Compressive Sensing and Quantum State Tomography with Simple Variable Attenuator and Click Detector Model

James Dixon
Supervisors: Prof. J. Jeffers, Dr D. Oi, R. Pausa
University of Strathclyde, Physics Department

June - August 2021

Abstract

The field of Compressive Sensing (CS) in signal processing offers efficient signal determination methods by solving underdetermined linear systems and is already being used for Quantum State Tomography (QST). The work presented here models a simple threshhold detector with variable attentuation to simulate detection rate measurements for a range of detection efficiencies. Inspired by the literature on CS for QST, a convex optimisation procedure is developed to test whether reducing the mutual-coherence of the so-called measurement matrix of an underdetermined system can improve the efficacy of the optimisation procedure. Following this, the software demonstrates a capacity to evaluate combinations of detection efficiencies with which to measure and determine some unknown fock distribution.

1 Introduction

The aim of this work is to computationally investigate matrix mutual coherence and optimisation techniques inspired by Compressive Sensing (CS) from signal processing for quantum state tomography (QST). In particular, the single photon probability is sought in accordance with the need for knowledge of multiphoton state probability to evaluate security of protocols in quantum cryptography [1]. In addition, the software developed is demonstrated to facilitate the choice of detection efficiencies when experimentally identifying fock distributions.

Specifically, this work involves a computational model of a click detector (e.g. a single photon avalanche detector) with variable attenuation responding to a source of photons, of unknown fock state distribution, providing measurements from which the quantum state of the light source is estimated via convex optimisation procedures.

2 Background and Theory

2.1 Photon Counting

Assume a single mode source diagonal in the fock basis,

$$\rho = \sum_{n=0}^{\infty} p(n) \left|n\right\rangle \left\langle n\right|, \tag{1}$$

that emits n photons with probability $p(n)$. We have a setup that registers photodetection with conditional probability $P(\checkmark|\eta, n)$, which will account for dark count d_A and total system detection efficiency (including filter, channel loss and detector) η.

$$P(\checkmark|\eta, n) = \eta_n + (1-\eta_n)d_A = 1 - (1-d_A)(1-\eta)^n, \tag{2}$$

where $\eta_n = 1-(1-\eta)^n$ - the click probability for n photons arriving at the detector. The probability of the detector firing $P(\checkmark)$, measured as the average count rate, is

$$P(\checkmark) = \sum_{n=0}^{\infty} P(\checkmark|\eta, n)p(n). \tag{3}$$

Repeated measurements of the average count rate, for varying η, may be represented by, vector of average count rates,

$$\boldsymbol{y} = \boldsymbol{M}\boldsymbol{x}. \tag{4}$$

Where $\boldsymbol{M}$ is the measurement matrix with elements given by Eq. (2), $m_{i,j} = P(\checkmark|\eta_i, n_j)$ for $\eta_i \in \{\eta_1, ...\eta_k\}$ (for k detection efficiencies) and $n_j \in \{0, 1, ..., n_{max}\}$. The signal vector $\boldsymbol{x}$ is the diagonal elements of ρ, up to n_{max}. For computational practicalities, the number of photons per pulse is assumed to be limited to n_{max}. With $k < n_{max}$ this is an underdetermined system. It is the task of this work to solve the system for $\boldsymbol{x}$, thus, finding the probabilities $p(n)$.

2.2 Compressive Sensing

2.2.1 Essential Idea

In signal processing, Compressive Sensing is a technique in which a signal is compressed as it is detected and allows a sampling rate, much lower than the Nyquist rate, from which the original signal can be recovered [2]. The essential ideas of CS necessary for this work follow.

A signal vector $\boldsymbol{r}$, in a sparsifying orthonormal basis ψ, is represented

$$\boldsymbol{r} = \psi \boldsymbol{s} \tag{5}$$

and a compressed measurement, $\boldsymbol{y}$, of $\boldsymbol{r}$

$$\boldsymbol{y} = \varphi \boldsymbol{r}. \tag{6}$$

Here, φ is an $(m \times n)$ measurement matrix that randomly selects from $\boldsymbol{r}$ yielding a measurement $\boldsymbol{y}$ of length $m << n$. The compressed measurement of a sparsified signal $\boldsymbol{s}$, of length n, can be represented by the underdetermined system

$$\boldsymbol{y} = \Theta \boldsymbol{s}. \tag{7}$$

Here, $\Theta := \varphi\psi$ is an $(m \times n)$ reconstruction matrix and $\boldsymbol{y}$ a measurement vector of m measurements. A computationally feasible way to solve this system is by convex optimisation and finding an estimate of $\boldsymbol{s}$, denoted $\tilde{\boldsymbol{s}}$, that minimises the $l_1 - norm$,

$$\tilde{\boldsymbol{s}} = min(||\boldsymbol{s}||_1), \tag{8}$$

such that $\Theta \boldsymbol{s} = \boldsymbol{y}$. In CS, the non-sparse signal is then found by reverse transforming the sparsifying basis. To read further on CS see Ref. [2].

2.2.2 Coherence

It has been shown that the non-sparse signal is perfectly recovered if and only if $\boldsymbol{s}$ is sufficiently sparse and a property called the restricted isometry property (RIP) is satisfied by Θ [3]. However, forming matrices that satisfy the RIP is NP-hard. Reliable reconstruction of $\tilde{\boldsymbol{s}}$ is possible if Θ is incoherent, that is, its mutual coherence $\mu(\Theta)$ is sufficiently small [2, 4, 5]. The mutual coherence of a matrix with columns normalised to be unit vectors is found as the largest non diagonal element of the Gram matrix $G = \Theta^T\Theta$.

$$\mu(\Theta) = \underbrace{max}_{i \neq j} |G_{i,j}| \tag{9}$$

The lower $\mu(\Theta)$, the less coherent and less correlation between columns of Θ meaning fewer measurements required for reliable reconstruction.

2.2.3 Compressive Sensing for Quantum State Tomography

In the interest of placing this work in context, CS for QST was pioneered in works including [6, 7, 8] and these papers were particularly important in developing the work presented here.

Comparing Eq.s (4) and (7), in both instances we have underdetermined systems with known measurement matrix and measurement vectors. Given these, the literature on CS provides many strategies for estimating the original signal. A common approach [2, 6, 7, 8, 9] for QST is convex optimisation by basis pursuit. This ensures that the Fock distribution is normalised, $tr(\rho) = 1$ and all diagonal elements of ρ are non-negative. The exact methods used in this work are set out in Sec. 4

3 Project Aims

In this work, a computational model of a photon counting experiment and a protocol for estimating the Fock distribution of the source of photons is developed. To this end, a convex optimisation procedure is implemented inspired by the techniques in CS. In addition, different combinations of detection efficiencies are used and the relationship between the coherence of the corresponding measurement matrix and the reliability of the signal estimate, in particular $P_{0,1}$, is investigated. The hypothesis that reducing the coherence of the measurement matrix will facilitate the optimisation objective is tested. The software was developed in Python and will be available at `https://github.com/jmsdixon/Photon-Stats-ConVxOpt`.

