec/mat/matcol convert 2 matcol #####
491   if LinearAlgebra:-ColumnDimension(M)=2 and M[1,1]<>0 then

```

```

492 Matc:=vec2matcol(M):
493 elif M[1,1]=0 then
494 Matc:=mat2matcol(M):
495 else
496 Matc:=Matrix(M):
497 fi:
498
499 if nops(Matc[1, 2]) = 2 and s = 1 then
500 Kout := Matrix(LinearAlgebra:-RowDimension(Matc), 3);
501 for i to LinearAlgebra:-RowDimension(Matc) do
502 Kout[i, 1] := Matc[i, 1];
503 Kout[i, 2] := [Matc[i, 3][1], Matc[i, 2][2]];
504 Kout[i, 3] := [Matc[i, 2][1], Matc[i, 3][2]];
505 end do;
506 if M[1,1]=0 then
507 return matcol2mat(Kout);
508 else
509 return Kout;
510 fi:
511 elif nops(Matc[1, 2]) = 2 and s = 2 then
512 Kout := Matrix(LinearAlgebra:-RowDimension(Matc), 3);
513 for i to LinearAlgebra:-RowDimension(Matc) do
514 Kout[i, 1] := Matc[i, 1];
515 Kout[i, 2] := [Matc[i, 2][1], Matc[i, 3][2]];
516 Kout[i, 3] := [Matc[i, 3][1], Matc[i, 2][2]];
517 end do;
518
519 if M[1,1]=0 then
520 return matcol2mat(Kout);
521 else
522 return Kout;
523 fi:
524
525 else
526 print("Partial Transpose only works for two mode states.");
527 print("Trace Out the other states or make sure it is a VEC, MAT or MATCOL");
528 end if;
529 end proc:
530
531 #####
532 #####
533 #####
534 #####
535 ##### State Properties #####
536

```

```

537 #####
538 #####
539
540
541
542 LogNegativity:=proc(M)
543   local LogNeg;
544   LogNeg := log[2](2*Negativity(M) + 1);
545   return LogNeg;
546 end proc;
547
548
549
550 #####
551 Negativity:=proc(M)
552   local i, j, M1, M0, Kout, Eig, Neg;
553
554   if M[1, 1] <> 0 and LinearAlgebra:-ColumnDimension(M) = 3 then
555     M0 := StatePartialTranspose(M, 1);
556
557   elif M[1, 1] <> 0 and LinearAlgebra:-ColumnDimension(M) = 2 then
558     M0:=StatePartialTranspose(vec2matcol(M),1)
559
560   elif M[1, 1] = 0 then
561
562     M0 := mat2matcol(M);
563     M0 := StatePartialTranspose(M0, 1);
564
565   else
566
567     print("is your state a well defined MAT or MATCOL object ?");
568   end if;
569
570 findKnd(M0):
571   M1 := matcol2mat(M0);
572   Kout := Matrix(LinearAlgebra:-RowDimension(M1) - 1,
573                 LinearAlgebra:-ColumnDimension(M1) - 1);
574
575   for i from 2 to LinearAlgebra:-ColumnDimension(M1) do for j from 2 to
576     LinearAlgebra:-RowDimension(M1) do Kout[i - 1, j - 1] := M1[i, j]
577
578   end do;
579 end do;
580
581 Eig := simplify(LinearAlgebra:-Eigenvalues(Kout));

```

```

582
583
584     if M[1, 1] <> 0 and LinearAlgebra:-ColumnDimension(M) = 2 then
585         Neg := add(1/2*abs(Eig[i]) - 1/2*Eig[i],
586 i = 1 .. LinearAlgebra:-Dimension(Eig))/StateTrace(vec2matcol(M));
587     else
588         Neg := add(1/2*abs(Eig[i]) - 1/2*Eig[i],
589 i = 1 .. LinearAlgebra:-Dimension(Eig))/StateTrace(M);
590     fi;
591
592     return Neg;
593
594
595 end proc;
596
597
598
599 #####      #####
600 Entropy:=proc(State) local Entro, s, Matt, Mattl, Eign;
601
602
603 if State[1,1]<>0 and
604 LinearAlgebra:-ColumnDimension(State)=2 and
605 nops(State[1,2])=1 then
606 Matt:=vec2mat(State):
607 elif State[1,1]<>0 and
608 LinearAlgebra:-ColumnDimension(State)=3 and
609 nops(State[1,2])=1 then
610 Matt:=matcol2mat(State);
611 elif State[1,1]=0 and nops(State[1,2])=1 then
612 Matt:=Matrix(State);
613 else
614 print("is your state (vec, mat or matcol) well defined?");
615 print("this procedure only works after tracing out.
616 Your state must have one mode only");
617 fi;
618
619 Mattl:=LinearAlgebra:-DeleteColumn(LinearAlgebra:-DeleteRow(Matt,1),1):
620 Eign:=LinearAlgebra:-Eigenvalues(Mattl)/StateTrace(Matt):
621 Entro:=simplify(-add(Eign[s]*log[2](Eign[s]),
622 s=1..LinearAlgebra:-Dimension(Eign)));
623 return Entro;
624 end proc;
625
626

```

```

627
628 ##### ##### ##### ##### #####
629 Energy:=proc(M)
630 local Nor,Ene,i,s;
631 global hbar,nu;
632
633 Nor:=IsNormalized(M):
634 if Nor=1 then
635 else
636 print("if your state is not normalized, so is your Energy");
637 fi:
638 hbar := `&hbar`;
639
640
641 if M[1,1]<>0 and LinearAlgebra:-ColumnDimension(M)=2 then
642
643 Ene:=hbar*nu*simplify(
644 (add(M[i,1]*conjugate(M[i,1])*(add(M[i,2][s],s=1..nops(M[i,2]))+
645 nops(M[i,2])/2), i=1..LinearAlgebra:-RowDimension(M)));
646
647 elif M[1,1]<>0 and LinearAlgebra:-ColumnDimension(M)=3 then
648
649 Ene:=hbar*nu*simplify(add(
650 `if `(M[i,2]=M[i,3],M[i,1]*(add(M[i,2][s],s=1..nops(M[i,2]))+
651 nops(M[i,2])/2),0), i=1..LinearAlgebra:-RowDimension(M));
652 return Ene;
653 else
654 print("is your state a MAT or MATCOL object ?");
655 fi:
656
657 end proc:
658
659
660 ##### ##### ##### ##### #####
661 StateSort:=proc(Ma)
662 local i,j,Out,L,M;
663 global ind;
664 findKnd(Ma):
665
666 if Ma[1, 1] <> 0 and LinearAlgebra:-ColumnDimension(Ma) = 3 then
667
668 M := indexstate(Ma);
669 M := Tribullesmatcol(M);
670
671

```

```

672
673 ##### Add the repeated elements #####
674
675 L:=[1 ,seq(`if `^(M[ i -1,2]<>M[ i ,2] or M[ i -1,3]<>M[ i ,3] ,i ,NULL) ,
676 i =2.. LinearAlgebra:-RowDimension(M)) ];
677 Out:=M[L ,[ 1 .. 3 ]];
678
679
680 if LinearAlgebra:-RowDimension(Out)<>1 then
681 for j from 1 to LinearAlgebra:-RowDimension(Out)-1 do
682 for i from L[j]+1 to L[j+1]-1 do
683 Out[ j ,1]:=Out[ j ,1]+M[ i ,1];
684 od;
685 od;
686 ##### not to forget the last elements #####
687 if i< LinearAlgebra:-RowDimension(M) then
688 while i<LinearAlgebra:-RowDimension(M) do
689 if M[ i ,2]=M[ i +1 ,2] and M[ i ,3]=M[ i +1 ,3] then
690 Out[ LinearAlgebra:-RowDimension(Out) ,1]:=Out[ LinearAlgebra:-RowDimension(Out) ,1]+M[ i +1 ,1];
691 i := i +1;
692 else
693 fi;
694 od;
695 else
696 fi;
697 ##### in case it's only one element#####
698 elif LinearAlgebra:-RowDimension(Out)=1 then
699 for i from 2 to LinearAlgebra:-RowDimension(M) do
700 Out[ 1 ,1]:=Out[ 1 ,1]+M[ i ,1];
701 od;
702 else fi;
703 #####back from index to modes#####
704 Out:=modesmatcol(Out);
705 return Out;
706
707
708
709 ##########
710 elif Ma[1 , 1] <> 0 and LinearAlgebra:-ColumnDimension(Ma) = 2 then
711
712 M := indexvec(Ma);
713 M := Tribullesvec(M);
714
715
716 ##### Add the repeated elements #####

```

```

717
718 L:=[1 ,seq( `if `((M[ ind -1,2]<>M[ ind ,2] ,ind ,NULL),
719 ind =2.. LinearAlgebra:-RowDimension(M))];
720 Out:=M[L,[ 1..2]]:
721
722
723 if LinearAlgebra:-RowDimension(Out)<>1 then
724 for j from 1 to LinearAlgebra:-RowDimension(Out)-1 do
725   for i from L[j]+1 to L[j+1]-1 do
726     Out[j,1]:=Out[j,1]+M[i,1];
727   od:
728 od:
729 ##### not to forget the last elements #####
730 if i< LinearAlgebra:-RowDimension(M) then
731   while i<LinearAlgebra:-RowDimension(M) do
732     if M[i,2]=M[i+1,2] then
733       Out[LinearAlgebra:-RowDimension(Out),1]:=-
734     Out[LinearAlgebra:-RowDimension(Out),1]+M[i+1,1];
735     i:=i+1;
736   else
737     fi:
738   od;
739 else
740   fi:
741 ##### in case it's only one element#####
742 elif LinearAlgebra:-RowDimension(Out)=1 then
743 for i from 2 to LinearAlgebra:-RowDimension(M) do
744   Out[1,1]:=Out[1,1]+M[i,1];
745 od:
746 else fi;
747 Out:=modesvec(Out):
748 return Out;
749
750 else
751   print("is your state a VEC or MATCOL? Is it well defined?")
752 end if;
753 end proc;
754
755 #####      #####
756
757 StateApprox:=proc(M::Matrix ,L,n)
758 local Ma,lili , todelete ,i;
759 Ma:=Matrix(M):
760 todelete :=[]:
761

```

```

762 if nops(L)=0 then
763
764   for i from 1 to LinearAlgebra:-RowDimension(Ma) do
765     if evalf(abs(Ma[i, 1])) < 10^(-n) then
766       todelete:=[op(todelete),i]:
767     else
768       fi:
769     od:
770
771 else
772
773   for i from 1 to LinearAlgebra:-RowDimension(Ma) do
774
775     if degree(Ma[i,1],{ op(L)})>n and whattype(Ma[i,1])='+' then
776       Ma[i,1]:=expand(Ma[i,1]):
777
778       lili:=convert(expand(Ma[i,1]),list):
779       lili:=map(x->if '(degree(x,{ op(L)})>n,NULL,x), lili):
780
781       if nops(lili)=0 then
782         todelete:=[op(todelete),i]:
783       else
784         Ma[i,1]:=add(lili[j],j=1..nops(lili)):
785       fi:
786
787     elif degree(Ma[i,1],{ op(L)})>n and whattype(Ma[i,1])='*' then
788
789       todelete:=[op(todelete),i]:
790
791     fi:
792
793   od:
794   fi:
795
796   Ma:=LinearAlgebra:-DeleteRow(Ma,todelete):
797   return Ma;
798
799 end proc:
800
801 #####      #####
802 ExpandState:=proc(A)
803 local i,j, Out;
804 Out:=A:
805 Out:=LinearAlgebra:-Map[(i,j)->evalb(j=1)](x->expand(x),Out):
806 return Out;

```

```

807 end proc:
808 ##### ##### ##### ##### #####
809
810 EvalState:=proc(M) local OutState;
811
812 OutState:=M;
813 OutState:=LinearAlgebra:-Map[(i,j)->evalb(j=1)](x->evalf(x),OutState);
814 return OutState;
815 end proc:
816
817 ##### ##### ##### ##### #####
818 IsHermitian:=proc(M)
819 local i, j, halt, M1, M2, M3, s;
820 halt := 0;
821 if M[1, 1] <> 0 and LinearAlgebra:-ColumnDimension(M) = 3 then
822 M1 := StateComplexConjugate(M);
823 M1 := StateSort(M1);
824 M2 := StateSort(M);
825 for i to LinearAlgebra:-RowDimension(M) while halt = 0 do
826 if M1[i, 1] = M2
827 [i, 1] and M1[i, 2] = M2[i, 2] and M1[i, 3] = M2[i, 3] then
828 else
829 halt := 1
830 end if;
831 end do;
832 if halt = 0 then
833 print("The density matrix is Hermitian");
834 return halt;
835 else
836 print("The density matrix is NOT Hermitian");
837 return halt;
838 end if;
839 elif M[1, 1] = 0 then
840 M1 := LinearAlgebra:-DeleteRow(M, 1);
841 M1 := LinearAlgebra:-DeleteColumn(M1, 1);
842 M2 := LinearAlgebra:-HermitianTranspose(M1);
843 M3 := M1 - M2;
844 s := 0;
845 for i to LinearAlgebra:-RowDimension(M3) do for j to
846 LinearAlgebra:-ColumnDimension(M3) do if M3[i, j] <> 0 then
847 s := s + 1
848 else
849 end if;
850 end do;
851 end do;

```

```

852     if s = 0 then
853         print("it is a Hermitian density matrix");
854         return halt;
855     else
856         print("it is NOT a Hermitian density matrix");
857         halt := 1;
858         return halt;
859     end if;
860   else
861     print("is your MATCOL or MAT well defined ?");
862   end if;
863 end proc;
864
865
866 #####      #####
867 IsNormalized:=proc(M)
868 local i, Tra;
869 if LinearAlgebra:-ColumnDimension(M) = 2 and M[1, 1] <> 0 then
870   Tra := simplify(sqrt(add(M[i, 1]^2, i = 1 .. LinearAlgebra:-RowDimension(M)
871   )));
872   if Tra = 1 then
873     print("the state vector is normalized");
874     return Tra;
875   else
876     print("the norm of the state vector is", Tra)
877   end if;
878 else
879   Tra := StateTrace(M);
880   if Tra = 1 then
881     print("the state is normalized");
882     return Tra;
883   else
884     print("the state is NOT normalized");
885     print("the trace of the state is", Tra);
886   end if;
887 end if;
888 end proc;
889
890
891
892 #####      #####
893 findKnd:=proc(M::Matrix)
894 local i, dtemp;
895 global K, d;
896 if M[1, 1] <> 0 then

```