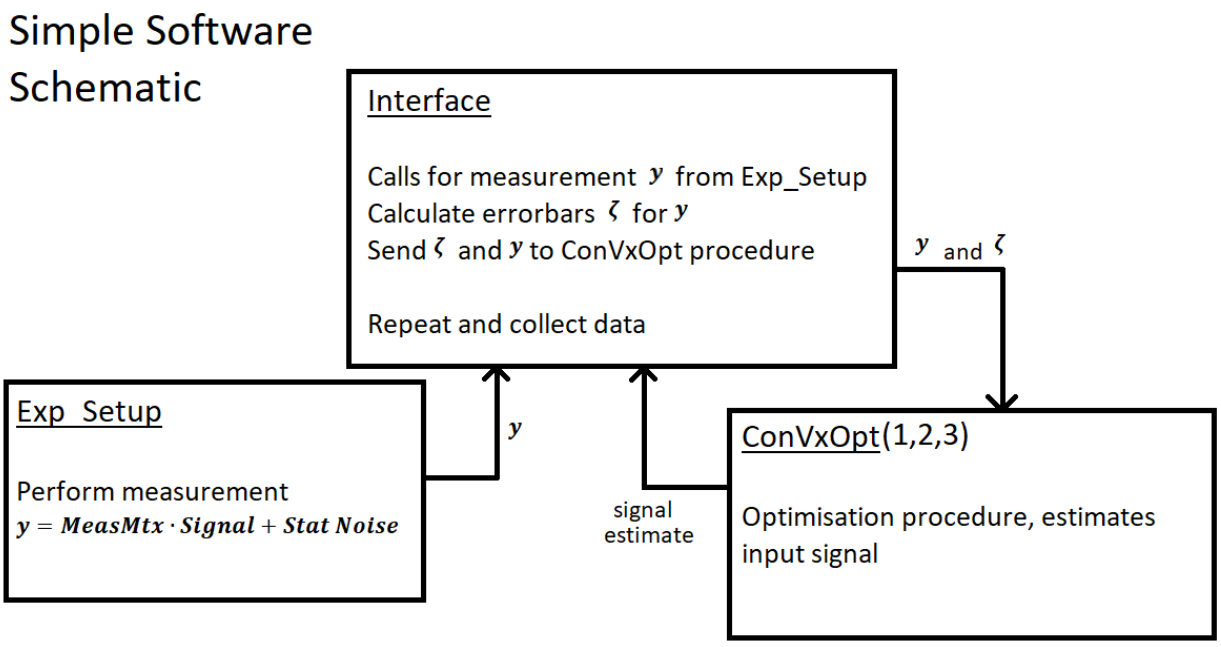


Figure 1: Simple schematic of the software.

4 Measurement and Optimisation Model

A simple schematic of the composition of the software is seen in Fig. 1. The interface allows particular specifications of parameters and data collection setup. The specifications include the signals to be measured and combinations of efficiencies with which to measure. It calls for simulated measurements from the experimental setup given the specifications and then sends the measurement with 'estimated-errorbars' to the convex optimisation procedure. With the signal estimates, analysis is made and the data and analysis are written to txt files. The input signal is displayed in Fig. 3a.

Figure 2 displays the assumed experimental setup for the detection simulation model. As seen, a photon source is detected by a simple click detector via a variable attenuator permitting the variation of η. The model was developed to simulate a photon counting experiment, which measures the count rate $P_i(\checkmark) = y_i$ of the setup detector in response to the attenuated photon source signal for each detection efficiency η_i of a combination of efficiencies comprising k different efficiencies. Performing experimental determinations of $P_i(\checkmark)$ requires computing the mean count rate and features associated statistical noise. We therefor add the noise term $\boldsymbol{\epsilon}$ to Eq.(4),

$$\boldsymbol{y} = \boldsymbol{M}\boldsymbol{x} + \boldsymbol{\epsilon}. \tag{10}$$

The noise term is a vector of randomly generated elements ϵ_i proportional to y_i and normally distributed about zero with a standard deviation σ. For the methodology presented here, one measurement will refer to once taking the product of the measurement matrix and the signal and adding noise, Eq. 10.

A separate program was written to form combinations of η and calculate the coherence of their respective measurement matrices. Coherence was calculated according to Eq. (9).

With the simulated measurement vector with added statistical noise, $\boldsymbol{y}$, provided by the experimental setup, the interface computes experimental data errorbars, forming ζ_i for each y_i. This is done similarly to the statistical noise vector $\boldsymbol{\epsilon}$ and the errorbars represent the standard error of the measurement mean values. Supplying $\boldsymbol{y}, \boldsymbol{M}$ and $\boldsymbol{\zeta}$, an estimation of the signal $\tilde{\boldsymbol{x}}$ is computed by one of the 3 convex optimisation methods below. The procedures make use of Python's scipy.optimize package and the Sequential Least

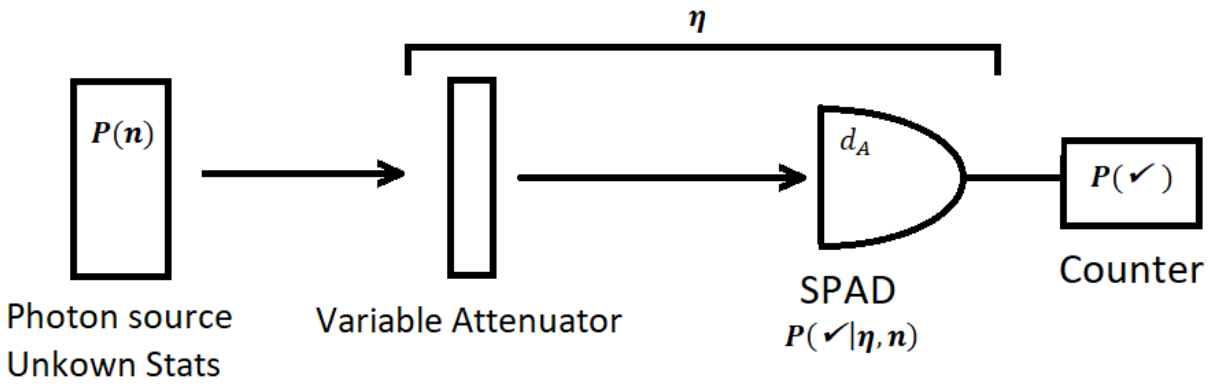


Figure 2: Simple schematic of source of photons of unknown statistics detected by a single photon avalanche photodiode, allowing for a variable detection efficiency η with detector dark count d_A.