```

897      K := nops(M[1, 2]);
898      dtemp := 0;
899      if LinearAlgebra:-ColumnDimension(M)=3 then
900          for i to LinearAlgebra:-RowDimension(M) do
901              if dtemp < max(op(M[i, 2]),op(M[i,3])) then
902                  dtemp := max(op(M[i, 2]),op(M[i,3]))
903              else
904                  end if;
905              end do;
906          d := dtemp + 1;
907          return K, d;
908      elif LinearAlgebra:-ColumnDimension(M)=2 then
909          for i to LinearAlgebra:-RowDimension(M) do
910              if dtemp < max(op(M[i, 2])) then
911                  dtemp := max(op(M[i, 2]))
912              else
913                  end if;
914              end do;
915          d := dtemp + 1;
916          return K, d;
917      fi;
918      elif M[1, 1] = 0 then
919          K := nops(M[1, 2]);
920          dtemp := 0;
921          for i to LinearAlgebra:-RowDimension(M) do
922              if dtemp < max(op(M[i, 1])) then
923                  dtemp := max(op(M[i, 1]))
924              else
925                  end if;
926              end do;
927          d := dtemp + 1;
928          return K, d;
929      else
930          print("is your state VEC, MAT or MATCOL well defined ?");
931      end if;
932  end proc;
933
934
935 #####      #####      #####      #####
936 Tribullesmatcol:=proc(L)
937     local i, j, tempa, tempi, tempo, T;
938     T := L;
939     for i to LinearAlgebra:-RowDimension(L) - 1 do for j to
940         LinearAlgebra:-RowDimension(L) - i do if T[j + 1, 2] < T[j, 2] then
941             tempa := T[j, 1];

```

```

942         tempi := T[j, 2];
943         tempo := T[j, 3];
944         T[j, 1] := T[j + 1, 1];
945         T[j, 2] := T[j + 1, 2];
946         T[j, 3] := T[j + 1, 3];
947         T[j + 1, 1] := tempa;
948         T[j + 1, 2] := tempi;
949         T[j + 1, 3] := tempo;
950     elif T[j, 2] = T[j + 1, 2] and T[j + 1, 3] < T[j, 3] then
951         tempa := T[j, 1];
952         tempi := T[j, 2];
953         tempo := T[j, 3];
954         T[j, 1] := T[j + 1, 1];
955         T[j, 2] := T[j + 1, 2];
956         T[j, 3] := T[j + 1, 3];
957         T[j + 1, 1] := tempa;
958         T[j + 1, 2] := tempi;
959         T[j + 1, 3] := tempo;
960     else
961     end if;
962   end do;
963 end do;
964 return T;
965 end proc;
966
967
968 #####      #####
969 Tribullesvec:=proc(L)
970   local i, j, tempa, tempi, T;
971   T := L;
972   for i to LinearAlgebra:-RowDimension(L) - 1 do for j to
973     LinearAlgebra:-RowDimension(L) - i do if T[j + 1, 2] < T[j, 2] then
974       tempa := T[j, 1];
975       tempi := T[j, 2];
976       T[j, 1] := T[j + 1, 1];
977       T[j, 2] := T[j + 1, 2];
978       T[j + 1, 1] := tempa;
979       T[j + 1, 2] := tempi;
980     else
981     end if;
982   end do;
983 end do;
984 return T;
985 end proc;
986

```

```

987 ##### #####
988 VectorModes:=proc(i)
989     local Imat;
990     global K,d;
991     Imat := convert(i-1, 'base', d);
992     while nops(Imat) <> K do Imat := [op(Imat), 0] end do;
993     return Imat;
994 end proc;
995
996
997 ##### #####
998 VectorRow:=proc(Indi, f::integer)
999     local Imat, runi, Nimat, Indix, x;
1000    global K,d;
1001    Indix := Indi;
1002    while nops(Indix) < K do Indix := [op(Indix), 0] end do;
1003    Nimat := nops(convert(Indix, base, f, 10));
1004    x := convert(Indix, base, f, 10);
1005    Imat := 1 + (sum(10^(rungi-1)*x[rungi], runi = 1 .. Nimat));
1006    return Imat;
1007 end proc;
1008
1009
1010 #####
1011 #####
1012
1013 ##### Declaration Procedures #####
1014
1015 #####
1016 #####
1017
1018 SqueezedVac:=proc(m, r, lambda)
1019     local V, i;
1020     global K, d;
1021     K := m;
1022     d := r;
1023     V := Matrix(r, 2);
1024     if m = 1 then
1025         for i to r do
1026             V[i, 1] := lambda^(i-1);
1027             V[i, 2] := [i-1]; end do;
1028         return V;
1029     elif m = 2 then
1030         for i to r do V[i, 1] := lambda^(i-1);
1031             V[i, 2] := [i-1, i-1];

```

```

1032      end do;
1033      return V;
1034 else
1035   print("is it a single mode or a 2 mode squeezed vacuum?")
1036 end if;
1037 end proc;
1038
1039
1040 #####      #####      #####      #####
1041
1042 Fock:=proc(i)
1043 local M;
1044 M:=Matrix([1,[i]]);
1045 return M;
1046 end proc;
1047
1048
1049 #####      #####      #####      #####
1050 CoherentState:=proc(m, r, alpha)
1051 local V, i;
1052
1053 global K, d;
1054 K := m;
1055
1056 d := r;
1057 V := Matrix(d, 2);
1058
1059 if m = 1 then
1060   for i to r do
1061     V[i, 1] := (alpha^(i - 1))/(sqrt(factorial(i - 1))) ;
1062     V[i, 2] := [i - 1]; end do;
1063
1064   return V;
1065 elif m = 2 then
1066
1067   for i to r do
1068     V[i, 1] := (alpha^(i - 1))/(sqrt(factorial(i - 1))) ;
1069     V[i, 2] := [i - 1, i - 1]; end do;
1070   return V;
1071
1072 elif m = 3 then
1073   for i to r do
1074     V[i, 1] := (alpha^(i - 1))/(sqrt(factorial(i - 1))) ;
1075     V[i, 2] := [i - 1, i - 1, i - 1]; end do;
1076

```

```

1077     return V;
1078   else
1079
1080     print("is it a single mode or a 2 mode squeezed vacuum?")
1081   end if;
1082
1083 end proc;
1084
1085 #####
1086 #####           #####           #####           #####
1087 IdentityState:=proc(Nrphot,NrModes)
1088   local Id1 , Id ,i ;
1089
1090   Id:=Matrix(Nrphot+1,3):
1091   for i from 1 to Nrphot+1 do
1092     Id [i ,1]:=1:
1093     Id [i ,2]:=[i -1];
1094     Id [i ,3]:=[i -1];
1095   od:
1096
1097   if NrModes=1 then
1098     return Id ;
1099   elif NrModes = 2 then
1100     Id:=TensorProduct(Id ,[1] ,Id ,[2]);
1101     return Id ;
1102   elif NrModes = 3 then
1103     Id1:=TensorProduct(Id ,[1] ,Id ,[2]);
1104     Id:=TensorProduct(Id1 ,[1 ,2] ,Id ,[3]);
1105     return Id ;
1106   elif NrModes>3 then
1107     print("for more than 3 modes, tensor two identity states")
1108   else
1109   fi:
1110 end proc:
1111
1112
1113 #####
1114 #####           #####           #####           #####
1115 Vac:=proc(Nmodes) local Lis;
1116   Lis:=[seq(0 ,i =1..Nmodes )];
1117   return Matrix([1 ,Lis ]);
1118 end proc:
1119 #####
1120 #####           #####           #####           #####
1121 TensorVac:=proc(M, m)

```

```

1122     local i , s ;
1123     global K, d ;
1124     K := findKnd(M)[1];
1125     d := findKnd(M)[2];
1126     K := K + m;
1127     if M[1, 1] <> 0 and LinearAlgebra:-ColumnDimension(M) = 2 then
1128         for i to LinearAlgebra:-RowDimension(M) do s := 1;
1129         while s <= m do
1130             M[i, 2] := [op(M[i, 2]), 0]; s := s + 1;
1131             end do;
1132         end do;
1133         return M;
1134     elif M[1, 1] <> 0 and LinearAlgebra:-ColumnDimension(M) = 3 then
1135         for i to LinearAlgebra:-RowDimension(M) do s := 1;
1136         while s <= m do M[i, 2] := [op(M[i, 2]), 0];
1137             M[i, 3] := [op(M[i, 3]), 0];
1138             s := s + 1;
1139             end do;
1140         end do;
1141         return M;
1142     elif M[1, 1] = 0 then
1143         for i from 2 to LinearAlgebra:-ColumnDimension(M) do s := 1;
1144         while s <= m do M[1, i] := [op(M[1, i]), 0];
1145             M[i, 1] := [op(M[i, 1]), 0];
1146             s := s + 1;
1147             end do;
1148         end do;
1149         return M;
1150     else
1151         print("is your state VEC, MAT or MATCOL well defined ?");
1152     end if;
1153 end proc;

1154
1155
1156 ######
1157 ######
1158
1159 ###### Translation Procedures #####
1160
1161 ######
1162 ######
1163
1164
1165 Trim:=proc(X)
1166     local s, i, Y, Indi;
```

```

1167 global d, K;
1168   if type(X, Vector) = true then
1169     s := 0;
1170
1171 #count the number of
1172 #non-zero entries
1173   for i to d^K do
1174     if X[i] <> 0 then
1175       s := s + 1
1176     else
1177     end if;
1178   end do;
1179 ##
1180   Y := Matrix(s, 2);
1181   s := 1;
1182   for i to d^K do if X[i] <> 0 then
1183     Indi := VectorModes(i);
1184     Y[s, 1] := X[i];
1185     Y[s, 2] := Indi;
1186     s := s + 1;
1187   else
1188   end if;
1189   end do;
1190   Y;
1191 elif type(X, Matrix) = true and LinearAlgebra:-ColumnDimension(X) = 2 then
1192   Y:=Matrix(X):
1193   Y:= Y[[seq('if '(Y[j,1]=0,NULL,j),j=1..LinearAlgebra:-RowDimension(Y))],1..2];
1194   return Y;
1195
1196 elif type(X, Matrix) = true and LinearAlgebra:-ColumnDimension(X)=3 then
1197   Y:=Matrix(X):
1198   Y:= Y[[seq('if '(Y[j,1]=0,NULL,j),j=1..LinearAlgebra:-RowDimension(Y))], 1..3];
1199   return Y;
1200   else
1201   end if;
1202 end proc;
1203
1204
1205
1206 ##### #####
1207 vec2mat:=proc(V::Matrix)
1208   local M, i, j, dimi;
1209   dimi := LinearAlgebra:-RowDimension(V);
1210   M := Matrix(dimi + 1, dimi + 1);
1211   for i from 2 to dimi + 1 do M[i, 1] := V[i - 1, 2] end do;

```

```

1212     for i from 2 to dimi + 1 do M[1, i] := V[i - 1, 2] end do;
1213     for i from 2 to dimi + 1 do for j from 2 to dimi + 1 do
1214     M[i, j] := V[i - 1, 1]*conjugate(V[j - 1, 1])
1215     end do;
1216   end do;
1217   M;
1218 end proc;
1219
1220
1221 #####      #####      #####      #####      #####
1222 vec2matcol:=proc(V::Matrix)
1223   local M, i, j;
1224   M := [];
1225   for i to LinearAlgebra:-RowDimension(V) do
1226     for j to LinearAlgebra:-RowDimension(V) do
1227       M:=[op(M),[V[i,1]*conjugate(V[j,1]),V[i,2], V[j,2]]]:
1228     end do;
1229   end do;
1230   convert(M,Matrix );
1231 end proc;
1232
1233
1234
1235 #####      #####      #####      #####      #####
1236 vec2poly:=proc(V::Matrix)
1237   local i, s, Poly;
1238   global K, a;
1239
1240   if whattype(a[1])<>indexed then
1241     print("unassign the mode variables a[1], a[2], etc before running vec2poly");
1242   else
1243
1244     findKnd(V):
1245     Poly := 0;
1246     for i to LinearAlgebra:-RowDimension(V) do
1247       Poly := V[i, 1]*(product((a[s]^V[i, 2][s])/
1248       (sqrt(factorial(V[i, 2][s]))), s = 1 .. K))
1249       + Poly;
1250     end do;
1251     Poly;
1252   fi :
1253 end proc;
1254
1255
1256 #####      #####      #####      #####      #####

```

```

1257 mat2matcol:=proc(M)
1258   local i, j, s, K;
1259   K := Matrix((LinearAlgebra:-RowDimension(M) - 1)^2, 3);
1260   s := 1;
1261   for i from 2 to LinearAlgebra:-ColumnDimension(M) do
1262     for j from 2 to LinearAlgebra:-RowDimension(M) do
1263       K[s, 1] := M[i, j];
1264       K[s, 2] := M[i, 1];
1265       K[s, 3] := M[1, j];
1266       s := s + 1;
1267     end do;
1268   end do;
1269   K := Trim(K);
1270 end proc;
1271
1272 #####
1273 mat2poly:=proc(M::Matrix)
1274   local i, j, s, dimi, Poly;
1275   global d,K;
1276
1277   Poly := 0;
1278   dimi := LinearAlgebra:-RowDimension(M);
1279
1280   if whattype(a[1])<>indexed then
1281     print("unassign mode variables a[1],a[2],b[1],b[2], etc before running vec2poly");
1282   else
1283
1284
1285     for i from 2 to dimi do
1286       for j from 2 to dimi do
1287         Poly := M[i, j]*(product(
1288           a[s]^M[i, 1][s]*b[s]^M[1, j][s])/(
1289             sqrt(factorial(M[i, 1][s])*factorial(M
1290               [1, j][s]))), s = 1 .. K)) + Poly;
1291     end do;
1292   end do;
1293   Poly;
1294 fi;
1295 end proc;
1296
1297 #####
1298 matcol2mat:=proc(M1)
1299   local i, j, s, ListNoR0l, halt, efofi, placed, M, Kout;

```