Squares Programming Algorithm. The first method, we'll call ConVxOpt1, is inspired by the methods in Ref.s [4, 7].

$$min||P_{0,1}||_1 \tag{11}$$
$$s.t.\ ||\boldsymbol{y} - \boldsymbol{M}\tilde{\boldsymbol{x}}||_2 \leq min(\boldsymbol{\zeta}) \tag{12}$$
$$and\ \sum_{i=1}^{n} \tilde{x}_i = 1 \tag{13}$$

The second ConVxOpt2, inspired by Ref. [9].

$$min||P_{0,1}||_1 \tag{14}$$
$$s.t.\ abs(y_i - \boldsymbol{M}_{i,:}\tilde{\boldsymbol{x}}) \leq \zeta_i \tag{15}$$
$$and\ \sum_{i=1}^{n} \tilde{x}_i = 1 \tag{16}$$

Here, $\boldsymbol{M}_{i,:}$ refers to the i^{th} row of the measurement matrix.

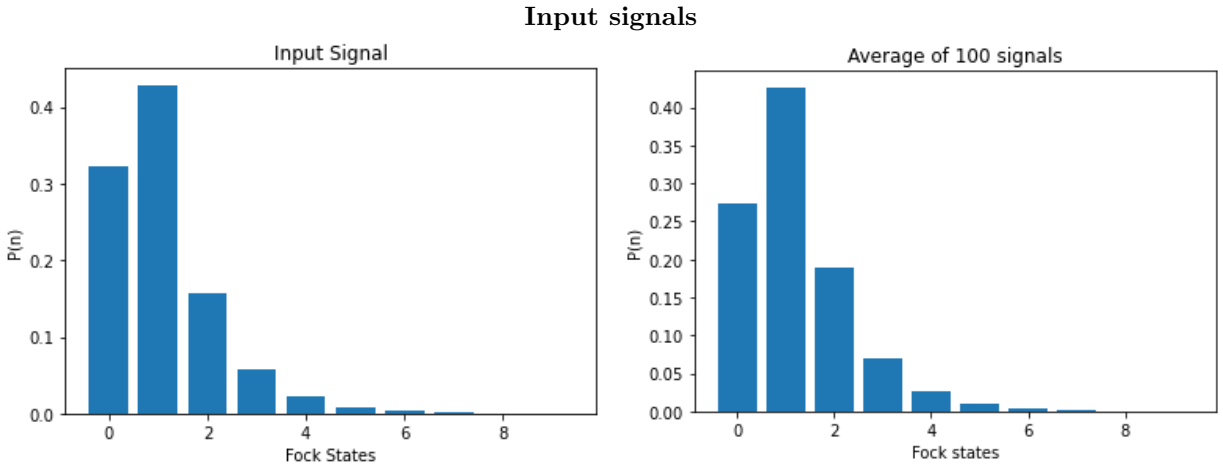
To contrast ConVxOpt2 - minimising the single photon probability - a third method was developed, ConVxOp3. Here, the multi-photon probability P_m is minimised.

$$min||P_m||_1 \tag{17}$$
$$s.t.\ abs(y_i - \boldsymbol{M}_{i,:}\tilde{\boldsymbol{x}}) \leq \zeta_i \tag{18}$$
$$and\ \sum_{i=1}^{n} \tilde{x}_i = 1 \tag{19}$$

This will help to indicate if low values for coherence are facilitating the optimisation objective.

5 Methodology

Unless otherwise stated, the standard deviation for the noise term σ and the errorbars is 10^{-4} and the dark count rate is set to 10^{-7}. Also, each (6×10) measurement matrix was comprised for 6 detection efficiencies and 10 fock states.



(a) Input Signal

(b) Average of 100 Signals

Figure 3: (a) the single input signal used. (b) in the cases that 100 signals are used, they are random variations of such a distribution.

Initially, ConVxOpt1 and ConVxOpt2 are contrasted for performance given the input signal seen in Fig. 3a and the combination of 6 evenly spaced detection efficiencies in the range from 0.133 to 0.8 (see App. A). 20000 measurements and optimisation estimations (where one measurement/optimisation refers to computing Eq. (10) and applying a ConVxOpt method once) are made for each method. The process of a single measurement and estimation will simply be called an estimation from now on. Given the better performance of ConVxOpt2 in underestimating $P_{0,1}$, and a shorter computation time, this method is used for the remainder of the investigation unless otherwise stated.

Next, combinations of 6 different detection efficiencies are formed to have relatively small or large coherences; it proved that producing measurement matrices with coherence less than 0.999 was very difficult. To do this, sets of 15[1] possible detection efficiencies were formed, see App. A for definitions. All combinations of 6 detection efficiencies without repeats were computed for each set and their coherence calculated. From each of these sets, 500 were chosen at random to form a group and the 50 with the lowest coherence values another group. For the groups of 50 combinations, each combination estimated a single input signal 100 times and the relationship between the coherence and relative error of P_{01} was investigated. The relative error here referring to the relative difference between the signal and the estimated signal. The same was done for the groups of 500 but with only 50 estimations to reduce computation time.

Next, both ConVxOpt2 and ConVxOpt3 were used to estimate the 100 signals with 500 random combinations from Set 10, 5 times each. 5 estimations of each of 100 signals was sufficient to get an overall picture of each combination's behaviour within a practical time frame - approximately 12 hours. With a opposing optimisiation objective, the results for ConVxOpt3 provide a contrast to the hypothesis-confirming results of ConVxOpt2.

Finally, to find experimentally implementable combinations of detection efficiencies that facilitate a precise and accurate estimation of P_{01}, the following method was applied. A set of detection efficiencies, Set 14 (see App. A) was formed assuming the use of 50% neutral density filters. From Set 14, 500 random combinations of 6 were chosen, again, with no repeated efficiencies. Every combination includes the largest detection efficiency to ensure a broad range (distinguishing Set 14 from Set 2). For each detection efficiency combination and each of the 100 signals, 5 estimations were made. Inspecting the results, the combinations that provided the largest multiphoton probability errors were seen to have the most precise and accurate P_{01} values, so were extracted. These combinations were then run for 50 estimations of 100 signals each and

[1]With the exception of Set 7 - see App. A

the 'best' performing combination was chosen. Here 'best' means a good accuracy at estimating P_{01} but with slight underestimation bias, to 'err on the side of caution' baring in mind the security requirements in quantum cryptography. This 'best' combination is then used to estimate the set of 100 signals 200 times each and then to estimate the single input signal 20000 times.