```

1302   findKnd(M1):
1303     M := StateSort(M1);
1304     ListNoR0l := [[M[1, 2], 1]];
1305     for i to LinearAlgebra:-RowDimension(M) - 1 do
1306       s := 1;
1307       halt := 0;
1308       while s <= nops(ListNoR0l) and halt = 0 do
1309         if M[i + 1, 2] = ListNoR0l[s, 1] then
1310           ListNoR0l[s, 2] := ListNoR0l[s, 2] + 1;
1311           halt := 1;
1312         else
1313           s := s + 1
1314         end if;
1315       end do;
1316       if nops(ListNoR0l) < s then
1317         ListNoR0l := [op(ListNoR0l), [M[i + 1, 2], 1]]
1318       else
1319         end if;
1320       end do;
1321       Kout := Matrix(nops(ListNoR0l) + 1, nops(ListNoR0l) + 1);
1322       for i from 2 to nops(ListNoR0l) + 1 do
1323         Kout[1, i] := ListNoR0l[i - 1, 1];
1324         Kout[i, 1] := ListNoR0l[i - 1, 1];
1325       end do;
1326       for i to nops(ListNoR0l) do
1327         efofi := add(ListNoR0l[t, 2], t = 1 .. i - 1);
1328         placed := 1;
1329         j := 2;
1330         while placed <= ListNoR0l[i, 2] do
1331           if M[efofi + placed, 3] = Kout[1, j] then
1332             Kout[i + 1, j] := M[efofi + placed, 1];
1333             placed := placed + 1;
1334           else
1335             j := j + 1
1336           end if;
1337         end do;
1338       end do;
1339     return Kout;
1340   end proc;
1341
1342
1343 #####      #####      #####      #####
1344 matcol2poly:=proc(M::Matrix)
1345   local i, s, dimi, Poly;
1346   global d,K;

```

```

1347
1348 findKnd(M):
1349
1350     Poly := 0;
1351     dimi := LinearAlgebra:-RowDimension(M);
1352     if whattype(a[1])<>indexed then
1353         print("unassign the mode variables a[1], a[2], etc before running matcol2poly");
1354     else
1355
1356         for i to dimi do Poly := M[i, 1]*(product((a[s]^M[i, 2][s]*b[s]^M[i, 3][s])/
1357             (sqrt(factorial(M[i, 2][s])*factorial(M[i, 3][s]))), s = 1 .. K)) + Poly
1358     end do;
1359     Poly;
1360     fi:
1361 end proc;
1362
1363
1364 #####      #####
1365 poly2vec:=proc(Poly)
1366     local i, Indimodes, Indi, terminospoly, cuantos, s, W;
1367     global d,K;
1368     ## define optical modes ##
1369     Indimodes:=[seq(a[i],i=1..K)];
1370     Indi:=[seq(0,i=1..K)];
1371
1372     if whattype(Poly)=`*` then
1373         ## the polynomial has only one term ##
1374         ## extract the coefficient ##
1375         terminospoly:=coeffs(Poly, Indimodes);
1376         ## extract the degree of each optical mode ##
1377         for s to K do
1378             Indi[s]:=degree(Poly,[a[s]]);
1379         od:
1380
1381     W:=Matrix([ sqrt(product(factorial(Indi[m]),m=1 .. K))*terminospoly , Indi ]);
1382     W:=simplify(W);
1383     return W;
1384
1385     elif whattype(Poly)=`+` then
1386         ## the polynomial has more than one term ##
1387         cuantos := nops(Poly);
1388         W := Matrix(cuantos, 2);
1389         terminospoly := op(Poly);
1390         for i to cuantos do
1391             for s to K do

```

```

1392      Indi[s] := degree(terminospoly[i], [a[s]]):
1393      end do;
1394      W[i, 1] := sqrt(product(factorial(Indi[m]), m = 1 .. K))*coeffs(
1395          terminospoly[i], Indimodes);
1396      W[i, 2] := Indi;
1397      end do;
1398      W := simplify(W);
1399      return W;
1400  else
1401      print("is your polynomial well defined in the modes a[1], a[2], etc..?");
1402
1403  fi:
1404 end proc;
1405
1406 #####
1407 ##### poly2matcol:=proc(Poly) #####
1408 ##### #####
1409 local M, i, s, Poly2, pol, Indimodes, Jndimodes, Indi, Jndi;
1410 global d,K;
1411 Indi:=[seq(0,i=1..K)]:
1412 Jndi:=[seq(0,i=1..K)]:
1413 Indimodes:=[seq(a[i],i=1..K)]:
1414 Jndimodes:=[seq(b[i],i=1..K)]:
1415
1416 if whattype(Poly)=`*` then
1417 ## the polynomial has only one term ##
1418 ## extract the coefficient ##
1419 pol:=coeffs(Poly, [op(Indimodes),op(Jndimodes)]);
1420 ## extract the degree of each optical mode ##
1421 for s to K do
1422 Indi[s]:=degree(Poly,[a[s]]):
1423 Jndi[s]:=degree(Poly,[b[s]]):
1424 od:
1425 ## reconstruct the matcol object ##
1426 M:=Matrix([ sqrt(product(factorial(Indi[m])*factorial(Jndi[m]),m=1 .. K))*pol , Indi, Jndi ]);
1427 M:=simplify(M):
1428 return M;
1429
1430
1431 elif whattype(Poly)=`+` then
1432
1433 Poly2 := collect(Poly, [op(Indimodes), op(Jndimodes)], `distributed `);
1434 pol := op(Poly2);
1435 M := Matrix(nops(Poly2), 3);
1436 for i to nops(Poly2) do

```

```

1437   for s to K do
1438     Indi[s] := degree(pol[i], [a[s]]);
1439     Jndi[s] := degree(pol[i], [b[s]]);
1440   end do;
1441   M[i, 1] := sqrt(product(factorial(Indi[m])*factorial(Jndi[m]),
1442 m = 1 .. K))*_
1443   coeffs(pol[i], [op(Jndimodes), op(Indimodes)]);
1444   M[i, 2] := Indi;
1445   M[i, 3] := Jndi;
1446 end do;
1447 M;
1448
1449 else
1450 print("is your polynomial well defined in the modes a[1],a[2],b[1],b[2], etc ?");
1451 fi;
1452 end proc;
1453
1454 #####
1455 indexstate:=proc(M)
1456 local Out;
1457 global K,d;
1458 findKnd(M):
1459 if LinearAlgebra:-ColumnDimension(M)=3 then
1460   Out:=Matrix(M):
1461   Out:=LinearAlgebra:-Map[(i,j)->evalb(j=2)](x->VectorRow(x,d),Out):
1462   Out:=LinearAlgebra:-Map[(i,j)->evalb(j=3)](x->VectorRow(x,d),Out):
1463   return Out:
1464 elif
1465
1466 elif
1467
1468 LinearAlgebra:-ColumnDimension(M)=2 then
1469   Out:=Matrix(M):
1470   Out:=LinearAlgebra:-Map[(i,j)->evalb(j=2)](x->VectorRow(x,d),Out):
1471   return Out:
1472
1473 elif
1474
1475 M[1,1]=0 then
1476   Out:=Matrix(M):
1477   Out:=LinearAlgebra:-Map[(i,j)->evalb(i=1)](x->VectorRow(x,d),Out):
1478   Out:=LinearAlgebra:-Map[(i,j)->evalb(j=1)](x->VectorRow(x,d),Out):
1479   Out[1,1]:=0:
1480   return Out:
1481

```

```

1482
1483   fi :
1484       end proc;
1485
1486
1487 #####      #####      #####      #####
1488 modesmatcol:=proc(M)
1489     local i, dimi, Karma;
1490     global d,K;
1491     dimi := LinearAlgebra:-RowDimension(M);
1492     Karma := Matrix(dimi, 3);
1493     for i to dimi do
1494         Karma[i, 1] := M[i, 1];
1495         Karma[i, 2] := VectorModes(M[i, 2]);
1496         Karma[i, 3] := VectorModes(M[i, 3]);
1497     end do;
1498     return Karma;
1499 end proc;
1500
1501
1502 #####      #####      #####      #####
1503 indexvec:=proc(M)
1504     local Out;
1505     Out:=Matrix(M):
1506     Out:=LinearAlgebra:-Map[(i,j)->evalb(j=2)](x->VectorRow(x,d),Out):
1507     return Out;
1508 end proc;
1509
1510
1511 #####      #####      #####      #####
1512
1513 modesvec:=proc(M)
1514     local Out;
1515     Out:=Matrix(M):
1516     Out:=LinearAlgebra:-Map[(i,j)->evalb(j=2)](x->VectorModes(x),Out):
1517     return Out;
1518 end proc;
1519
1520
1521 #####      #####      #####      #####
1522 #####      #####      #####      #####
1523
1524 #####      Linear Optics      #####
1525 #####      Quantum Operations      #####
1526

```

```

1527 #####
1528 #####
1529
1530 ## beam splitter##
1531
1532 BS:=proc(M::Matrix , m1, m2)
1533   local Out;
1534   global K, d;
1535   if M[1, 1] <> 0 and LinearAlgebra:-ColumnDimension(M) = 2 then
1536     Out := vecBS(M, m1, m2);
1537     K, d := findKnd(Out);
1538     print("d is now", d);
1539     return Out;
1540   elif M[1, 1] <> 0 and LinearAlgebra:-ColumnDimension(M) = 3 then
1541     Out := matcolBS(M, m1, m2);
1542     K, d := findKnd(Out);
1543     print("d is now", d);
1544     return Out;
1545   else
1546     print("Beam Splitter works with VEC and MATCOL")
1547     print("are they well defined?")
1548   end if;
1549 end proc;
1550
1551
1552 #####
1553 myBS:=proc(M::Matrix , m1, m2, t , r)
1554   local Out;
1555   global K, d;
1556   if M[1, 1] <> 0 and LinearAlgebra:-ColumnDimension(M) = 2 then
1557     Out := myvecBS(M, m1, m2, t , r);
1558     print("d is now", d);
1559     return Out;
1560   elif M[1, 1] <> 0 and LinearAlgebra:-ColumnDimension(M) = 3 then
1561     print("d is now", d);
1562     Out := mymatcolBS(M, m1, m2, t , r);
1563     K, d := findKnd(Out);
1564     return Out;
1565   else
1566     print("Beam Splitter works with VEC and MATCOL")
1567     print("are they well defined?")
1568   end if;
1569 end proc;
1570
1571

```

```

1572 ##### ##### ##### ##### ##### #####
1573 PS:=proc(M::Matrix , m1::integer , phi)
1574 local i , j , Indimodes , Jndimodes , Poly , Poly1 , Poly2 , Finalmatrix , Finalvector;
1575 global d , K;
1576 findKnd(M):
1577
1578 if M[1 , 1] <> 0 and LinearAlgebra:-ColumnDimension(M) = 2 then
1579 Indimodes := []:
1580 for i to K do Indimodes := [op(Indimodes) , a[i]] end do;
1581 Poly1 := vec2poly(M);
1582
1583 Poly := subs(a[m1] = exp(I*phi)*b[m1] , Poly1 );
1584 Poly := subs(b[m1] = a[m1] , Poly );
1585
1586 Poly := expand(map((x)->subs({a[m1]=exp(I*phi)*a[m1]} , x) , Poly1 )):
1587
1588 Poly := collect(Poly , Indimodes , 'distributed ');
1589 Finalvector := poly2vec(Poly);
1590 return Finalvector;
1591 elif M[1 , 1] <> 0 and LinearAlgebra:-ColumnDimension(M) = 3 then
1592 Indimodes := [];
1593 for i to K do Indimodes := [op(Indimodes) , a[i]] end do;
1594 Jndimodes := [];
1595 for j to K do Jndimodes := [op(Jndimodes) , b[i]] end do;
1596 Poly1 := matcol2poly(M);
1597 Poly := subs(a[m1] = exp(I*phi)*c[m1] , Poly1 );
1598 Poly := expand(subs(b[m1] = exp(-I*phi)*d[m1] , Poly));
1599 Poly := subs(c[m1] = a[m1] , Poly );
1600 Poly := subs(d[m1] = b[m1] , Poly );
1601 Poly2 := collect(Poly , [op(Indimodes) , op(Jndimodes)] , 'distributed ');
1602 Finalmatrix := poly2matcol(Poly2 );
1603 return Finalmatrix;
1604 else
1605 print("This PHASE SHIFTER works only with VEC and MATCOL");
1606 print("Is your state well defined ?");
1607 end if;
1608 end proc;
1609
1610
1611 ##### ##### ##### ##### ##### #####
1612 vecBS:=proc(Vec :: Matrix , m1, m2)
1613 myvecBS(Vec,m1,m2,t,r);
1614 end proc;
1615
1616

```

```

1617 #####      #####      #####      #####      #####
1618 myvecBS:=proc(Vec::Matrix , m1, m2, t , r)
1619     local Indimodes , Poly , Poly1 , Finalvector;
1620     global d,K;
1621     findKnd(Vec):
1622 ## declare optical modes
1623         Indimodes:=[seq(a[indice],indice=1..K)]:
1624 ## generate polynomial of modes
1625         Poly1 := vec2poly(Vec);
1626 ## substitute BS transformation
1627         Poly := expand(map((x)->
1628 subs({a[m1]=t*a[m1] + r*a[m2], a[m2]=-r*a[m1]+t*a[m2]},x), Poly1)):
1629         Poly := collect(Poly , Indimodes , 'distributed ' );
1630 ## convert back to vec object
1631         Finalvector := poly2vec(Poly);
1632 K, d :=findKnd(Finalvector):
1633 Finalvector:=StateSort(Finalvector):
1634 return Finalvector;
1635 end proc:
1636
1637
1638 #####      #####      #####      #####      #####
1639 matcolBS:=proc(M::Matrix , m1, m2)
1640     local i , Indimodes , Jndimodes , Poly , Poly1 , Poly2 , Finalmatrix;
1641     global d , K;
1642
1643 Indimodes:=[seq(a[i],i=1..K)]:
1644 Jndimodes:=[seq(b[i],i=1..K)]:
1645
1646 Poly1 := matcol2poly(M);
1647 Poly := expand(map((x)->
1648 subs({a[m1]=t*a[m1] + r*a[m2], a[m2]=-r*a[m1]+t*a[m2]},x),
1649 Poly1)):
1650 Poly2 := expand(map((x)->
1651 subs({b[m1]=conjugate(t)*b[m1] + conjugate(r)*b[m2],
1652 b[m2]=-conjugate(r)*b[m1]+conjugate(t)*b[m2]},x),
1653 Poly)):
1654
1655 Poly2 := collect(Poly2 , [op(Indimodes) , op(Jndimodes)], 'distributed ' );
1656 Finalmatrix := poly2matcol(Poly2 );
1657 return Finalmatrix;
1658 end proc;
1659
1660 #####      #####      #####      #####      #####
1661 mymatcolBS:=proc(M::Matrix , m1, m2,t,r)

```

```

1662 local i , Indimodes , Jndimodes , Poly , Poly1 , Poly2 , Finalmatrix ;
1663 global d,K;
1664
1665 Indimodes:=[seq(a[i],i=1..K)]:
1666 Jndimodes:=[seq(b[i],i=1..K)]:
1667
1668 Poly1 := matcol2poly(M);
1669 Poly := expand(map((x)→
1670 subs({a[m1]=t*a[m1] + r*a[m2], a[m2]=-r*a[m1]+t*a[m2]},x),
1671 Poly1)):
1672 Poly2 := expand(map((x)→
1673 subs({b[m1]=conjugate(t)*b[m1] + conjugate(r)*b[m2],
1674 b[m2]=-conjugate(r)*b[m1]+conjugate(t)*b[m2]},x),
1675 Poly)):
1676
1677 Poly2 := collect(Poly2 , [op(Indimodes), op(Jndimodes)], ‘distributed ’);
1678 Finalmatrix := poly2matcol(Poly2 );
1679 return Finalmatrix ;
1680 end proc;
1681
1682
1683 #####
1684 ##### BuildUnitary:=proc(Lys)
1685
1686 local t , j , M, Uni;
1687 global K, d;
1688
1689 if whattype(op(Lys[1]))<>exprseq then
1690 print("is your list of the form [[1,2,t,r]] or [[1,2,t,r],[1,phi]], etc ?");
1691 else
1692
1693 for t to nops(Lys) do M[t] := Matrix(K, K);
1694 for j to K do M[t][j, j] := 1 end do;
1695 if nops(Lys[t]) = 2 then
1696 M[t][Lys[t][1], Lys[t][1]] := exp(I*Lys[t][2]);
1697 elif nops(Lys[t]) = 3 then
1698 M[t][Lys[t][1], Lys[t][1]] := Lys[t][3];
1699 M[t][Lys[t][2], Lys[t][2]] := Lys[t][3];
1700 M[t][Lys[t][2], Lys[t][1]] := -simplify(sqrt(1 - Lys[t][3]^2));
1701 M[t][Lys[t][1], Lys[t][2]] := simplify(sqrt(1 - Lys[t][3]^2));
1702 elif nops(Lys[t]) = 4 then
1703 M[t][Lys[t][1], Lys[t][1]] := Lys[t][3];
1704 M[t][Lys[t][2], Lys[t][2]] := Lys[t][3];
1705 M[t][Lys[t][2], Lys[t][1]] := -Lys[t][4];
1706 M[t][Lys[t][1], Lys[t][2]] := Lys[t][4];

```

```

1707      else
1708          end if ;
1709      end do ;
1710      Uni := LinearAlgebra:=IdentityMatrix(K, K);
1711      for j to nops(Lys) do Uni := LinearAlgebra:=Multiply(Uni,M[j]) end do ;
1712      return Uni;
1713  fi ;
1714  end proc ;
1715
1716
1717 ##### #####
1718 UnitaryEvolution:=proc(U, M)
1719     local i,c, e,f,g,h,
1720     Po, Uli,
1721     Indimodes, Jndimodes,
1722     Modes, Modesket, Modesbra, Finalvector;
1723     global K, d;
1724
1725
1726 #####
1727 ##### for VEC #####
1728 #####
1729
1730
1731     if M[1, 1]  $\neq$  0 and LinearAlgebra:=ColumnDimension(M) = 2 then
1732
1733         Po := vec2poly(M);
1734
1735         Indimodes:=[seq(a[i],i=1..K)]:
1736
1737         Modes:=Vector(K):
1738         for i from 1 to K do
1739             Modes[i] := e[i];
1740         end do:
1741
1742         c:=LinearAlgebra:=Multiply(U, Modes):
1743
1744         for i from 1 to K do
1745             Po:=subs(a[i] = c[i], Po);
1746         od:
1747
1748         for i from 1 to K do
1749             Po:=subs(e[i]=a[i],Po);
1750         od:
1751

```

```

1752         Po := simplify(expand(Po));
1753         Po := collect(Po, Indimodes, 'distributed');
1754         Finalvector := poly2vec(Po);
1755         findKnd(Finalvector);
1756         print("d is now", d);
1757         return Finalvector;
1758
1759 #####
1760 ##### for MATCOL #####
1761 #####
1762
1763     elif M[1, 1] <> 0 and LinearAlgebra:-ColumnDimension(M) = 3 then
1764
1765     Uli:=Matrix(U);
1766     Po := matcol2poly(M);
1767
1768     Indimodes:=[seq(a[i], i=1..K)]:
1769     Jndimodes:=[seq(b[i], i=1..K)]:
1770
1771     ## create vector of modes ##
1772     Modesket:=Vector(K):
1773     Modesbra:=Vector(K):
1774
1775     for i from 1 to K do
1776         Modesket[i] := e[i]:
1777         Modesbra[i] := f[i]:
1778     end do:
1779
1780     ### apply unitary to modes ###
1781     g:=LinearAlgebra:-Multiply(Uli, Modesket);
1782     ## we take the complex conjugate (to conjugate phases, etc)
1783     ## however we transpose again, since it's applied to the
1784     ## vector of modes (and not to the conjugate of it)
1785     h:=LinearAlgebra:-Multiply(
1786     LinearAlgebra:-Transpose(
1787     LinearAlgebra:-HermitianTranspose(Uli, 'inplace'=false)), Modesbra);
1788
1789     ## substitute transformed modes in Polynomial ##
1790     for i from 1 to K do
1791     Po:=subs(a[i] = g[i], Po);
1792     Po:=subs(b[i] = h[i], Po);
1793     od:
1794
1795     ## substitute back to original names ##
1796     for i from 1 to K do

```

```

1797 Po:=subs(e[i]=a[i],Po);
1798 Po:=subs(f[i]=b[i],Po);
1799 od;
1800
1801 Po := simplify(expand(Po));
1802
1803 Po := collect(Po, [op(Indimodes),op(Jndimodes)], 'distributed');
1804
1805
1806         Finalvector := poly2matcol(Po);
1807         findKnd(Finalvector);
1808         print("d is now", d);
1809
1810 return Finalvector;
1811
1812 else
1813     print("this procedure works so far for VEC and MATCOL");
1814 end if;
1815 end proc;
1816
1817
1818
1819 #####
1820 #####
1821
1822 #####                                     #####
1823 #####                                     ## Measurements ##                                     #####
1824
1825 #####
1826 #####
1827
1828
1829 APD:=proc(result,eta,Nrphot) local i,
1830 POVM;
1831
1832 if is(Nrphot,integer)=false then
1833 print("is the number of photons an integer?");
1834 else
1835 fi;
1836 if result=0 then
1837 POVM:=Matrix(Nrphot+1,3):
1838 for i from 1 to Nrphot+1 do
1839 POVM[i,1]:=eta^(2*(i-1)):
1840 POVM[i,2]:=[i-1];
1841 POVM[i,3]:=[i-1];

```

```

1842 od:
1843 return POVM;
1844
1845 elif result=1 then
1846
1847 POVM:=Matrix (Nrphot ,3):
1848 for i from 1 to Nrphot do
1849 POVM[ i ,1]:=1 - eta^(2*i):
1850 POVM[ i ,2]:=[ i ]:
1851 POVM[ i ,3]:=[ i ]:
1852 od:
1853 return POVM;
1854
1855 else
1856 print("is your result 0 photons or any photons ?");
1857 fi:
1858
1859 end proc:
1860
1861
1862 #####      #####
1863
1864 Project:=proc(M1, L, M2)
1865 local projecteur, rho;
1866 if LinearAlgebra:-ColumnDimension(M1) = 2 and
1867 LinearAlgebra:-ColumnDimension(M2 ) = 2 and M2[1, 1] <> 0 then
1868 Projectvecvec(M1, L, M2)
1869 elif LinearAlgebra:-ColumnDimension(M1) = 2 and
1870 LinearAlgebra:-ColumnDimension(M2) = 3 and M2[1, 1] <> 0 then
1871 projecteur := Quantavo:-vec2matcol(M1);
1872 Projectmatcol(projecteur, L, M2);
1873 elif LinearAlgebra:-ColumnDimension(M1) = 3 and
1874 LinearAlgebra:-ColumnDimension(M2) = 2 and M2[1, 1] <> 0 then
1875 rho := Quantavo:-vec2matcol(M2);
1876 Projectmatcol(M1, L, rho);
1877 elif LinearAlgebra:-ColumnDimension(M1) = 3 and
1878 LinearAlgebra:-ColumnDimension(M2) = 3 and M2[1, 1] <> 0 then
1879 Projectmatcol(M1, L, M2)
1880 else
1881 print("this procedure takes (vec/matcol,list ,vec/matcol) as inputs");
1882 print("are your states well defined ?");
1883 end if;
1884 end proc;
1885
1886 #####      #####

```

```

1887 Projectvecvec:=proc(psi::Matrix,L,V::Matrix)
1888   local Ket, W, Outtemp, projector, equal, s, i, j;
1889   projector := Quantavo:-vec2matcol(psi);
1890   W := [];
1891
1892   for i to LinearAlgebra:-RowDimension(projector) do
1893     for j to LinearAlgebra:-RowDimension(V) do
1894       equal := 1;
1895
1896       for s to nops(L) do
1897         if projector[i, 3][s] = V[j, 2][L[s]] then
1898           else
1899             equal := 0;
1900           break;
1901         end if;
1902       end do;
1903
1904       if equal = 1 then
1905         Ket := V[j, 2];
1906
1907         for s to nops(L) do
1908           Ket := subsop(L[s] = projector[i, 2][s], Ket)
1909         end do;
1910
1911         W:=[ op(W),[ projector[i,1]*V[j,1],Ket ]];
1912
1913       else
1914         end if;
1915       end do;
1916     end do;
1917     Outtemp := convert(W,Matrix);
1918
1919     Outtemp:=StateSort(Outtemp);
1920     return Outtemp;
1921   end proc;
1922
1923
1924 #####      #####
1925 Projectmatcol:=proc(psi::Matrix,L,rho::Matrix)
1926
1927   local Ket, equal,W,
1928   Outtemp, psibar, Out,
1929   s,i,j;
1930
1931   W:=[];

```

```

1932 for i from 1 to LinearAlgebra:-RowDimension(psi) do
1933   for j from 1 to LinearAlgebra:-RowDimension(rho) do
1934
1935   #####compare bra from POVM and ket from density op.#####
1936   equal:=1:
1937   for s from 1 to nops(L) do
1938     if psi[i,3][s]=rho[j,2][L[s]] then
1939       #they are the same#
1940     else
1941       equal:=0:
1942     break;
1943   fi:
1944   od:
1945
1946
1947   if equal = 1 then
1948     Ket:=rho[j,2]:
1949
1950     for s from 1 to nops(L) do
1951       Ket:=subsop(L[s]=psi[i,2][s],Ket):
1952     od:
1953
1954 W:=[op(W),[psi[i,1]*rho[j,1],Ket,rho[j,3]]];
1955
1956 else
1957   fi:
1958
1959 od:
1960 od:
1961
1962 Outtemp:=convert(W,Matrix):
1963 psibar:=StateComplexConjugate(psi):
1964
1965 W:=[];
1966 for i from 1 to LinearAlgebra:-RowDimension(psibar) do
1967   for j from 1 to LinearAlgebra:-RowDimension(Outtemp) do
1968
1969   #####compare ket from adjoint POVM and bra from density op.#####
1970   equal:=1:
1971   for s from 1 to nops(L) do
1972     if psibar[i,2][s]=Outtemp[j,3][L[s]] then
1973       #they are the same#
1974     else
1975       equal:=0:
1976     break;

```

```

1977 fi :
1978 od :
1979
1980
1981 if equal = 1 then
1982     Ket:=Outtemp[j,3]:
1983
1984     for s from 1 to nops(L) do
1985         Ket:=subsop(L[s]=psibar[i,3][s],Ket):
1986     od :
1987 W:=[op(W),[ psibar[i,1]*Outtemp[j,1],Outtemp[j,2],Ket ]];
1988
1989
1990 else
1991 fi :
1992
1993 od :
1994 od :
1995
1996 if nops(W) = 0 then
1997     print("there is no overlap, your state is zero after this measurement")
1998 else
1999     Out:=convert(W,Matrix):
2000     ##### Sort and add repeated entries #####
2001     Out:=StateSort(Out);
2002     return Out;
2003 fi
2004 end proc:
2005
2006
2007 #####      #####      #####      #####      #####
2008 POVMresult:=proc(psi::Matrix,L,rho::Matrix)
2009
2010 local Ket, equal ,W,
2011 Outtemp , Out ,
2012 Todelete , same , Lordered ,
2013 s,i,j ;
2014
2015 findKnd(rho):
2016
2017 #####      #####      #####      #####
2018 ##### for POVM and density matrix #####
2019 #####      #####      #####      #####
2020 #####      #####      #####      #####
2021

```