5.1 Results

5.1.1 Comparison of ConVxOpt1 and ConVxOpt2

Figure 4 shows histograms of the number of estimations of $P_{0,1}$ at different values for (a) ConVxOpt1 and (b) ConVxOpt2, 20000 estimations were taken for each. The actual signal value for $P_{0,1} \approx 0.750$, displayed by the red dashed line in Fig. 4. The average $P_{0,1}$ estimations by ConVxOpt1 and ConVxOpt2 were $P_{0,1} \approx 0.749$ and $P_{0,1} \approx 0.741$ respectively and the fractions of $P_{0,1}$ over-estimation were 41.99% and 5.81%. Given our desire for underestimation, ConVxOpt2 is clearly superior, whilst ConVxOpt1 is arguably better given a desire for an accurate $P_{0,1}$ estimation. Another important factor in using ConVxOpt2 is it provides a significant improvement in computation time. ConVxOpt1 took approximately 5 hours for 20000 measurements and optimisations, whereas ConVxOpt2 only around 40 minutes. It's interesting to note that the ConVxOpt1 histogram shape appears somewhat like a Normal distribution about the actual signal value of $P_{0,1}$, but with a protrusion - perhaps due to the optimisation objective minimising $P_{0,1}$. It is, however, possible that this protruding shape diminishes for greater numbers of estimations.

5.1.2 Low coherence and estimated $P_{0,1}$ accuracy

In Fig. 5, we see the average relative error (between the input and estimated signals such that negativity indicates underestimation) in the $P_{0,1}$ estimates for the random samples of 500 from the 13 sets of detection efficiencies in Fig. 5a and the 50 with the minimum values of coherence from each set in Fig. 5b, plotted against coherence. The random 500 show no obvious relation between lower coherence and average relative error in $P_{0,1}$, though the 2 sets with the largest coherence have relative errors of small magnitude. The 50 minimum coherence combinations indicate such a relation is possible, with a much lower relative error associated with the sets of lower coherence combinations. As the ConVxOpt2 optimisation objective is to minimise $P_{0,1}$, this could be a due to lower coherence facilitating the optimisation, which would be expected to provide a lower $P_{0,1}$.

5.1.3 Low coherence and 100 signals

With 500 random combinations from Set 10 and 100 signals, the average $P_{0,1}$ relative error plotted against coherence is displayed below for ConVsOpt2 and ConVxOpt3 in Fig. 6. We see no obvious and significant difference in the estimation results as we might expect given that the 2 methods are minimising alternative elements of the estimated signals. For ConVxOpt2, we see the reduction in $P_{0,1}$ as coherence decreases, as expected if lower coherence enhances the optimisation. However, we see the same behaviour for ConVxOpt3, which is designed to minimise the multiphoton probability, P_m. This suggests that some other factor is causing a lower $P_{0,1}$ estimate as coherence decreases. More precisely, for ConVxOpt2, the overall average relative error of $P_{0,1}$ for the 500 combinations is -0.0845 and for ConVxOpt3 is -0.0841. The difference is less than a single percentage and lower for ConVxOpt2. From this data, we can say that minimising mutual coherence in the measurement matrix does not appear to be facilitating the optimisation objective.

5.1.4 Finding an optimal and practically implementable combination of detection efficiencies

From Set 14, the average $P_{0,1}$ relative error, between the signal estimates and the input signal, is plotted against the fraction of the total error in the signal in P_m for 500 random combinations each estimating 100 signals in Fig. 7a. The standard deviation of the average $P_{0,1}$ relative errors, of all measurements of all signals, is in Fig. 7b. We see that the average $P_{0,1}$ relative errors appear to converge to zero as the fraction of the error in the signal in P_m increases (as expected) and from Fig. 7b, that the precision similarly