```

2022 if LinearAlgebra:-ColumnDimension(psi)=3 and
2023   LinearAlgebra:-ColumnDimension(rho)=3 then
2024 W:=[];
2025 for i from 1 to LinearAlgebra:-RowDimension(psi) do
2026 for j from 1 to LinearAlgebra:-RowDimension(rho) do
2027
2028 #####compare bra from POVM and ket from density op#####
2029 equal:=1;
2030 for s from 1 to nops(L) do
2031 if psi[i,3][s]=rho[j,2][L[s]] then
2032 #they are the same#
2033 else
2034 equal:=0;
2035 break;
2036 fi;
2037 od;
2038
2039
2040 if equal = 1 then
2041   Ket:=rho[j,2];
2042
2043   for s from 1 to nops(L) do
2044     Ket:=subsop(L[s]=psi[i,2][s],Ket);
2045   od;
2046
2047 W:=[op(W),[psi[i,1]*rho[j,1],Ket,rho[j,3]]];
2048
2049 else
2050 fi;
2051
2052 od;
2053 od;
2054
2055
2056 Outtemp:=Trim(convert(W,Matrix));
2057
2058
2059 ##### in case no state is left #####
2060 if Outtemp=NULL or Dimensions(Outtemp)[1]=0 then
2061 print("No state is left");
2062 else
2063
2064
2065 ##### tracing out measured modes #####
2066

```

```

2067 Todelete := []:
2068 for i from 1 to LinearAlgebra:-RowDimension(Outtemp) do
2069 same := 0:
2070 for s from 1 to nops(L) while same=0 do
2071 if Outtemp[i,2][L[s]] = Outtemp[i,3][L[s]] then
2072 else
2073 same := 1:
2074 fi:
2075 od:
2076
2077 if same=0 then
2078 else
2079 Todelete := [op(Todelete), i]:
2080 fi:
2081
2082 od:
2083
2084 Outtemp := LinearAlgebra:-DeleteRow(Outtemp, Todelete);
2085
2086
2087
2088 ##### delete mode elements #####
2089 Lordered := sort(L, '>');
2090 for s from 1 to nops(L) do
2091 Outtemp := LinearAlgebra:-Map[(i,j)->evalb(j=2)](x->subsop(Lordered[s]=NULL,x),Outtemp);
2092 Outtemp := LinearAlgebra:-Map[(i,j)->evalb(j=3)](x->subsop(Lordered[s]=NULL,x),Outtemp);
2093 od;
2094
2095 ##### sort (which will also add repeated elements) #####
2096
2097
2098 Outtemp := StateSort(Outtemp):
2099 return Outtemp;
2100 fi:
2101
2102 ##### for POVM and pure state #####
2103 #####
2104 #####
2105 #####
2106
2107 elif LinearAlgebra:-ColumnDimension(psi)=3 and
2108 LinearAlgebra:-ColumnDimension(rho)=2 then
2109
2110 Out := vec2matcol(rho):
2111

```

```

2112 W:=[];
2113 for i from 1 to LinearAlgebra:-RowDimension(psi) do
2114 for j from 1 to LinearAlgebra:-RowDimension(Out) do
2115
2116 #####compare bra from POVM and ket from density op#####
2117 equal:=1;
2118 for s from 1 to nops(L) do
2119 if psi[i,3][s]=Out[j,2][L[s]] then
2120 #they are the same#
2121 else
2122 equal:=0;
2123 break;
2124 fi;
2125 od;
2126
2127
2128 if equal = 1 then
2129 Ket:=Out[j,2];
2130
2131 for s from 1 to nops(L) do
2132 Ket:=subsop(L[s]=psi[i,2][s],Ket);
2133 od;
2134
2135 W:=[op(W),[psi[i,1]*Out[j,1],Ket,Out[j,3]]];
2136
2137 else
2138 fi;
2139
2140 od;
2141 od;
2142
2143 Outtemp:=Trim(convert(W,Matrix));
2144
2145
2146 ##### in case no state is left #####
2147 if Outtemp=NULL or Dimensions(Outtemp)[1]=0 then
2148 print("No state is left");
2149 else
2150
2151
2152
2153 ##### tracing out measured modes #####
2154
2155 Todelete:=[];
2156 for i from 1 to LinearAlgebra:-RowDimension(Outtemp) do

```



```

2202 Probability:=proc(psi::Matrix,L,rho::Matrix)
2203
2204 local Ket, equal,W, POV, Out,
2205 Sumdiag, Prob,
2206 s,i,j;
2207
2208 ##### choose vec/matcol and convert if needed #####
2209
2210 if psi[1,1]=0 or rho[1,1]=0 then
2211 print("this procedure accepts VEC/MATCOL as input, is your state well defined?")
2212
2213 ##### for vec POVM and vec state #####
2214 elif LinearAlgebra:-ColumnDimension(psi)=2 and
2215 LinearAlgebra:-ColumnDimension(rho)=2 then
2216 POV:=vec2matcol(psi);
2217 Out:=vec2matcol(rho);
2218
2219 ##### for vec POVM and vec state #####
2220 elif LinearAlgebra:-ColumnDimension(psi)=2 and
2221 LinearAlgebra:-ColumnDimension(rho)=3 then
2222 POV:=vec2matcol(psi);
2223 Out:=Matrix(rho);
2224
2225 ##### for POVM and pure state #####
2226 elif LinearAlgebra:-ColumnDimension(psi)=3 and
2227 LinearAlgebra:-ColumnDimension(rho)=2 then
2228 POV:=Matrix(psi);
2229 Out:=vec2matcol(rho);
2230
2231 ##### for POVM and density matrix #####
2232 elif LinearAlgebra:-ColumnDimension(psi)=3 and
2233 LinearAlgebra:-ColumnDimension(rho)=3 then
2234 POV:=Matrix(psi);
2235 Out:=Matrix(rho);
2236 else
2237 print("are the POVM and State well defined?");
2238 print("this procedure handles POVM (matcol), State (matcol/vec)");
2239 fi:
2240 ##########
2241
2242 W:=[];
2243 for i from 1 to LinearAlgebra:-RowDimension(POV) do
2244 for j from 1 to LinearAlgebra:-RowDimension(Out) do
2245
2246 #####compare bra from POVM and ket from density op.#####

```

```

2247 equal:=1:
2248 for s from 1 to nops(L) do
2249 if POV[i,3][s]=Out[j,2][L[s]] then
2250 #they are the same#
2251 else
2252 equal:=0: #they are different#
2253 break;
2254 fi:
2255 od:
2256
2257
2258 if equal = 1 then
2259 Ket:=Out[j,2]:
2260
2261 for s from 1 to nops(L) do
2262 Ket:=subsop(L[s]=POV[i,2][s],Ket):
2263 od:
2264
2265 W:=[op(W),[POV[i,1]*Out[j,1],Ket,Out[j,3]]];
2266
2267 else
2268 fi:
2269
2270 od:
2271 od:
2272
2273 ##### Tr(Pi.rho) taking the trace #####
2274
2275 Sumdiag:=0:
2276 for i from 1 to nops(W) do
2277
2278 if W[i][2]=W[i][3] then
2279 Sumdiag:=Sumdiag+W[i][1];
2280 else
2281 fi:
2282 od:
2283
2284 ##### final prob #####
2285
2286 Prob:=Sumdiag/StateTrace(Out); #make sure state is normalized
2287
2288
2289 if Prob<>1 then
2290 print("remember to check that your POVM elements are Positive");
2291 print("and that they add-up to the Identity");

```

```

2292 else
2293   fi :
2294
2295   return Prob;
2296
2297 end proc:
2298
2299 ##########
2300 ##########
2301 ##########
2302 ########## #####
2303 ########## ## Display Procedures ## #####
2304
2305 ##########
2306 ##########
2307
2308 Dket:=proc(Indi)
2309   local s, ket;
2310   global d, K;
2311   ket := '|';
2312   for s to K do ket := cat(ket, Indi[s]) end do;
2313   ket := cat(ket, '>');
2314   ket;
2315 end proc;
2316
2317
2318 ##### ##### ##### ##### ##### #####
2319 Dbra:=proc(Jndi)
2320   local s, bra;
2321   global d, K;
2322   bra := '<';
2323   for s to K do bra := cat(bra, Jndi[s]) end do;
2324   bra := cat(bra, '|');
2325   bra;
2326 end proc;
2327
2328
2329 ##### ##### ##### ##### ##### #####
2330 Dbaket:=proc(Indi, Jndi)
2331   local s, braket;
2332   global d, K;
2333   braket := '|';
2334   for s to K do braket := cat(braket, Indi[s]) end do;
2335   braket := cat(braket, '><');
2336   for s to K do braket := cat(braket, Jndi[s]) end do;

```

```

2337         braket := cat(braket, '|');
2338         braket;
2339     end proc;
2340
2341
2342 #####      #####
2343 Dstate:=proc(M)
2344     local Sa;
2345 ## find K and d ##
2346 findKnd(M):
2347     if M[1, 1] <> 0 and LinearAlgebra:-ColumnDimension(M) = 2 then
2348         Sa := Dvec(M);
2349         return Sa;
2350     elif M[1, 1] <> 0 and LinearAlgebra:-ColumnDimension(M) = 3 then
2351         Sa := Dmatcol(M);
2352         return Sa;
2353     elif M[1, 1] = 0 and LinearAlgebra:-ColumnDimension(M) =
2354         LinearAlgebra:-RowDimension(M) then
2355         Sa := Dmat(M);
2356         return Sa;
2357     else
2358         print("is your object VEC, MAT or MATCOL well defined?")
2359     end if;
2360 end proc;
2361
2362
2363 #####      #####
2364 Dvec:=proc(V::Matrix)
2365     local dimi, i, Y;
2366     global d, K;
2367     dimi := LinearAlgebra:-RowDimension(V);
2368     interface(rtablesize = dimi + 10);
2369     Y := Matrix(dimi, 3);
2370     for i to dimi do Y[i, 1] := V[i, 1];
2371         Y[i, 2] := '      ';
2372         Y[i, 3] := Dket(V[i, 2]);
2373     end do;
2374     Y;
2375 end proc;
2376
2377
2378 #####      #####
2379 Dmat:=proc(M)::Matrix)
2380     local Moe, i, j, dimi;
2381     dimi := LinearAlgebra:-RowDimension(M);

```

```

2382     interface(rtablesize = dimi + 40);
2383     Moe := Matrix(dimi, dimi);
2384     for i from 2 to dimi do Moe[1, i] := Dbra(M[1, i]);
2385     Moe[i, 1] := Dket(M[i, 1]);
2386     end do;
2387     for i from 2 to dimi do for j from 2 to dimi do Moe[i, j] := M[i, j]
2388     end do;
2389     end do;
2390     Moe;
2391   end proc;
2392
2393
2394 #####      #####
2395 Dmatcol:=proc(M::Matrix)
2396   local i, dimi, Y;
2397   dimi := LinearAlgebra:-RowDimension(M);
2398   interface(rtablesize = dimi + 10);
2399   Y := Matrix(dimi, 3);
2400   for i to dimi do Y[i, 1] := M[i, 1];
2401   Y[i, 2] := ' ';
2402   Y[i, 3] := Dbraket(M[i, 2], M[i, 3]);
2403   end do;
2404   Y;
2405   end proc;
2406
2407 #####
2408 #####
2409 #####      ## Ploting Procedures ##
2410
2411 #####
2412 PlotState:=proc(M, h, H)
2413   local i, L, V1, WW, Mat;
2414   global d;
2415   findKnd(M):
2416     if M[1, 1]  $\neq$  0 and LinearAlgebra:-ColumnDimension(M) = 2 then
2417       V1 := Matrix(M);
2418       WW := NULL;
2419       for i to LinearAlgebra:-RowDimension(V1) do WW := WW,
2420         [VectorRow(V1[i, 2], d), V1[i, 1]];
2421       end do;
2422       return histo([WW]);
2423     elif M[1, 1]  $\neq$  0 and LinearAlgebra:-ColumnDimension(M) = 3 then
2424       Mat := indexstate(M);
2425       L := NULL;
2426       for i to LinearAlgebra:-RowDimension(Mat) do

```

```

2427      L := L, geom3d:-draw(barra(Mat, i, h, H))
2428  end do;
2429  return plots:-display3d({L}, axes = boxed);
2430 elif M[1, 1] = 0 then
2431   Mat := Matrix(M);
2432   Mat := mat2matcol(Mat);
2433   Mat := indexstate(Mat);
2434   L := NULL;
2435   for i to LinearAlgebra:-RowDimension(Mat) do L := L,
2436     geom3d:-draw(barra(Mat, i, h, H))
2437   end do;
2438   return plots:-display3d({L}, axes = boxed);
2439 else
2440   print("is your mat/matcol well defined? are amplitudes positive numbers?")
2441 end if;
2442 end proc;
2443
2444
2445 #####
2446 #####          #####          #####          #####
2447 barra:=proc(M, i, h, H)
2448 local alpha, beta, gama, delta;
2449 geom3d:-point(alpha, M[i, 2] + h, M[i, 3] - h, 0),
2450 geom3d:-point(beta, M[i, 2] + h, M[i, 3] + h, 0),
2451 geom3d:-point(gama, M[i, 2] - h, M[i, 3] - h, 0),
2452 geom3d:-point(delta, M[i, 2] + h, M[i, 3] - h, H*M[i, 1]);
2453 geom3d:-dsegment(d1, [alpha, beta]), geom3d:-dsegment(d2, [alpha, gama]),
2454 geom3d:-dsegment(d3, [alpha, delta]);
2455 return geom3d:-parallelepiped(pp, [d1, d2, d3]);
2456 end proc;
2457
2458
2459 #####
2460 histo:=proc(L::list)
2461 local k, poly, S;
2462 S := NULL;
2463 for k in L do poly := [[k[1], 0], [k[1] + 1, 0], [k[1] + 1, k[2]],
2464 [k[1], k[2]]];
2465 S := S,
2466 plots[polygonplot](poly, color = COLOR(RGB, 0.1960, 0.6000, 0.8000));
2467 end do;
2468 plots[display]({S});
2469 end proc;
2470
2471 #####

```