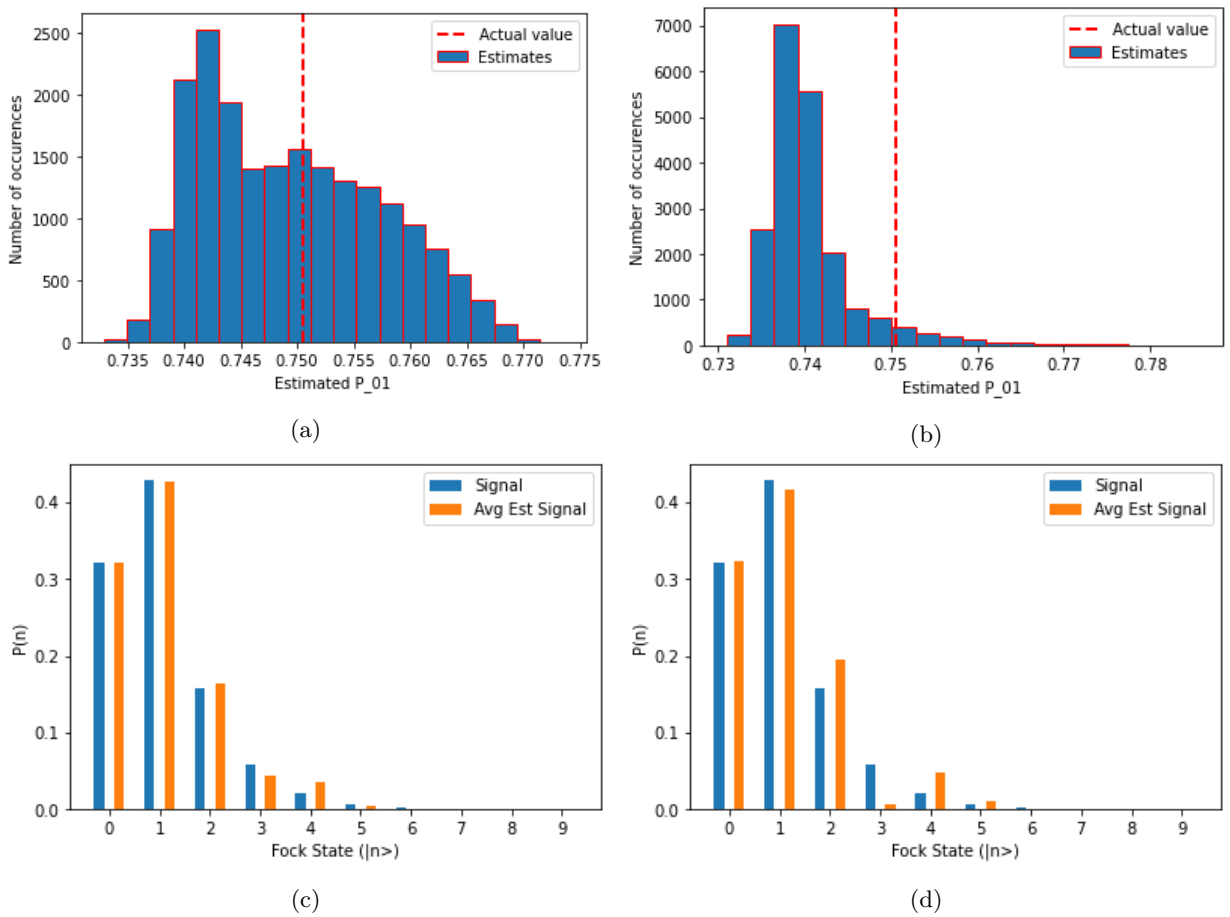


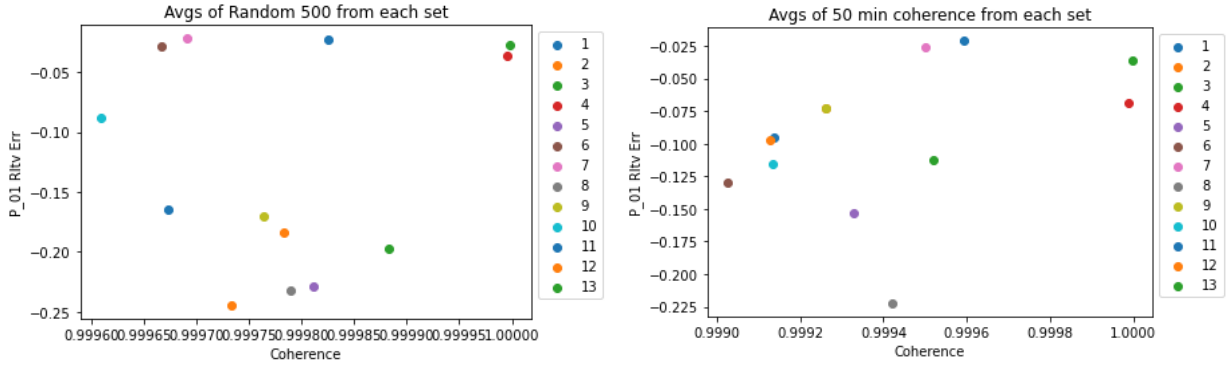
Figure 4: Histograms of estimated $P_{0,1}$ for 20000 measurements for (a) ConVxOpt1 and (b) ConVxOpt2. The rate of underestimation of the actual value is clearly greater for ConVxOpt2. (c) shows the average signal estimate and the input signal for ConVxOpt1 and (d) shows the same for ConVxOpt2. In both cases the error between the estimated and input signal $P_{0,1}$ is very small.

increases. Figure 7c shows the 14 combinations with the largest fraction of total signal error in P_m, all but one have average $P_{0,1}$ relative error less than zero. The combination, indicated in the red circle, with the lowest average $P_{0,1}$ relative error was chosen in alignment with our desire to error as underestimation of $P_{0,1}$. In addition, we note that this combination also has one of the highest fraction of error in P_m, affording us confidence in its accuracy. This combination of efficiencies is {8.00e-01, 4.00e-01, 2.00e-01, 1.00e-01, 5.00e-02, 9.77e-05}.

With this combination of detection efficiencies, the results of 200 estimations of each of 100 signals are presented. Figures 8a and 8b show the average $P_{0,1}$ relative error and the average $P_{0,1}$ estimate for each signal. We can see that all the $P_{0,1}$ relative errors are negative, indicating underestimation as preferred, and the high degree of precision in the $P_{0,1}$ relative errors and estimates - indicated by the very narrow errorbars. The errorbars in these figures are the standard errors of the means of the 200 estimations.

Figure 9 displays (a) the average $P_{0,1}$ relative error against the signal $P_{0,1}$, (b) the average $P_{0,1}$ absolute error against the signal $P_{0,1}$ and (c) the average $P_{0,1}$ relative error against the average $P_{0,1}$ absolute error. From Fig. 9a, the relative error in $P_{0,1}$ estimation is seen to become greater in magnitude as the signal P_{01} decreases, as does the absolute error in $P_{0,1}$ (Fig. 9b). A corresponding decrease in precision is apparent

Average $P_{0,1}$ relative errors for each set vs coherence

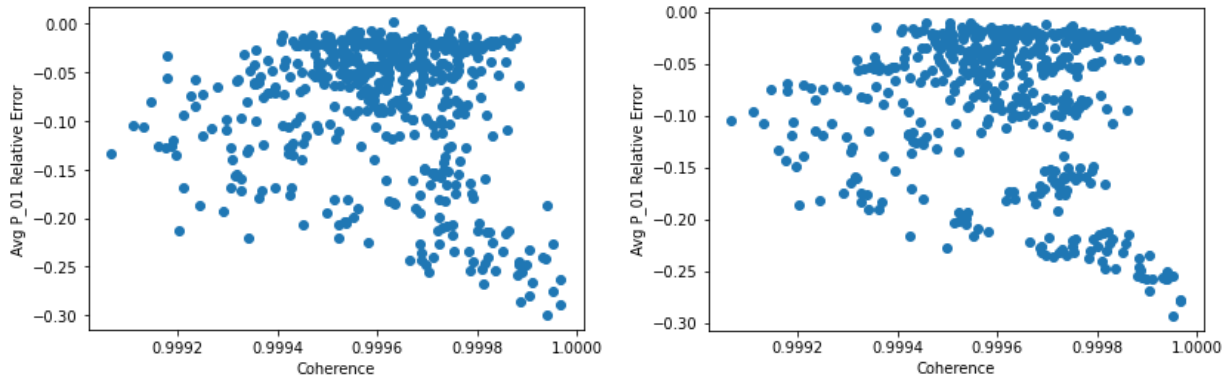


(a) 500 random combinations

(b) 50 minimum coherence combinations

Figure 5: Set averages of $P_{0,1}$ relative errors against coherence for (a) 500 randomly chosen combinations from each set and (b) the combinations from each set with the 50 minimum coherences. (a) shows no obvious relation, whereas (b) indicates the lower coherence may reduce the average $P_{0,1}$ estimate, with a correspondingly more negative relative error. Sets are labelled in the legends on the right of each graph.

ConVxOpt 2 versus 3: 500 combinations estimating 100 signals



(a) ConVxOpt2

(b) ConVxOpt3

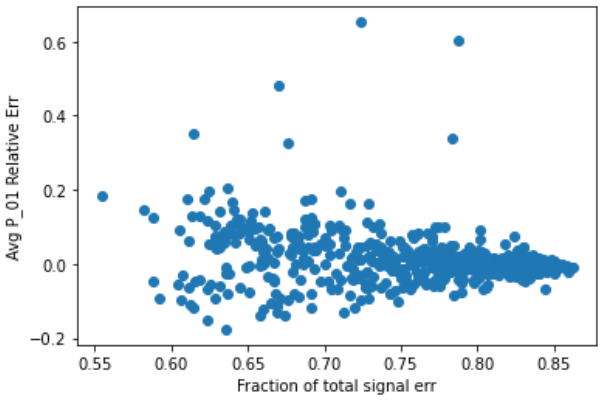
Figure 6: With 500 randomly chosen combinations from Set 10 estimating 100 signals, the average $P_{0,1}$ relative errors are plotted against coherence for (a) ConVxOpt2 with an overall average relative error of $P_{0,1} = -0.0845$ and (b) ConVxOpt3 with an overall average relative error of $P_{0,1} = -0.0841$. If low coherence was facilitating the optimisation objective, then ConVxOpt2 would be expected to have significantly lower average $P_{0,1}$ relative errors.

in the larger errorbars corresponding to smaller $P_{0,1}$ signals. This demonstrates a decrease in accuracy is more likely as the signal $P_{0,1}$ decreases, but to the negative side - aligning with our preference for errors that underestimate. Also, the loss in precision is not significant enough to compromise the underestimation of $P_{0,1}$. However, as seen in Fig. 8a, there are no signals with $P_{0,1}$ lower than approximately 0.35 and the optimisation method's accuracy may deteriorate significantly for signals with $P_{0,1}$ lower than this level. Taking estimation data for a wider range and distribution of signals would be useful in fully analysing the utility of this combination of detection efficiencies.

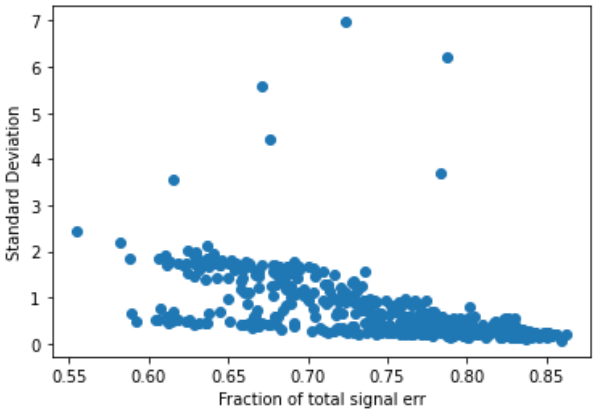
Finally, 20000 estimations were made of the single signal, as used for comparing ConVxOpt1 and 2, with the 'best' combination of efficiencies. The average estimate of $P_{0,1} \approx 0.737$, the input signal $P_{0,1} \approx 0.750$ with a 8.89% overestimation rate. Figure 10 shows the histogram of $P_{0,1}$ estimations. The performance of

this combination is very similar to the evenly spaced set of efficiencies. This combination, however, has a lower average $P_{0,1}$ estimate and over-estimates nearly twice as often. Nonetheless, this combination is very straight forward to implement in a practical setting.

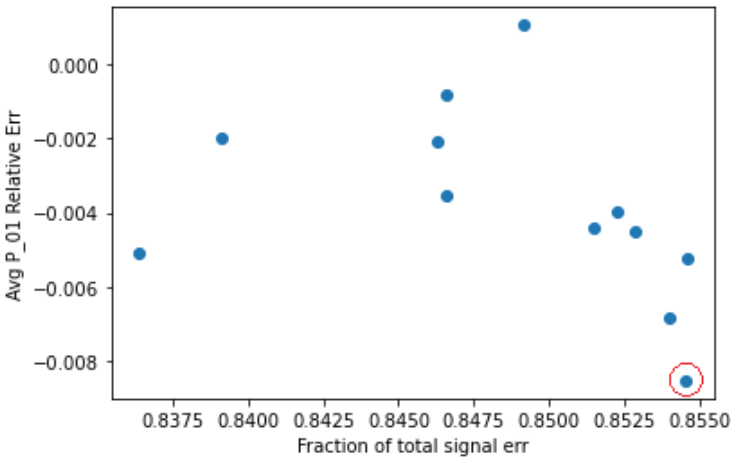
Set 14 combinations estimating 100 signals