```
2472 ##### ##### ##### ##### #####  
2473 end module:  
2474 ##### ##### ##### ##### #####  
2475 ##### ##### ##### ##### #####
```

Bibliography

- [ABJ⁺03] J. B. Altepeter, D. Branning, E. Jeffrey, T. C. Wei, P. G. Kwiat, R. T. Thew, J. L. O'Brien, M. A. Nielsen, and A. G. White. Ancilla-assisted quantum process tomography. *Phys. Rev. Lett.*, 90(19):193601, 2003.
- [ACRL⁺08] M. Avenhaus, H. B. Coldenstrodt-Ronge, K. Laiho, W. Mauerer, I. A. Walmsley, and C. Silberhorn. Photon number statistics of multimode parametric down-conversion. *Physical Review Letters*, 101:053601, 2008.
- [ADB05] H. Aschauer, W. Dür, and H. J. Briegel. Multiparticle entanglement purification for two-colorable graph states. *Physical Review A*, 71:012319, 2005.
- [ADMS95] Arvind, B. Dutta, N. Mukunda, and R. Simon. Two-mode quantum systems: Invariant classification of squeezing transformations and squeezed states. *Phys. Rev. A*, 52(2):1609–1620, Aug 1995.
- [ADR82] A. Aspect, J. Dalibard, and G. Roger. Experimental test of bell's inequalities using time- varying analyzers. *Phys. Rev. Lett.*, 49(25):1804–1807, 1982.
- [AEI07] G. Adesso, M. Ericsson, and F. Illuminati. Coexistence of unlimited bipartite and genuine multipartite entanglement: Promiscuous quantum correlations arising from discrete to continuous-variable systems. *Physical Review A (Atomic, Molecular, and Optical Physics)*, 76(2):022315, 2007.
- [AFW⁺06] O Alibart, J Fulconis, G K L Wong, S G Murdoch, W J Wadsworth, and J G Rarity. Photon pair generation using four-wave mixing in a microstructured fibre: theory versus experiment. *New Journal of Physics*, 8(5):67, 2006.

- [AGL⁺04] U. L. Andersen, O. Glöckl, S. Lorenz, G. Leuchs, and R. Filip. Experimental demonstration of continuous variable quantum erasing. *Phys. Rev. Lett.*, 93(10):100403, Sep 2004.
- [AGR82] A. Aspect, P. Grangier, and G. Roger. Experimental realization of einstein-podolsky-rosen-bohm gedankenexperiment: A new violation of bell's inequalities. *Phys. Rev. Lett.*, 49(2):91–94, 1982.
- [AI06] G. Adesso and F. Illuminati. Continuous variable tangle, monogamy inequality, and entanglement sharing in gaussian states of continuous variable systems. *New Journal of Physics*, 8:15, 2006.
- [AI07] G. Adesso and F. Illuminati. Strong monogamy of bipartite and genuine multipartite entanglement: The gaussian case. *Physical Review Letters*, 99:150501, 2007.
- [AI08] G. Adesso and F. Illuminati. Genuine multipartite entanglement of gaussian states: Strong monogamy, unitary localization, scaling behavior, and molecular sharing structure. *Arxiv.org*, 2008.
- [AJWT⁺03] Vaziri Alipasha, Pan Jian-Wei, Jennewein Thomas, Weihs Gregor, and Zeilinger Anton. Concentration of higher dimensional entanglement: Qutrits of photon orbital angular momentum. *Phys. Rev. Lett.*, 91(22):227902, 2003.
- [AMS97] Arvind, N. Mukunda, and R. Simon. Gaussian-wigner distributions and hierarchies of nonclassical states in quantum optics: The single-mode case. *Phys. Rev. A*, 56(6):5042–5052, 1997.
- [AOBG07] H. Jeong A. Ourjoumtsev, R. Tualle Brouri, and P. Grangier. Generation of optical *Schrödinger cats* from photon number states. *Nature*, 448:784, 2007.
- [AP06] K. M. R. Audenaert and M. B. Plenio. When are correlations quantum? – verification and quantification of entanglement by simple measurements. *New Journal of Physics*, 8:266, 2006.
- [APE03] K. Audenaert, M. B. Plenio, and J. Eisert. Entanglement cost under positive-partial-transpose-preserving operations. *Phys. Rev. Lett.*, 90:027901, 2003.

- [APR81] A. Aspect, Grangier P., and G. Roger. Experimental tests of realistic local theories via bell's theorem. *Phys. Rev. Lett.*, 47(7):460–463, Aug 1981.
- [AS08] Koenraad M. R. Audenaert and S. Scheel. Quantum tomographic reconstruction with error bars: a kalman filter approach. *arXiv.org:0809.3359*, 2008.
- [ASS⁺03a] D. Achilles, C. Silberhorn, C. Sliwa, K. Banaszek, and I. A. Walmsley. Fiber-assisted detection with photon number resolution. *Optics Letters*, 28:2387, 2003.
- [ASS⁺03b] D. Achilles, C. Silberhorn, C. Sliwa, K. Banaszek, I. A. Walmsley, M. J. Fitch, B. C. Jacobs, T. B. Pittman, and J. D. Franson. Photon number resolving detection using time-multiplexing. *arXiv.org*, quant-ph/0310183v1, 2003.
- [ASW06] D. Achilles, C. Silberhorn, and I. A. Walmsley. Direct, loss-tolerant characterization of nonclassical photon statistics. *Physical Review Letters*, 97:043602, 2006.
- [AZ04] M. Bellini A. Zavatta, S. Viciani. Quantum-to-classical transition with single-photon-added coherent states of light. *Science*, 306(5696):660 – 662, 2004.
- [BAS⁺06] A. J. Bennett, P. Atkinson, P. See, M. B. Ward, R. M. Stevenson, Z. L. Yuan, D. C. Unitt, D. J. P. Ellis, K. Cooper, D. A. Ritchie, and A. J. Shields. Single-photon-emitting diodes: a review. *PHYS.STAT.SOL*, 243:3730, 2006.
- [BBC⁺93a] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters. Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels. *Phys. Rev. Lett.*, 70:1895, 1993.
- [BBC⁺93b] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters. Teleporting an unknown quantum state via dual classical and einstein-podolsky-rosen channels. *Phys. Rev. Lett.*, 70:1895–1899, 1993.
- [BBP⁺96] C. H. Bennett, G. Brassard, S. Popescu, B. Schumacher, J. A. Smolin, and W. K. Wootters. Purification of noisy entanglement and faithful teleportation via noisy channels. *Phys. Rev. Lett.*, 76:722, 1996.
- [BBPS96] C. H. Bennett, H. Bernstein, S. Popescu, and B. Schumacher. Concentrating partial entanglement by local operations. *Phys. Rev. A*, 53:2046, 1996.

- [BDSW96] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, and W. K. Wootters. Mixed-state entanglement and quantum error correction. *Phys. Rev. A*, 54(5):3824–3851, 1996.
- [BEE00] D. Bouwmeester, A. Ekert, and A. Zeilinger (Eds.). *The Physics of Quantum Information*. Springer-Verlag, 2000.
- [Bel64] J. S. Bell. On the Einstein-Podolsky-Rosen paradox. *Physics*, 1:195, 1964. Reprinted in J. S. Bell “Speakable and unspeakable in quantum mechanics”, Cambridge University Press, 1987.
- [Bel87] J. S. Bell. *Speakable and unspeakable in quantum mechanics: collected papers on quantum philosophy*. CAMBRIDGE, 1987.
- [BESP03] D.E. Browne, J. Eisert, S. Scheel, and M.B. Plenio. Driving non-gaussian to gaussian states with linear optics. *Physical Review A*, 67:062320, 2003.
- [BHPC03] N. Boulant, T. F. Havel, M. A. Pravia, and D. G. Cory. Robust method for estimating the lindblad operators of a dissipative quantum process from measurements of the density operator at multiple time points. *Phys. Rev. A*, 67(4):042322, 2003.
- [BL05] S. L. Braunstein and P. Van Loock. Quantum information with continuous variables. *Reviews of Modern Physics*, 77:513–577, 2005.
- [Boh35] N. Bohr. Can quantum-mechanical description of physical reality be considered complete? *Phys. Rev.*, 48(8):696–702, 1935.
- [BPD⁺99] D. Bouwmeester, J.W. Pan, M. Daniell, H. Weinfurter, and A. Zeilinger. Observation of three-photon greenberger-horne-zeilinger entanglement. *Phys. Rev. Lett.*, 82(7):1345–1349, 1999.
- [Bra07] F.G.S.L. Brandão. private communication, 2007.
- [Bro05] D. E. Browne. *Generation and Manipulation of Entanglement in Quantum Optical Systems*. PhD thesis, 2005.

- [BRWKan99] K. Banaszek, C. Radzewicz, K. Wódkiewicz, and J. S. Krasiński. Direct measurement of the wigner function by photon counting. *Phys. Rev. A*, 60(1):674–677, 1999.
- [BS02] S. D. Bartlett and B. C. Sanders. Universal continuous-variable quantum computation: Requirement of optical nonlinearity for photon counting. *Physical Review A*, 65:042304, 2002.
- [BV04] Stephen B. and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [BVK98] S. Bose, V. Vedral, and P. L. Knight. Multiparticle generalization of entanglement swapping. *Phys. Rev. A*, 57(2):822–829, 1998.
- [BW92] C. H. Bennett and S. J. Wiesner. Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states. *Phys. Rev. Lett.*, 69:2881, 1992.
- [BW99a] K. Banaszek and K. Wódkiewicz. Nonlocality of the einstein-podolsky-rosen state in the phase space. *ACTA PHYS.SLOV.*, 49:491, 1999.
- [BW99b] K. Banaszek and K. Wódkiewicz. Testing quantum nonlocality in phase space. *Phys. Rev. Lett.*, 82(10):2009–2013, 1999.
- [C.98] Macchiavello C. On the analytical convergence of the qpa procedure. *Physics Letters A*, 246:385–388(4), 1998.
- [CG69] K. E. Cahill and R. J. Glauber. Density operators and quasiprobability distributions. *Phys. Rev.*, 177(5):1882–1902, 1969.
- [CHSH69] J. F. Clauser, M. A. Horne, A. Shimony, and R. A. Holt. Proposed experiment to test local hidden-variable theories. *Phys. Rev. Lett.*, 23(15):880–884, 1969.
- [CKW00] V. Coffman, J. Kundu, and W. K. Wootters. Distributed entanglement. *Phys. Rev. A*, 61(5):052306, 2000.
- [CLP01] D. Collins, N. Linden, and S. Popescu. The non-local content of quantum operations. *Phys. Rev. A*, 64:032302, 2001.

- [CLS⁺08] O. Cohen, J. S. Lundeen, B. J. Smith, G. Puentes, P. J. Mosley, and I. A. Walmsley. Tailored photon-pair generation in optical fibers. *arXiv.org:0809.0071*, 2008.
- [CN96] I. L. Chuang and M. A. Nielsen. Prescription for experimental determination of the dynamics of a quantum black box. *arXiv.org:/quant-ph/9610001*, 1996.
- [CRS07] H B Coldenstrodt-Ronge and C Silberhorn. Avalanche photo-detection for high data rate applications. *Journal of Physics B: Atomic, Molecular and Optical Physics*, 40(19):3909–3921, 2007.
- [CS96] A. R. Calderbank and P. W. Shor. Good quantum error-correcting codes exist. *Phys. Rev. A*, 54(2):1098–1105, 1996.
- [DAB03] W. Dür, H. Aschauer, and H. J. Briegel. Multiparticle entanglement purification for graph states. *Physical Review Letters*, 91:107903, 2003.
- [DBL⁺03] A. Dolinska, B. C. Buchler, P. K. Lam, T. C. Ralph, and W. P. Bowen. Teleportation of continuous variable polarisation states. *Physical Review A*, 68:052308, 2003.
- [DEJ⁺96] D. Deutsch, A. Ekert, R. Jozsa, C. Macchiavello, S. Popescu, and A. Sanpera. Quantum privacy amplification and the security of quantum cryptography over noisy channels. *Phys. Rev. Lett.*, 77:2818–2821, 1996.
- [DGI⁺07] S. J. Devitt, A. D. Greentree, R. Ionicioiu, J. L. O’Brien, W. J. Munro, and L. C. L. Hollenberg. The photonic module: an on-demand resource for photonic entanglement. *Physical Review A*, 76:052312, 2007.
- [DLP95] G. M. D’Ariano, U. Leonhardt, and H. Paul. Homodyne detection of the density matrix of the radiation field. *Phys. Rev. A*, 52(3):R1801–R1804, Sep 1995.
- [DM98] G. M. D’Ariano and L. Maccone. Measuring quantum optical hamiltonians. *Phys. Rev. Lett.*, 80(25):5465–5468, 1998.
- [DMMS99] Wim van Dam, Frederic Magniez, Michele Mosca, and Miklos Santha. Self-Testing of One-Qubit Gates. 1999.

- [DMP04] G. M. D’Ariano, L. Maccone, and P. L. Presti. Quantum calibration of measurement instrumentation. *Phys. Rev. Lett.*, 93(25):250407, 2004.
- [DW05] I. Devetak and A. Winter. Distillation of secret key and entanglement from quantum states. *PROC.R.SOC.LOND.A*, 461:207, 2005.
- [EBA07] J. Eisert, F. G. S. L. Brandão, and K. M. R. Audenaert. Quantitative entanglement witnesses. *New Journal of Physics*, 9:46, 2007.
- [EBSP04] J. Eisert, D. E. Browne, S. Scheel, and M. B. Plenio. Distillation of continuous-variable entanglement with optical means. *Annals of Physics*, 311:431, 2004.
- [EHKB04] H. S. Eisenberg, J. F. Hodelin, G. Khoury, and D. Bouwmeester. Multiphoton path entanglement by non-local bunching. *Phys. Rev. Lett.*, 94:090502, 2004.
- [Ein53] A. Einstein. Elementare überlegungen zur Interpretation der Grundlagen der Quanten-Mechanik. (German) [Elementary considerations for the interpretation of the foundation of quantum mechanics]. In *Scientific papers presented to Max Born*, pages 33–40. Hafner Publishing Co. Inc., New York, NY, USA, 1953.