(a) 500 random combinations

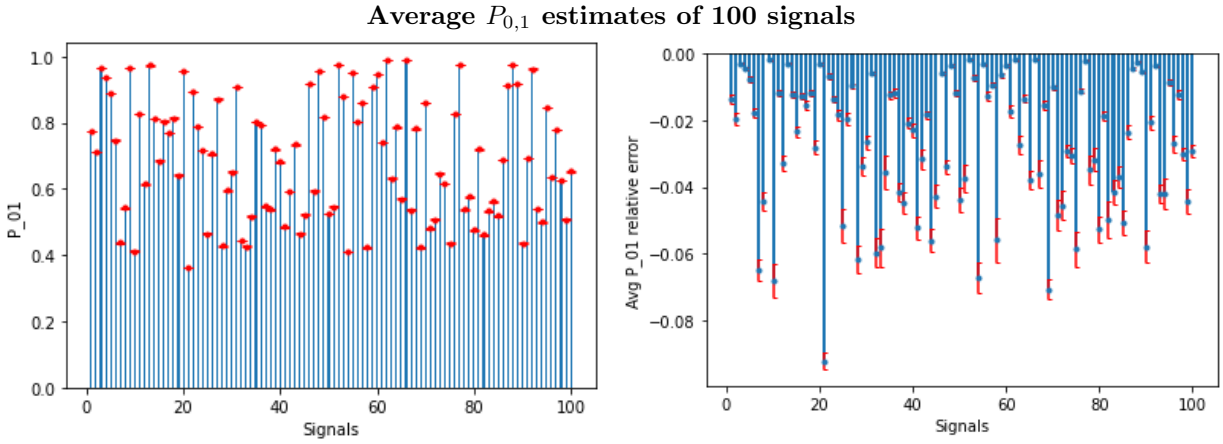


(b) 500 random combinations



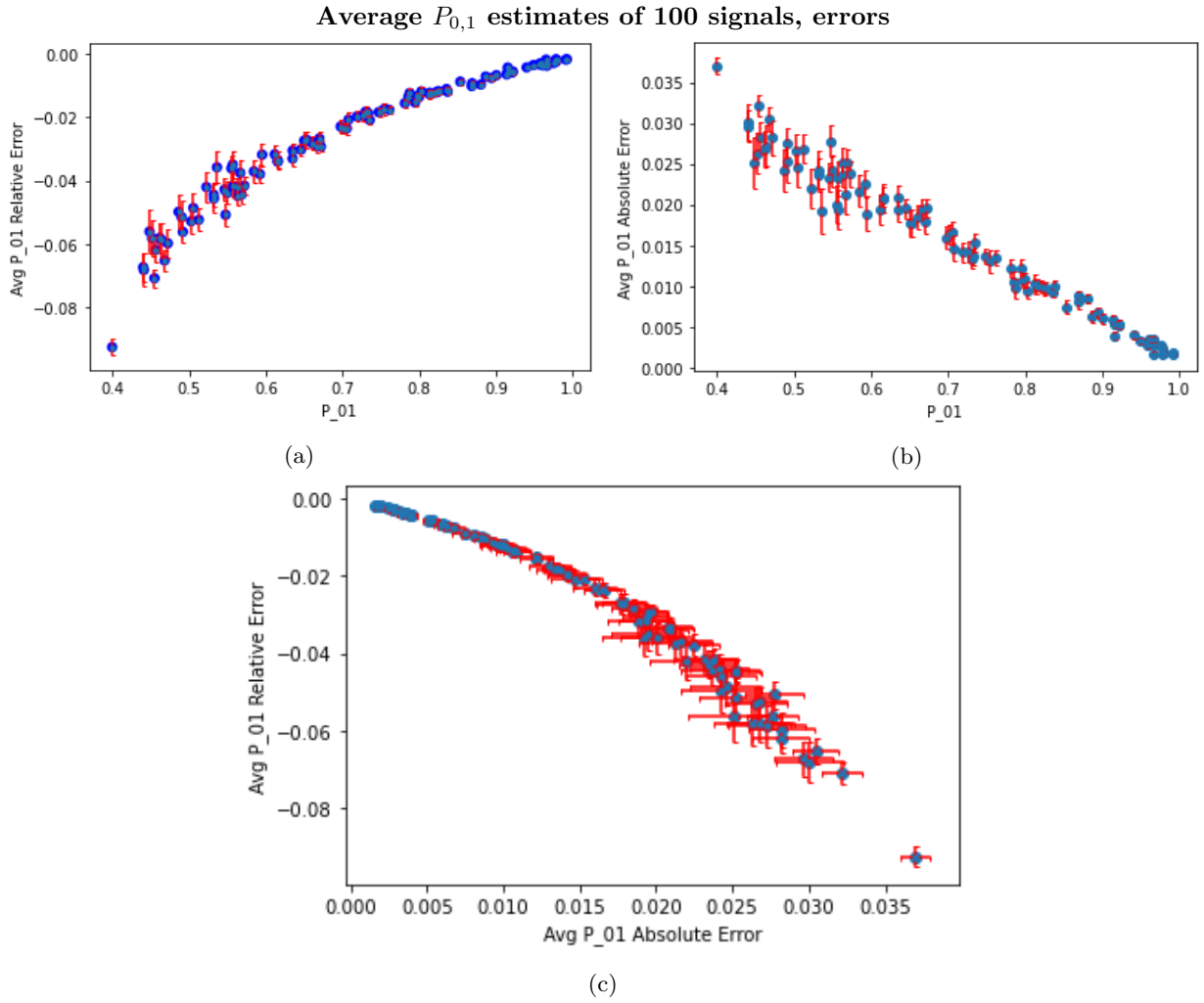
(c) 14 most accurate combinations

Figure 7: Combinations from Set 14 estimating 100 signals. (a) shows the average $P_{0,1}$ relative error against the fraction of the total signal error in P_m. (b) shows the standard deviation of the $P_{0,1}$ relative error against the fraction of the total signal error in P_m. (c) shows the average $P_{0,1}$ relative error against the fraction of the total signal error in P_m for the 14 combinations with the most accurate/precise estimations of $P_{0,1}$. The red circle indicates the combination selected as 'optimal'.



(a) (b)

Figure 8: Averages of 200 estimates of 100 signals by the 'best' combination of detection efficiencies. (a) shows the average estimates and (b) the relative errors. As seen the relative errors are all negative, as preferred. The errorbars are the standard error of the mean for the 200 estimations of each signal. Their narrowness in (a) attests to the precision of the procedure after just 200 estimations.



(a) (b)

(c)

Figure 9: (a) Shows the average relative error in $P_{0,1}$ increasing in magnitude as $P_{0,1}$ decreases. (b) Shows the average absolute error in $P_{0,1}$ increasing as $P_{0,1}$ decreases. And (c) shows the average relative error in $P_{0,1}$ increasing in magnitude as the average absolute error in $P_{0,1}$ increases. This indicates that the procedure may be become unreliable as the signal $P_{0,1}$ becomes small. However, we see that the procedure is increasingly accurate and precise as $P_{0,1}$ approaches 1.