- [EJPP00] J. Eisert, K. Jacobs, P. Papadopoulos, and M. B. Plenio. Optimal local implementation of nonlocal quantum gates. *Phys. Rev. A*, 62:052317, 2000.
- [Eke91] A. K. Ekert. Quantum cryptography based on Bell’s theorem. *Phys. Rev. Lett.*, 67:661, 1991.
- [EP03] J. Eisert and M. B. Plenio. Introduction to the basics of entanglement theory in continuous-variable systems. *International Journal of Quantum Information*, 1:479, 2003.
- [EPB⁺07] J. Eisert, M. B. Plenio, D. E. Browne, S. Scheel, and A. Feito. On the experimental feasibility of continuous-variable optical entanglement distillation. *Optics and Spectroscopy*, 103:181, 2007.
- [EPR35] A. Einstein, B. Podolsky, and N. Rosen. Can quantum-mechanical description of physical reality be considered complete? *Phys. Rev.*, 47(10):777–780, 1935.

- [ESP02a] J. Eisert, S. Scheel, and M. B. Plenio. Distilling gaussian states with gaussian operations is impossible. *Phys. Rev. Lett.*, 89(13):137903, 2002.
- [ESP02b] J. Eisert, C. Simon, and M. B. Plenio. On the quantification of entanglement in infinite-dimensional quantum systems. *JOURNAL OF PHYSICS A*, 35:3911, 2002.
- [EVWW01] T. Eggeling, K. G. H. Vollbrecht, R. F. Werner, and M. M. Wolf. Distillability via protocols respecting the positivity of partial transpose. *Phys. Rev. Lett.*, 87(25):257902, 2001.
- [FAW⁺05] J. Fulconis, O. Alibart, W. Wadsworth, P. Russell, and J. Rarity. High brightness single mode source of correlated photon pairs using a photonic crystal fiber. *Opt. Express*, 13(19):7572–7582, 2005.
- [FAWR07] J Fulconis, O Alibart, W J Wadsworth, and J G Rarity. Quantum interference with photon pairs using two micro-structured fibres. *New Journal of Physics*, 9(8):276, 2007.
- [Fcv01] J. Fiurášek. Maximum-likelihood estimation of quantum measurement. *Phys. Rev. A*, 64(2):024102, 2001.
- [Fcv02] J. Fiurášek. Gaussian transformations and distillation of entangled gaussian states. *Phys. Rev. Lett.*, 89(13):137904, Sep 2002.
- [FKLW01] M. H. Freedman, A. Kitaev, M. J. Larsen, and Z. Wang. Topological quantum computation. *arXiv.org:quant-ph/0101025*, 2001.
- [FMF03] J. Fiurášek, L. Mista, and R. Filip. Entanglement concentration of continuous variable quantum states. *Physical Review A*, 67:022304, 2003.
- [FMM⁺03] M. Freitag, Y. Martin, J.A. Misewich, R. Martel, and Ph. Avouris. Photoconductivity of single carbon nanotubes. *Nano Letters*, 3(8):1067–1071, 2003.
- [GDCZ01] G. Giedke, L. M. Duan, J. I. Cirac, and P. Zoller. Distillability criterion for all bipartite gaussian states. *Quant. Inf. Comp.*, 1:79, 2001.

- [GG02] F. Grosshans and P. Grangier. Continuous variable quantum cryptography using coherent states. *Physical Review Letters*, 88:057902, 2002.
- [GHSZ90] D. M. Greenberger, M. A. Horne, A. Shimony, and A. Zeilinger. Bell’s theorem without inequalities. *Am. J. Phys.*, 58:1131, 1990.
- [GIC02] G. Giedke and J. I. Cirac. Characterization of gaussian operations and distillation of gaussian states. *Phys. Rev. A*, 66(3):032316, Sep 2002.
- [Gis96] N. Gisin. Hidden quantum nonlocality revealed by local filters. *Physics Letters A*, 210(3):151–156, 1996.
- [GMN06] B. Ghosh, A. S. Majumdar, and N. Nayak. Environment assisted entanglement enhancement. *Physical Review A*, 74:052315, 2006.
- [GOC⁺01] G. N. Gol’tsman, O. Okunev, G. Chulkova, A. Lipatov, A. Semenov, K. Smirnov, B. Voronov, A. Dzardanov, C. Williams, and Roman Sobolewski. Picosecond superconducting single-photon optical detector. *Applied Physics Letters*, 79(6):705–707, 2001.
- [GRW07] O. Gühne, M. Reimpell, and R. F. Werner. Estimating entanglement measures in experiments. *Physical Review Letters*, 98:110502, 2007.
- [GRW08] O. Gühne, M. Reimpell, and R. F. Werner. Lower bounds on entanglement measures from incomplete information. *Physical Review A*, 77:052317, 2008.
- [GS07] S. Ghose and B. C. Sanders. Non-gaussian ancilla states for continuous variable quantum computation via gaussian maps. *Journal of Modern Optics*, 54:855–869, 2007.
- [GSV04] P. Grangier, B. Sanders, and J. Vuckovic. Focus on single photons on demand. *New Journal of Physics*, 6, 2004.
- [Har93] L. Hardy. Nonlocality for two particles without inequalities for almost all entangled states. *Phys. Rev. Lett.*, 71(11):1665–1668, 1993.
- [HBB⁺07] B. L. Higgins, D. W. Berry, S. D. Bartlett, H. M. Wiseman, and G. J. Pryde. Entanglement-free heisenberg-limited phase estimation. *Nature*, 450:393, 2007.

- [HCL01] P. Horodecki, J. I. Cirac, and M. Lewenstein. Bound entanglement for continuous variables is a rare phenomenon. *arXiv.org:quant-ph/0103076*, 2001.
- [Hel76] C.W. Helstrom. *Quantum detection and estimation theory*. Academic Press, New York, 1976.
- [HFD⁺07] B. Hage, A. Franzen, J. DiGuglielmo, P. Marek, J. Fiurasek, and R. Schnabel. On the distillation and purification of phase-diffused squeezed states. *New Journal of Physics*, 9:227, 2007.
- [HFT⁺05] K. Hirosawa, H. Furumochi, A. Tada, F. Kannari, M. Takeoka, and M. Sasaki. Photon number squeezing of ultra-broadband laser pulses generated by microstructure fibers. *arXiv.org:quant-ph/0501043*, 2005.
- [HH01] P. Horodecki and R. Horodecki. Distillation and bound entanglement. *Quantum Information and Computation*, 1(1):45–75, 2001.
- [HHH96] Michal Horodecki, Paweł Horodecki, and Ryszard Horodecki. Separability of mixed states: Necessary and sufficient conditions. *arXiv.org:quant-ph/9605038*, 1996.
- [HHH97] M. Horodecki, P. Horodecki, and R. Horodecki. Inseparable two spin- $\frac{1}{2}$ density matrices can be distilled to a singlet form. *Phys. Rev. Lett.*, 78:574, 1997.
- [HHHH07] R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki. Quantum entanglement. *arXiv.org:quant-ph/0702225, to appear in Rev. Mod. Phys.*, 2007.
- [HHR⁺05] H. Haeffner, W. Hänsel, C. F. Roos, J. Benhelm, D. Chek al kar, M. Chwalla, T. Koerber, U. D. Rapol, M. Riebe, P. O. Schmidt, C. Becher, O. Gühne, W. Dür, and R. Blatt. Scalable multi-particle entanglement of trapped ions. *Nature*, 438:643, 2005.
- [HK07] Myung-Joong Hwang and Yoon-Ho Kim. Practical scheme for non-postselection entanglement concentration using linear optical elements. *Phys. Lett. A*, 2007.
- [HMR06] Z. Hradil, D. Mogilevtsev, and J. Rehacek. Biased tomography schemes: an objective approach. *arXiv.org:quant-ph/0606042*, 2006.

- [HR07] N. Harrigan and T. Rudolph. Ontological models and the interpretation of contextuality, 2007.
- [HSG⁺05] R. H. Hadfield, M. J. Stevens, S. S. Gruber, A. J. Miller, R. E. Schwall, R. P. Mirin, and S. Woo Nam. Single photon source characterization with a superconducting single photon detector. *Opt. Express*, 13(26):10846–10853, 2005.
- [Ish97] C. J. Isham. Topos theory and consistent histories: The internal logic of the set of all consistent sets. *International Journal of Theoretical Physics*, 36:785, 1997.
- [IVAC04] S. Iblisdir, G. Van Assche, and N. J. Cerf. Security of quantum key distribution with coherent states and homodyne detection. *Phys. Rev. Lett.*, 93(17):170502, 2004.
- [JcvFcvH03] M. Ježek, J. Fiurášek, and Z. Hradil. Quantum inference of states and processes. *Phys. Rev. A*, 68(1):012305, 2003.
- [JK02a] H. Jeong and M. S. Kim. Efficient quantum computation using coherent states. *Phys. Rev. A*, 65(4):042305, Mar 2002.
- [JK02b] H. Jeong and M. S. Kim. Purification of entangled coherent states. *QUANTUM INFORMATION AND COMPUTATION*, 2:208, 2002.
- [KBLSG01] P. G. Kwiat, S. Barraza-Lopez, A. Stefanov, and N. Gisin. Experimental entanglement distillation and ‘hidden’ non-locality. *Nature*, 409:1014–1017, 2001.
- [Ken98a] A. Kent. Entangled mixed states and local purification. *Physical Review Letters*, 81:2839, 1998.
- [Ken98b] A. Kent. Entangled mixed states and local purification. *Physical Review Letters*, 81:2839, 1998.
- [Kit02] A. Yu. Kitaev. Fault-tolerant quantum computation by anyons. *Annals of Physics*, 303:2–30, 2002.
- [KJRPG⁺07] K L Pregnell K J Resch, R Prevedel, A. Gilchrist, G. J. Pryde, J. L. O’Brien, and A. G. White. Time-reversal and super-resolving phase measurement. *Phys. Rev. Lett.*, 98:223601, 2007.

- [Krü01] O. Krüger. *Verschränktheitsmaße Gaußcher Zustände mit 1×1 Moden*. Diplomarbeit Quantum Information, Institut für Matematische Physik, TU Braunschweig., 2001.
- [KTYH99] J. Kim, S. Takeuchi, Y. Yamamoto, and H. H. Hogue. Multiphoton detection using visible light photon counter. *Applied Physics Letters*, 74(7):902–904, 1999.
- [KW04] M. Koashi and A. Winter. Monogamy of entanglement and other correlations. *Physical Review A*, 69:022309, 2004.
- [LB99] S. Lloyd and S. L. Braunstein. Quantum computation over continuous variables. *Physical Review Letters*, 82:1784, 1999.
- [LDM⁺03] D. Leibfried, B. DeMarco, V. Meyer, D. Lucas, M. Barrett, J. Britton, W. M. Itano, B. Jelenković, C. Langer, T. Rosenband, and D. J. Wineland. Experimental demonstration of a robust, high-fidelity geometric two-ion qubit phase gate. *Nature*, 422:412–415, 2003.
- [LE04] C. K. Law and J. H. Eberly. Analysis and interpretation of high transverse entanglement in optical parametric down conversion. *Phys. Rev. Lett.*, 92(12):127903, 2004.
- [Leo97] U. Leonhardt. *Measuring the Quantum State of Light*. Cambridge University Press, 1997.
- [LFCR⁺08] J. S. Lundeen, A. Feito, H. Coldenstrodt-Ronge, K. L. Pregnell, Ch Silberhorn, T. C. Ralph, J. Eisert, M. B. Plenio, and I. A. Walmsley. Measuring measurement. *arXiv.org:0807.2444, to appear in Nature Physics*, 2008.
- [LHA⁺01] A. I. Lvovsky, H. Hansen, T. Aichele, O. Benson, J. Mlynek, and S. Schiller. Quantum state reconstruction of the single-photon fock state. *Phys. Rev. Lett.*, 87(5):050402, 2001.
- [LHSJ⁺08] T. Lund-Hansen, S. Stobbe, B. Julsgaard, H. Thyrrstrup, T. Sunner, M. Kamp, A. Forchel, and P. Lodahl. Experimental realization of highly-efficient broadband coupling of single quantum dots to a photonic crystal waveguide. *Physical Review Letters*, 101:113903, 2008.

- [LK87] R. Loudon and P. L. Knight. Squeezed light. *Journal of Modern Optics*, 34:709, 1987.
- [LKC⁺02] R. Laflamme, E. Knill, D. G. Cory, E. M. Fortunato, T. Havel, C. Miquel, R. Martinez, C. Negrevergne, G. Ortiz, M. A. Pravia, Y. Sharf, S. Sinha, R. Somma, and L. Viola. Introduction to nmr quantum information processing, 2002.
- [LKZ⁺98] R. Laflamme, E. Knill, W. H. Zurek, P. Catasti, and S. V. S. Mariappan. Nmr ghz. *PHIL.TRANS.ROY.SOC.LOND.A*, 356:1941, 1998.
- [LMK⁺96] U. Leonhardt, M. Munroe, T. Kiss, T. Richter, and M. G. Raymer. Sampling of photon statistics and density matrix using homodyne detection. *Opt. Commun.*, 127(1-3):144–160, 1996.
- [LO05] B. Lounis and M. Orrit. Single-photon sources. *Reports on Progress in Physics*, 68(5):1129–1179, 2005.
- [LP01] Hoi-Kwong Lo and S. Popescu. Concentrating entanglement by local actions: Beyond mean values. *Phys. Rev. A*, 63(2):022301, 2001.
- [LPD95] U. Leonhardt, H. Paul, and G. M. D’Ariano. Tomographic reconstruction of the density matrix via pattern functions. *Phys. Rev. A*, 52(6):4899–4907, 1995.
- [LR05] A. I. Lvovsky and M. G. Raymer. Continuous-variable optical quantum state tomography. *Arxiv.org:quant-ph/0511044*, 2005.
- [LSS99] A. Luis and L. L. Sánchez-Soto. Complete characterization of arbitrary quantum measurement processes. *Phys. Rev. Lett.*, 83(18):3573–3576, 1999.
- [LV95] U. Leonhardt and J. A. Vaccaro. Bell Correlations in Phase Space: Application to Quantum Optics. *Journal of Modern Optics*, 42:939–943, 1995.
- [LWE00] C. K. Law, I. A. Walmsley, and J. H. Eberly. Continuous frequency entanglement: Effective finite hilbert space and entropy control. *Phys. Rev. Lett.*, 84(23):5304–5307, 2000.