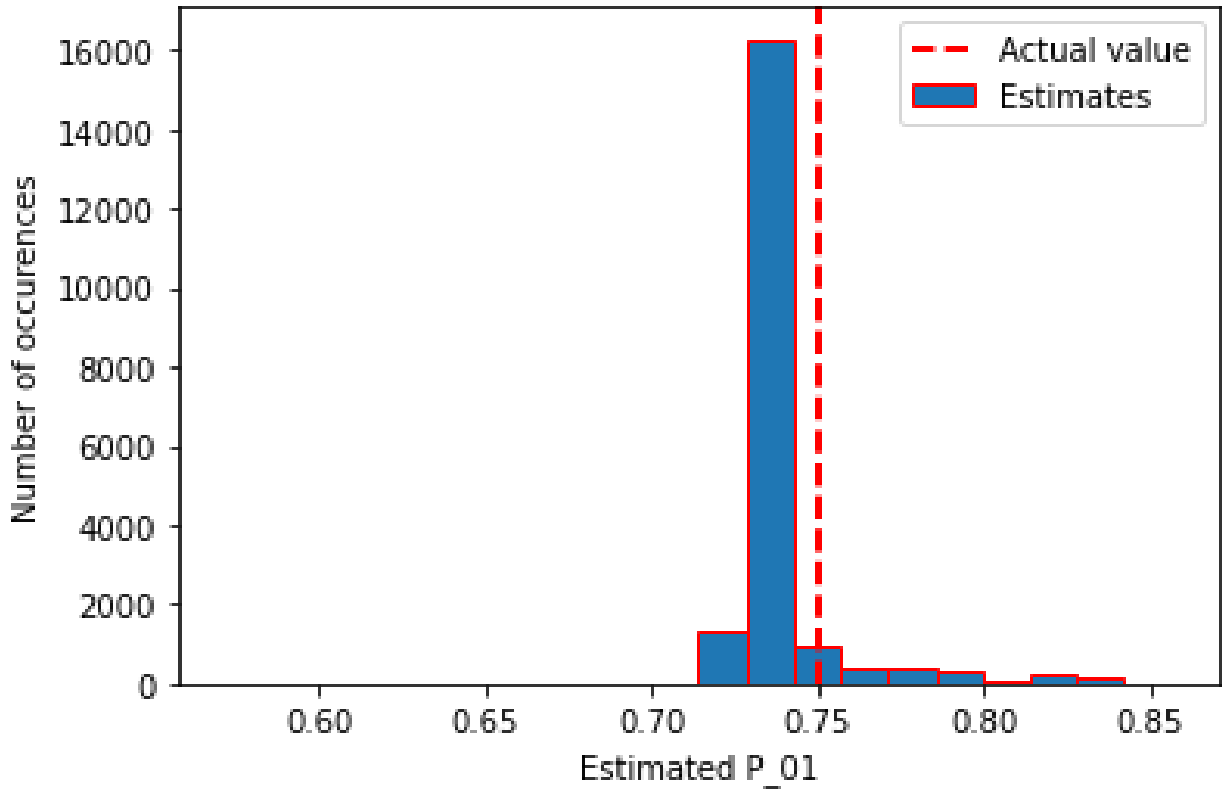


Figure 10: Histogram of 20000 estimations of the single input signal with the chosen 'best' combination of detection efficiencies. We see a sharp tall peak at 0.73 and a 'fat tail' extending into the over-estimation region - very similar to Fig. 4b

6 Conclusion

The hypothesis that a combination of detection efficiencies with a corresponding measurement matrix of lower mutual coherence would facilitate the optimisation objective was tested. This was done by contrasting 2 optimisation procedures with opposite objectives. However, as far as the data presented here suggests, the hypothesis does not appear to hold. The relative error in $P_{0,1}$ compared with the signal $P_{0,1}$ increases in magnitude with decreasing coherence for both optimisation protocols when the ConVxOpt3 protocol was expected to show a $P_{0,1}$ relative error decreasing in magnitude. This suggests the consistent underestimation of $P_{0,1}$ by all optimisation methods is being facilitated by some other factor. One possible suggestion is that convex optimisation has the effect of over-estimating smaller signal elements and under-estimating larger elements - and all the investigated signals are of a form with a large $P_{0,1}$. This could be explored by providing input signals with relatively larger P_m than $P_{0,1}$. It could also be the case that lower coherence does facilitate the optimisation objective. Perhaps, convex optimisation has a strong tendency to over/under estimate small/large elements and the optimisation facilitating effect of lower coherence is not strong enough to show in this work. Inspecting lower coherence combinations of detection efficiencies may provide more conclusive results in this regard. Optimising using the 50 minimum coherence combinations from Set 10 for the 100 signals with both ConVxOpt 2 and 3 could be a first point of further investigation.

This procedure, the measurement program and ConVxOpt2, is demonstrated to be effective in choosing an optimal combination of detection efficiencies for Fock state tomography. Whilst the combination chosen here was intended to be both accurate and provide an underestimate for $P_{0,1}$, it was to some extent arbitrary. One could use a similar methodology as presented above to choose a combination that provides a more accurate $P_{0,1}$ estimate on average. Or, to err further on the side of caution, a combination that provides lower estimates of $P_{0,1}$. As detailed above, this optimisation procedure appears to become less accurate and precise for signals with lower $P_{0,1}$ values. Further investigation would indicate if the procedure becomes ineffective for some sufficiently small P_{01} signal.

References

[1] Edo Waks, Charles Santori, and Yoshihisa Yamamoto. Security aspects of quantum key distribution with sub-poisson light. *Physical review. A, Atomic, molecular, and optical physics*, 66(4):423151–423157, 2002.

[2] Meenu Rani, S. B Dhok, and R. B Deshmukh. A systematic review of compressive sensing: Concepts, implementations and applications. *IEEE access*, 6:4875–4894, 2018.

[3] Emmanuel Candès. The restricted isometry property and its implications for compressed sensing. *Compte Rendus de l'Academie des Sciences*, 346:589–592, 05 2008.

[4] Marcio Pereira, Lisandro Lovisolo, Eduardo da Silva, and Marcello Campos. On the design of maximally incoherent sensing matrices for compressed sensing using orthogonal bases and its extension for biorthogonal bases case. *Digital Signal Processing*, 27, 04 2014.

[5] D.L. Donoho, M. Elad, and V.N. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on Information Theory*, 52(1):6–18, 2006.

[6] David Gross, Yi-Kai Liu, Steven T Flammia, Stephen Becker, and Jens Eisert. Quantum state tomography via compressed sensing. *Physical review letters*, 105(15):150401–150401, 2010.

[7] A Shabani, R.L Kosut, M Mohseni, H Rabitz, M.A Broome, M.P Almeida, A Fedrizzi, and A.G White. Efficient measurement of quantum dynamics via compressive sensing. *Physical review letters*, 106(10):100401–100401, 2011.

[8] Robert L. Kosut. Quantum process tomography via l1-norm minimization, 2009.

[9] Wei-Tao Liu, Ting Zhang, Ji-Ying Liu, Ping-Xing Chen, and Jian-Min Yuan. Experimental quantum state tomography via compressed sampling. *Phys. Rev. Lett.*, 108:170403, Apr 2012.

Appendices

A Detection Efficiency Combinations/Sets

Below is the definition, in Python, of the 6 evenly spaced detection efficiencies.

[0.8 - i*0.8/(6) for i in range(6)]

The following is a list of all the sets of detection efficiencies' definitions, from which groups of combinations were chosen. The intent was to form sets from which chosen combinations of efficiencies could form measurement matrices with relatively low coherence. Sets 2 and 14 are defined identically, however, the combinations chosen from Set 14 all include $\eta = 0.8$ whereas those chosen from Set 2 do not. Set 7 contains more possible efficiencies and was intended to provide very widely or narrowly spread combinations.

Set 1 := [(i + 1)*0.8/15 for i in range(15)]
Set 2 := [0.8/(2**i) for i in range(15)]
Set 3 := [0.8/(4**i) for i in range(15)]
Set 4 := [0.8 - 0.79/(2**i) for i in range(15)]
Set 5 := [0.5/(2**i) for i in range(15)]
Set 6 := [0.8 - 0.01*i for i in range(8)] + [0.01 + 0.01*i for i in range(7)]
Set 7 := [0.8 - 0.01*i for i in range(25)] + [0.01 + 0.01*i for i in range(25)]
Set 8 := [0.3/(np.sqrt(2)**i) for i in range(15)]
Set 9 := [0.8,0.4] + [0.2/(2**i) for i in range(13)]
Set 10 := [0.8/(np.sqrt(2)**i) for i in range(15)]
Set 11 := [0.8] + [(0.4/np.sqrt(2))/(np.sqrt(2)**i) for i in range(14)]
Set 12 := [0.8] + [0.1/(np.sqrt(2)**i) for i in range(14)]
Set 13 := [0.8 - 0.4/(np.sqrt(2)**i) for i in range(14)]
Set 14 := [0.8] + [0.4*(0.5)**i for i in range(14)]