- [LZG⁺07] Chao-Yang Lu, Xiao-Qi Zhou, O. Gühne, Wei bo Gao, Jin Zhang, Zhen sheng Yuan, Alexander G., Tao Yang, and Jian-Wei Pan. Experimental entanglement of six photons in graph states. *NATURE PHYSICS*, 3:91, 2007.

- [Mat02] R. Matsumoto. Conversion of a general quantum stabilizer code to an entanglement distillation protocol. *MATH.GEN.*, 36:8113, 2002.
- [MK06] David Menzies and Natalia Korolkova. Procrustean entanglement concentration of continuous variable states of light. *PHYS.REV.A*, 74:042315, 2006.
- [MK07] David Menzies and Natalia Korolkova. Weak values and continuous-variable entanglement concentration. *arXiv.org:0707.3917*, 2007.
- [MLS04] M.W. Mitchell, J.S. Lundeen, and A.M. Steinberg. Super-resolving phase measurements with a multiphoton entangled state. *Nature*, 429:161–164, 2004.
- [MLSW08] P J Mosley, J S Lundeen, B J Smith, and I A Walmsley. Conditional preparation of single photons using parametric downconversion: a recipe for purity. *New Journal of Physics*, 10(9):093011 (29pp), 2008.
- [MNMS03] A. J. Miller, S. Woo Nam, J. M. Martinis, and A. V. Sergienko. Demonstration of a low-noise near-infrared photon counter with multiphoton discrimination. *Applied Physics Letters*, 83(4):791–793, 2003.
- [Mos07] P. J. Mosley. *Generation of Heralded Single Photons in Pure Quantum States*. PhD thesis, 2007.
- [MPP⁺98] M. Murao, M. B. Plenio, S. Popescu, V. Vedral, and P. L. Knight. Multi-particle entanglement purification protocols. *Physical Review A*, 57:4075, 1998.
- [NBC⁺04] M. Navascues, J. Bae, J. I. Cirac, M. Lewenstein, A. Sanpera, and A. Acin. Key distillation from gaussian states by gaussian operations. *arXiv.org:quant-ph/0405047*, 2004.
- [NC00] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, UK, Cambridge, UK, 2000.
- [Nie03] M.A. Nielsen. Quantum computation by measurement and quantum memory. *Physics Letters A.*, 308:96, 2003.
- [NMR⁺08] P. Neumann, N. Mizuuchi, F. Rempp, P. Hemmer, H. Watanabe, S. Yamasaki, V. Jacques, T. Gaebel, F. Jelezko, and J. Wrachtrup. Multipartite entanglement among single spins in diamond. *Science*, 320(5881):1326–1329, 2008.

- [NNNH⁺06] J. S. Neergaard-Nielsen, B. Melholt Nielsen, C. Hettich, K. Moelmer, and E. S. Polzik. Generation of a superposition of odd photon number states for quantum information networks. *Physical Review Letters*, 97:083604, 2006.
- [ODTBG07] A. Ourjoumtsev, A. Dantan, R. Tualle-Brouri, and P. Grangier. Increasing entanglement between gaussian states by coherent photon subtraction. *Physical Review Letters*, 98:030502, 2007.
- [OKW00] T. Opatrny, G. Kurizki, and D. G. Welsch. Continuous-variable teleportation improvement by photon subtraction via conditional measurement. *Physical Review A*, 61:032302, 2000.
- [OP05a] S. Olivares and M. G A Paris. Photon subtracted states and enhancement of nonlocality in the presence of noise. *Journal of Optics B: Quantum and Semiclassical Optics*, 7(10):S392–S397, 2005.
- [OP05b] S. Olivares and M. G. A. Paris. Squeezed fock state by inconclusive photon subtraction. *SEMICLASS.OPT.*, 7:S616, 2005.
- [OTBG06] A. Ourjoumtsev, R. Tualle-Brouri, and P. Grangier. Quantum homodyne tomography of a two-photon fock state. *Physical Review Letters*, 96:213601, 2006.
- [OTBLG06] A. Ourjoumtsev, R. Tualle-Brouri, J. Laurat, and P. Grangier. Generating optical Schrödinger kittens for quantum information processing. *Science*, 312(5770):83–86, 2006.
- [PBP00] S. Parker, S. Bose, and M. B. Plenio. Entanglement quantification and purification in continuous-variable systems. *Phys. Rev. A*, 61(3):032305, 2000.
- [PCZ97] J. F. Poyatos, J. I. Cirac, and P. Zoller. Complete characterization of a quantum process: The two-bit quantum gate. *Phys. Rev. Lett.*, 78(2):390–393, 1997.
- [PK05] J. Perina and J. Krepelka. Multimode description of spontaneous parametric down-conversion. *Journal of Optics B: Quantum and Semiclassical Optics*, 7:246–252(7), 2005.
- [Ple05] M. B. Plenio. The logarithmic negativity: A full entanglement monotone that is not convex. *Physical Review Letters*, 95:090503, 2005.

- [Ple07] M. B. Plenio. private communication, 2007.
- [Pop95] S. Popescu. Bell’s inequalities and density matrices: Revealing ‘hidden’ nonlocality. *Phys. Rev. Lett.*, 74(14):2619–2622, 1995.
- [PP02] K. L. Pregnell and D. T. Pegg. Measuring the elements of the optical density matrix. *Physical Review A*, 66:013810, 2002.
- [Pre98] J. Preskill. Lecture notes for Physics 229: Quantum information and computation. *Caltech Lecture Notes*, 1998. <http://www.theory.caltech.edu/people/preskill/ph229/>.
- [Pre05] K. L. Pregnell. *Retrodictive quantum state engineering*. PhD in physics, Center for Quantum Computer Technology – Griffith University, 2005.
- [PV07] M. B. Plenio and S. Virmani. An introduction to entanglement measures. *QUANT. INF. COMP.*, 7:1, 2007.
- [qGlFzX03] Shang qing Gong, Xun li Feng, and Zhi zhan Xu. Generation of fock state and quantum entanglement in a coupled ladder atom–cavity system. *Journal of Optics B: Quantum and Semiclassical Optics*, 5(1):52–55, 2003.
- [Rai99] E. M. Rains. Rigorous treatment of distillable entanglement. *Phys. Rev. A*, 60(1):173–178, 1999.
- [Ral03] T. C. Ralph. Quantum key distribution with continuous variables in optics. *arXiv.org:quant-ph/0109096*, 2003.
- [RB02] R. Raussendorf and H. Briegel. Computational model underlying the one-way quantum computer. *QUANT. INF. COMP.*, 6:433, 2002.
- [RHKL07] J. Rehacek, Z. Hradil, E. Knill, and A. I. Lvovsky. Diluted maximum-likelihood algorithm for quantum tomography. *Physical Review A*, 75:042108, 2007.
- [ROW⁺07] K. J. Resch, J. L. O’Brien, T. J. Weinhold, K. Sanaka, and A. G. White. Entanglement generation by fock-state filtration. *Physical Review Letters*, 98:203602, 2007.

- [RRM06] P. P. Rohde, T. C. Ralph, and W. J. Munro. Practical limitations in optical entanglement purification. *Physical Review A*, 73:030301, 2006.
- [SA88] Y. H. Shih and C. O. Alley. New type of einstein-podolsky-rosen-bohm experiment using pairs of light quanta produced by optical parametric down conversion. *Phys. Rev. Lett.*, 61(26):2921–2924, 1988.
- [SBd02] B. C. Sanders, S. D. Bartlett, and H. de Guise. From qubits to continuous-variable quantum computation. *arXiv.org:quant-ph/0208008*, 2002.
- [SBRF93] D. T. Smithey, M. Beck, M. G. Raymer, and A. Faridani. Measurement of the wigner distribution and the density matrix of a light mode using optical homodyne tomography: Application to squeezed states and the vacuum. *Phys. Rev. Lett.*, 70(9):1244–1247, 1993.
- [Sch35] E. Schrödinger. Discussion of probability relations between separated systems. *Proc. Cambridge Philos. Soc.*, 31:555, 1935.
- [Sch01] W. P. Schleich. *Quantum Optics in Phase Space*. Wiley-VCH, Berlin,, 1st edition, 2001.
- [Shi07] A. J. Shields. Review: Semiconductor quantum light sources. *NATURE PHOTONICS*, 1:215, 2007.
- [Sho96] P. W. Shor. Fault-tolerant quantum computation. *ArXiv.org*, 1996.
- [Sho97] P. W. Shor. Polynomial time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Sci. Statist. Comput.*, 26:1484, 1997.
- [SOF⁺00] A. J. Shields, M. P. O’Sullivan, I. Farrer, D. A. Ritchie, R. A. Hogg, M. L. Leadbeater, C. E. Norman, and M. Pepper. Detection of single photons using a field-effect transistor gated by a layer of quantum dots. *Applied Physics Letters*, 76(25):3673–3675, 2000.
- [SOG06] F. W. Sun, Z. Y. Ou, and G. C. Guo. Projection measurement of the maximally entangled n-photon state for a demonstration of n-photon de broglie wavelength. *Phys. Rev. A*, 73:023808, 2006.

- [SPID05] A. Serafini, M. G. A. Paris, F. Illuminati, and S. De Siena. Quantifying decoherence in continuous variable systems. *QUANTUM SEMICLASS.OPT.*, 7:R19, 2005.
- [SSM87] R. Simon, E. C. G. Sudarshan, and N. Mukunda. Gaussian-wigner distributions in quantum mechanics and optics. *Phys. Rev. A*, 36(8):3868–3880, 1987.
- [Ste96] A. M. Steane. Error correcting codes in quantum theory. *Phys. Rev. Lett.*, 77(5):793–797, 1996.
- [Ste98] A. Steane. Quantum computing. *Rep. Prog. Phys.*, 61:117–173, 1998.
- [SV95] O. Steuernagel and J. A. Vaccaro. Reconstructing the density operator via simple projectors. *Phys. Rev. Lett.*, 75(18):3201–3205, 1995.
- [SWM⁺04] M. Sasaki, K. Wakui, J. Mizuno, M. Fujiwara, and M. Akiba. Epr beams and photon number detector: Toward synthesizing optical nonlinearity. *arXiv.org:quant-ph/0601058*, page 44, 2004.
- [Ter00] B. M. Terhal. Bell inequalities and the separability criterion. *Physics Letters A*, 271:319, 2000.
- [TYYF07] Y. Takeno, M. Yukawa, H. Yonezawa, and A. Furusawa. Observation of -9 db quadrature squeezing with improvement of phase stability in homodyne measurement. *Optics Express*, 15:4321, 2007.
- [U.L03] U.Leonhardt. Quantum physics of simple optical instruments. *REPT.PROG.PHYS.*, 66:1207, 2003.
- [Vai94] L. Vaidman. Teleportation of quantum states. *Phys. Rev. A*, 49(2):1473–1476, 1994.
- [vB00] P. van Loock and S. L. Braunstein. Multipartite entanglement for continuous variables: A quantum teleportation network. *Physical Review Letters*, 84:3482, 2000.
- [VP98] V. Vedral and M.B. Plenio. Entanglement measures and purification procedures. *Phys. Rev. A*, 57:1619, 1998.

- [VR89] K. Vogel and H. Risken. Determination of quasiprobability distributions in terms of probability distributions for the rotated quadrature phase. *Phys. Rev. A*, 40(5):2847–2849, Sep 1989.
- [Wat04] J. Watrous. Many copies may be required for entanglement distillation. *Phys. Rev. Lett.*, 93(1):010502, 2004.
- [Wer89] R. F. Werner. Quantum states with Einstein-Podolsky-Rosen correlations admitting a hidden-variable model. *Phys. Rev. A*, 40:4277, 1989.
- [WGK⁺04] M. M. Wolf, G. Giedke, O. Krüger, R. F. Werner, and J. I. Cirac. Gaussian entanglement of formation. *Phys. Rev. A*, 69(5):052320, 2004.
- [Wig32] E. Wigner. *Phys. Rev.*, 40:749, 1932.
- [WKU⁺05] M. B. Ward, O. Z. Karimov, D. C. Unitt, Z. L. Yuan, P. See, D. G. Gevaux, A. J. Shields, P. Atkinson, and D. A. Ritchie. On-demand single-photon source for 1.3 mu m telecom fiber. *Applied Physics Letters*, 86(20):201111, 2005.
- [WPA⁺03] P. Walther, Jian-Wei Pan, M. Aspelmeyer, R. Ursin, S. Gasparoni, and A. Zeilinger. De broglie wavelength of a nonlocal four-photon. *Nature*, 429:158–161, 2003.
- [WRFB08] W. Wasilewski, C. Radzewicz, R. Frankowski, and K. Banaszek. Statistics of multiphoton events in spontaneous parametric down-conversion. *arXiv.org:0805.1701*, 2008.
- [WTFS07] K. Wakui, H. Takahashi, A. Furusawa, and M. Sasaki. Photon subtracted squeezed states generated with periodically poled ktpo4. *Opt. Express*, 15(6):3568–3574, 2007.
- [Wün97] A. Wünsche. Radon transform and pattern functions in quantum tomography. *arXiv.org:quant-ph/9706024*, 1997.
- [WXK87] L. Wu, M. Xiao, and H.J. Kimble. Squeezed states of light from an optical parametric oscillator. *J. Opt. Soc. Am. B*, 4(10):1465–1475, 1987.

- [XbH03] Wang Xiang-bin and Fan Heng. Entanglement concentration by ordinary linear optical devices without postselection. *Phys. Rev. A*, 68(6):060302, 2003.
- [YKOI03] T. Yamamoto, M. Koashi, S. K. Ozdemir, and N. Imoto. Experimental extraction of an entangled photon pair from two identically decohered pairs. *Nature*, 421(6921):343–346, 2003.
- [ZHWidZ97] A. Zeilinger, M. A. Horne, H. Weinfurter, and M. Żukowski. Three-particle entanglements from two entangled pairs. *Phys. Rev. Lett.*, 78(16):3031–3034, 1997.
- [ZYC⁺03] Zhi Zhao, Tao Yang, Yu-Ao Chen, An-Ning Zhang, and Jian-Wei Pan. Experimental realization of entanglement concentration and a quantum repeater. *Phys. Rev. Lett.*, 90(20):207901, 2003.